Hochschule Hannover

University of Applied Sciences and Arts

Faculty I – Electrical and Information Engineering

in collaboration with

ABB AG

Master Thesis

Impact of mixed speed networks on the performance of Ethernet real time communication protocols like PROFINET

Document updated after submission

submitted from:

name:	Lukas Krapp
matriculation number:	1538578

period:

from:	01.03.22
till:	31.08.22
company supervisor:	Dipl.-Ing. Holger Grosse
first examiner:	Prof. Dr.-Ing. Karl-Heinz Niemann
second examiner:	Dipl.-Ing. Holger Grosse

I. Declaration of Independence

I hereby certify that I have prepared this master thesis independently and only using the sources and aids indicated. All passages in thesis that have been taken verbatim or in spirit from other sources are marked as such. I have not submitted thesis in the same or similar form to any other examination authority.

Hanover, 21. April 2023
Lukas Krapp

This is an updated and corrected version of the thesis. The updates, corrections and extensions are based on the review comments of the first and second examiners and other individuals that were provided after the submission of the thesis.

Hannover. 21. April 2023
Lukas Krapp

II. Abstract

The demand for a unified industrial network based on the Ethernet standard continues to rise. In the context of process automation, this means linking the individual network levels by means of a unified network communication, so that in the future all network participants speak the same language.

At the same time, this means that the different requirements of each individual network level will clash with one another and thus make unification difficult. One of these challenges concerns the convergence of network layers that operate at different datarates and thus form a '*mixed link speed*' network.

The combined use of industrial Ethernet with Ethernet-APL can lead to the described outcome. Ethernet-APL working in combination with industrial Ethernet, connects the field device level with the control/monitor level, which achieves a unified Ethernet architecture throughout the whole process automation network. The integration of Ethernet-APL working at $10\ \mathrm{Mbit/s}$ in an already existing industrial Ethernet network usually operating at higher link speeds concludes in a '*mixed link speed*' network.

To verify the impact on network performance in such scenarios with focus on potential packet loss of high-priority traffic, this thesis analyses the packet processing behavior of real Ethernet-APL switch devices when operating in a '*mixed link speed*' environment.

For this purpose, various measurements have been performed on the APL switches in question, based on different test scenarios with varying packet flow direction and packet utilization. The basis for this is provided by two best-practice '*mixed link speed*' sample networks from practice consisting of a $100\ \mathrm{Mbit/s}$ industrial Ethernet and $10\ \mathrm{Mbit/s}$ Ethernet-APL layer in which the APL switches were tested.

In a nutshell, the measurements showed that the APL switches are capable of successfully processing packets without any packet loss. This conclusion is based on various test conditions, including such tests, which set the basic requirements for down- and upstreaming packet data according to PROFINET specific robustness requirements.

However, even when further increasing packet load beyond these requirements, the APL switches managed to maintain correct packet processing in most test scenarios. Only a few specific packet overload scenarios lead to packet processing problems with packet loss. However, it should be pointed out that these tests focused on edge cases that are not necessarily likely to occur in real industrial networks.

In conclusion, it can be said, that by analyzing the packet-throughput behavior of the APL switches, the potential of packet loss in the '*mixed link speed*' networks tested is relatively low, so that it is not a threat to correct network behavior.

This ensures that the future implementation of '*mixed link speed*' networks should no longer pose a challenge in terms of packet utilization if the equipment is designed correctly.

III. Table of Contents

IV. Symbols, abbreviations and acronyms

Symbols

symbol	unit	description
FPC	Bit/cycle	Total Frame Payload per Cycle
FPS	Bit/s	Total Frame Payload per Second
PCC	packets/cycle	Packet Count per Cycle
PCS	packets/s	Packet Count per Second
PCT	s	Packet Cycle Time
PPC	Bit/cycle	Packet data Payload per Cycle
PPS	Bit/s	Packet data Payload per Second
PPT	s	Packet Processing Time
PPT_{type}	s	Packet Processing Time for a specific traffic type (UDP, TCP, ARP)
PPT_{total}	s	Total Packet Processing Time accumulated by the sum of multiple PPTs
$PPS_{\text{type,total}}$	s	Total packet processing time accumulated by the sum of a specific traffic type (UDP, TCP, ARP)
TPP	Bit	Total Packet Payload
$bufferCount_{\text{B}}$	Bit / packets	Buffer memory limit of packet processing hardware of APL switch by Manufacturer B
$bufferCount_{\text{B,min}}$	Bit	Minimum buffer memory limit of Manufacturer B
$bufferCount_{\text{B,max}}$	Bit	Maximum buffer memory limit of Manufacturer B
$bufferLength_{\text{A}}$	Bit	Buffer memory limit of packet processing hardware of APL switch by Manufacturer B
$bufferLength_{\text{A,min}}$	Bit	Minimum buffer memory limit of Manufacturer A
$bufferLength_{\text{A,max}}$	Bit	Maximum buffer memory limit of Manufacturer A
$datarate_{\text{in}}$	Bit/s	Datarate of the APL switch ingress port
$datarate_{\text{out}}$	Bit/s	Datarate of the APL switch egress port
$dataRate_{\text{A\&B,max}}$	Bit/s	Maximum datarate of the APL switch line of Manufacturer A&B
$linkspeed_{\text{ratio}}$	–	Ratio between APL switch ingress and egress port datarate
$l_{\text{APL,spur,max}}$	m	Maximum cable length of an APL spur line
$l_{\text{APL,trunk,max}}$	m	Maximum cable length of an APL trunk line
$packet_{\text{load}}$	Bit	Packet load of one or multiple packets
$packet_{\text{load,measured}}$	Bit	Measured packet load
$packet_{\text{load,type}}$	Bit	Packet load for a specific traffic type (UDP, TCP, ARP)

$packet_{\text{load,type,cycle}}$	Bit	Packet load for a specific traffic type (UDP, TCP, ARP) in a specific cycle
$packet_{\text{load,total}}$	Bit	Total packet load accumulated by the sum of multiple packet loads
$packet_{\text{size}}$	Bit	Packet size of an Ethernet frame
$packet_{\text{size,min}}$	Bit	Minimum packet size of an Ethernet frame
$packet_{\text{size,max}}$	Bit	Maximum packet size of an Ethernet frame
$packet_{\text{size,type}}$	Bit	Packet size of an Ethernet frame for a specific traffic type (UDP, TCP, ARP)
$queueLength_{\text{A}}$	Bit	Queue memory limit of packet processing hardware of APL switch by Manufacturer B
$queueLength_{\text{min}}$	Bit	Minimum queue memory limit of Manufacturer B
$queueLength_{\text{max}}$	Bit	Maximum queue memory limit of Manufacturer B
$T_{\text{traffic}_{\text{type}}}$	s	Set cycle time for a specific traffic type (UDP, TCP, ARP)
t_{bridge}	s	Bridge delay
$t_{\text{bridge,cut}-\text{through}}$	s	Cut-through bridge delay
$t_{\text{bridge,store\&forward}}$	s	Store & forward bridge delay
t_{cable}	s	Cable delay
t_{port}	s	Port delay
$t_{\text{port,RX}}$	S	Receival port delay
$t_{\text{port,TX}}$	s	Transmission port delay
t_{prop}	s	Propagation delay
x_{devices}	–	Number of field device emulators
$x_{\text{packets,type}}$	packets	Set number of packets for a specific traffic type (UDP, TCP, ARP)
$x_{\text{packets,type,max}}$	packets	Maximum number of packets processable for a specific traffic type (UDP, TCP, ARP) in the PPT of the packet processing hardware

List of abbreviations / acronyms

Abbreviation/ acronym	Description
APL	Advanced Physical Layer
ARP	Address Resolution Protocol
CRC	Cyclic Redundancy Check
DCP	Discovery and Configuration Protocol
D-MAC	Destination MAC
DUT	Device Under Test
FCS	Frame Check Sequence
FDB	Forwarding database
FDX	Full Duplex Communication
FIFO	First-In First-Out
FPC	Total Frame Payload per Cycle
FPS	Total Frame Payload per Second
IFG	Inter Frame Gap
IP	Internet Protocol
LAN	Local Area Network
LLC	Logical Link Control
MAC	Media Access Control
MDI	Media Dependent Interface
MII	Media Independent Interface
NRT	Non Real Time
PCC	Packet Count per Cycle
PCS	Packet Count per Second
PCT	Packet Cycle Time
PDU	Protocol Data Unit
PHY	Physical Layer
PoDL	Power Over Data Lines
PPC	Packet data Payload per Cycle
PPS	Packet data Payload per Second
PPT	Packet Processing Time

PROFINET	Process Field Network
PUP	Packet User Priority
QoS	Quality of Service
RT	Real Time
RTC	Real Time Cyclic
RX	Receival
SFD	Start of Frame Delimiter
S-MAC	Source MAC
SPE	Single Pair Ethernet
TAP	Test Access Point
TCP	Transmission Control Protocol
TPP	Total Packet Payload
TX	Transmission
UDP	User Datagram Protocol
VID	VLAN Identifier
VLAN	Virtual LAN

1. Introduction

The trend towards the use of Ethernet in automation networks is ongoing. Due to its high flexibility, speed, and bandwidth, Ethernet nowadays is not only widely used in homes and offices worldwide but finding its way into industrial applications. Especially in automation processes, where many field devices send data in relative short time spans, the requirements for a safe and fast data transfer are high. This makes the use of industrial Ethernet essential.

A new hardware-layer, specifically tailored for industrial applications, has been introduced in the form of Ethernet-APL ('*Advanced Physical Layer*'). Ethernet-APL is based on the Ethernet standard as specified in [EAPL_PPS_01] and [IEEE_8023CG_01] and implements a two-wire Ethernet-based communication for field devices and provides data and power over a two-wire cable. The operation in areas with potentially explosive atmosphere is also possible. This enables a modular, fast, and transparent Ethernet network structure throughout the entire plant. [PNO_EAPL_01]]

However, by integrating Ethernet-APL into the field, industrial networks in the future will face the challenge of operating at varying datarates at different locations in the network, resulting in a '*mixed link speed*' network. This can lead to limitations in packet-throughput and consequently to potential packet loss of system relevant data, which must be avoided. Therefore, the purpose of this thesis is to investigate the potential of packet loss in '*mixed link speed*' networks.

To fulfill the purpose of this thesis, the following chapters describe the problem and subsequent structure of measures to analyze and solve the problem, in more detail.

Chapter 1.1 describes the problem of packet processing in a mixed link speed network. It explains where corresponding problems areas arise when linking network layers that operate at different datarates and how these can lead to a potential risk for the system functionality.

Chapter 1.2 describes appropriate measures which are used to evaluate the impact of '*mixed link speed*' networks in regard to potential of packet loss.

Chapter 1.3 defines work packages suitable for implementing the measures identified in the previous chapter.

Additionally, the thesis consists out of the following, briefly described, chapters:

Chapter 2 summarizes theoretical background information needed to understand the terminology used in this thesis in full detail.

Chapter 3 describes the results of simulation test conducted that provides information how an Ethernet switch's internal packet process operates in detail.

Chapter 4 gives insight about the hardware test conducted on real Ethernet-APL switches, analyzing their packet-processing behavior, while being operated in a '*mixed link speed*' network.

Chapter 5 summarizes the given hardware limitations, provided by the manufacturers, in regard to packet processing capability of the tested APL switches.

Chapter 6 explains the structure of the hardware test setup based on the hardware and software used to conducting the measurements.

Chapter 7 summarizes all conducted measurements in a comprehensible and compact listed form.

Chapter 8 concludes thesis summarizing the found results to the initial problem analyzed and gives an outlook for potential improvements of the hardware test setup.

1.1. Problem description

The evolution of Ethernet provides a variety of different datarates, also called '*link speed*'. With respect to the use in process automation networks, the link speeds range from 10 Gbit/s at the upper end, all the way down to 10 Mbit/s. Currently 100 Mbit/s Fast-Ethernet is mainly used in the automation domain. This is going to change in the future. On the one hand, higher datarates are used for the main traffic routes of the automation network, often referred to as '*backbone*'. On the other hand, field devices such as sensors and actuators will be connected with 10 Mbit/s via a two-wire Ethernet in the future. This leads to a situation in which different parts of the network run at different link speeds. By combining the 100 Mbit/s industrial Ethernet layer with the 10 Mbit/s Ethernet-APL, layer a mixed link speed network is created. Two typical mixed link speed structures are shown below.

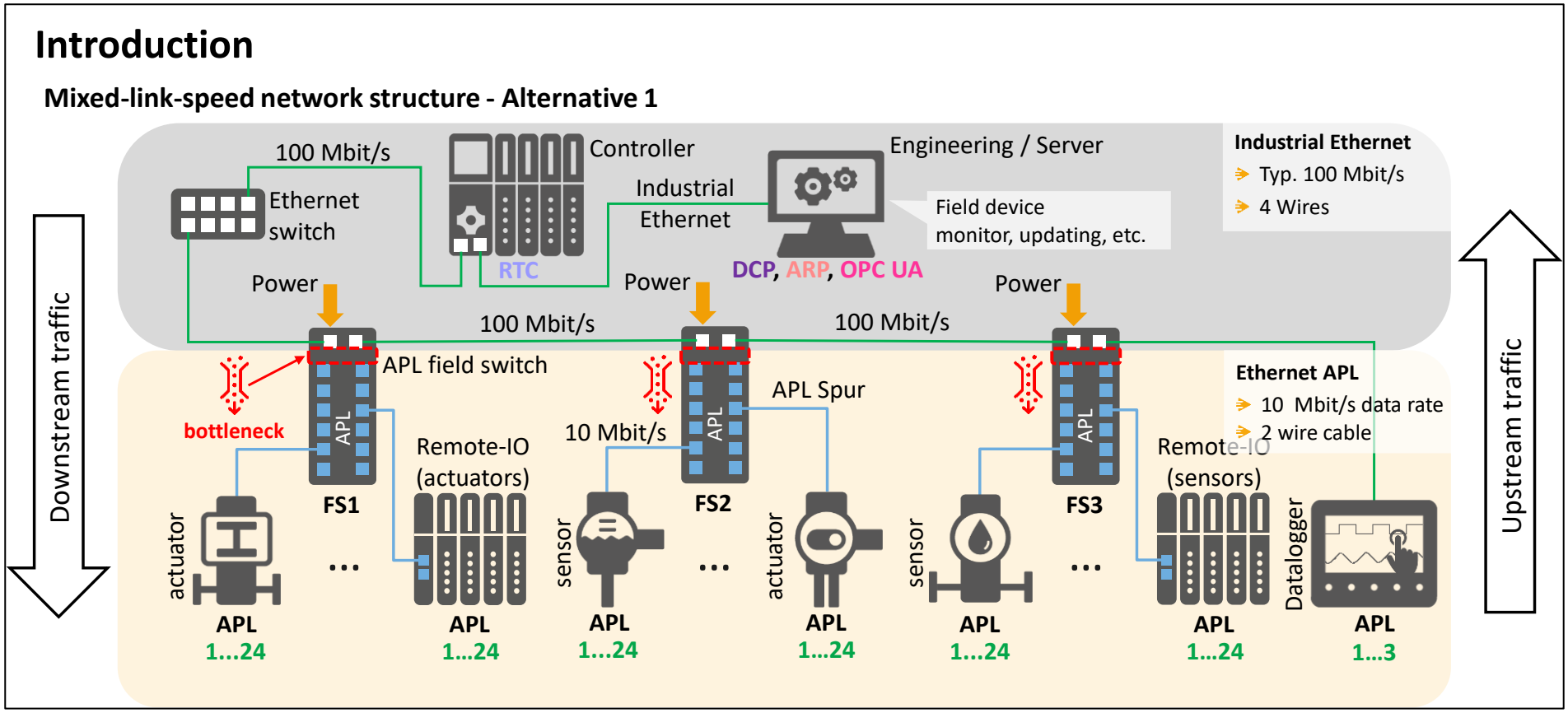

Figure 1: Mixed speed network including APL switches with industrial Ethernet (100 Mbits/s) connection, [NieK_EAPL_01, p.57 et seq.]

Figure 1 resembles a typical mixed link speed network and illustrates the connection between industrial Ethernet, shown by green connections, and Ethernet-APL, shown by blue connections.

The network features network participants typical for process automation and includes switches, controllers, Remote-IOs, field devices, servers, and data loggers. These participants exchange data with each other in the form of different traffic types (see chapter 2.1.3), while data flows both from the higher control/supervisory level to the field device level, labeled as '*Downstream traffic*', and vice versa, labeled as '*Upstream traffic*'.

Figure 1 shows that the upper-level participants of the network (i.e., controller, server) are connected via industrial Ethernet ports running at $100\ \mathrm{Mbit/s}$, whereas the lower-levels (i.e., field devices) are connected via Ethernet-APL spur ports operating at $10\ \mathrm{Mbit/s}$.

The field switches connecting the traffic flow between the upper and lower level handle the change in link speed connection between the $100\ \mathrm{Mbit/s}$ industrial Ethernet (green lines) and the $10\ \mathrm{Mbit/s}$ Ethernet-APL (blue lines). Said transmission change occurs at all Ethernet-APL field switches (i.e. FS1, FS2, FS3) in line. The field switches connecting the traffic flow between the upper and lower level (see red marking inside Figure 1 , which resembles the bridge entity inside a switch) is caused by the fact that the speed of the network goes down from $100\ \mathrm{Mbit/s}$ to $10\ \mathrm{Mbit/s}$.

When downstreaming traffic to the lower-level, the APL field switch handles packet transmission via its $100\ \mathrm{Mbit/s}$ industrial Ethernet port down to the respective filed device connected via $10\ \mathrm{Mbit/s}$ Ethernet-APL spur ports. When upstreaming traffic to the higher-level, multiple 10 Mbit/s Ethernet-APL spur ports merge in the APL field switch where the data is then forwarded through a single $100\ \mathrm{Mbit/s}$ industrial Ethernet port.

Packets that are passing through said levels to reach their destination, encounter a bottleneck at the APL field switch which can cause a congestion loss problem.

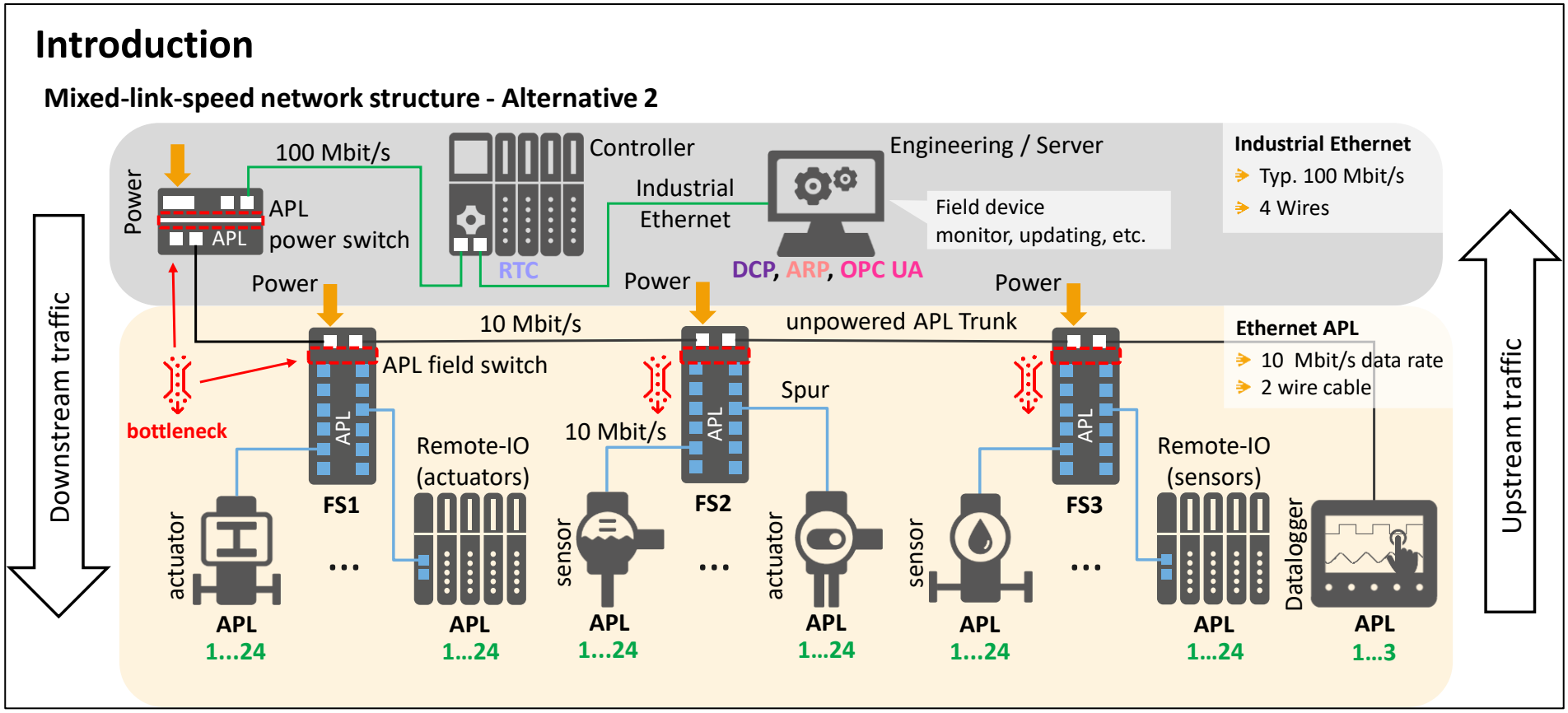

Figure 2: Mixed speed network including APL switches with Ethernet-APL (10 Mbit/s) connection, [NieK_EAPL_01, p.57 et seq.]

Figure 2 illustrates another example of a mixed-speed network, which results from the combination of industrial Ethernet and Ethernet-APL.

Here, the bottleneck resides at two different locations. When downstream traffic occurs, the APL power switch must handle the transmission speed change from its 100 Mbit/s industrial Ethernet port down to the 10 Mbit/s Ethernet-APL trunk connection. When upstream traffic occurs, the same bottleneck as described for Figure 1 arises. This time, however, the trunk port responsible for forwarding packets sent from the lower level is also limited to 10 Mbit/s.

The bottlenecks describe a situation in which data is travelling through the network at a high datarate, suddenly passes through a part of the network with a significantly lower transmission speed. This phenomenon is often described as a '*congestion problem*'. In case the switch is not able to handle the congestion situation, a congestion loss (drop of data packets) might happen.

Note: Additional information regarding congestion can be found in the PROFINET guideline. [PNO_8061_01, p. 164 et seq.]

Congestion loss can be depicted with a water bucket model.

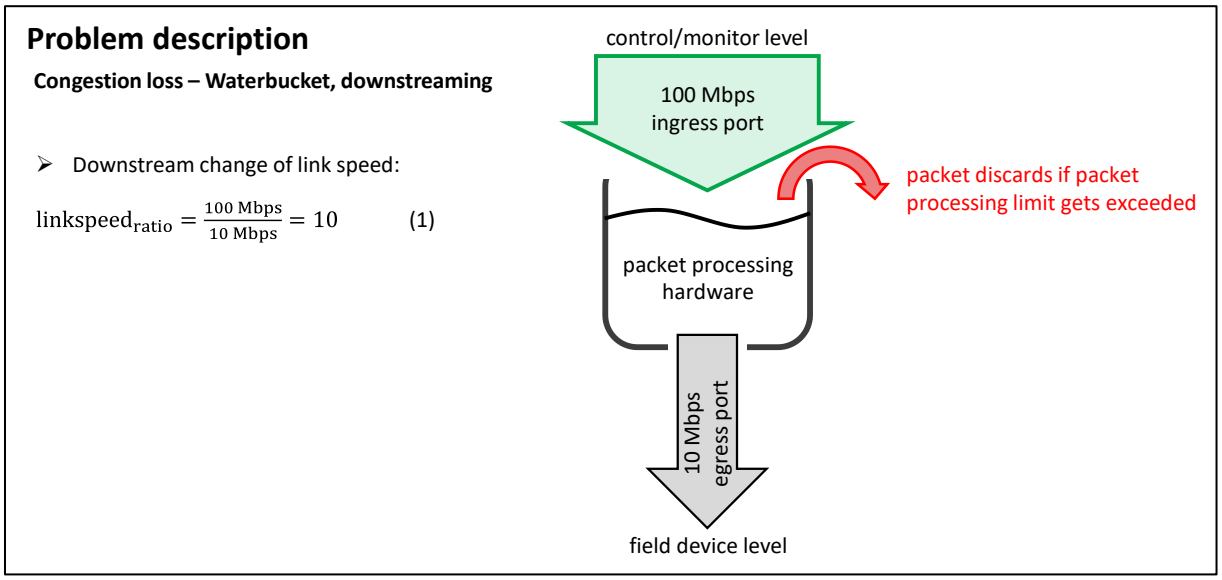

Figure 3: water bucket model, downstream flooding

Figure 3 shows the packet data traveling through the water bucket in downstream direction. In the figure, packet data is coming in from the upper control/monitor level down to the lower field device level. For these packets, the datarate transitions from $100\ \text{Mbit/s}$ down to $10\ \text{Mbit/s}$ occurs.

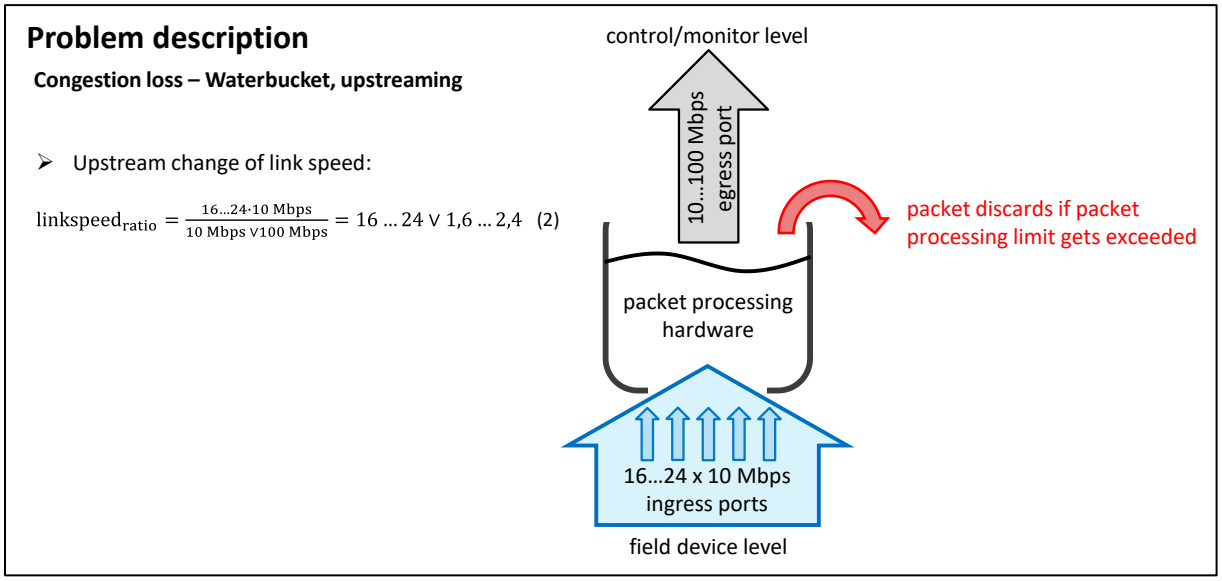

Figure 4: water bucket model, upstream flooding

Figure 4 resembles the opposite direction of data flow through the water bucket in comparison to Figure 3. Here data runs into the switch via multiple $10\ \mathrm{Mbit/s}$ connections up to one single $100\ \mathrm{Mbit/s}$ or $10\ \mathrm{Mbit/s}$ connection.

In both Figure 3 and Figure 4, the water bucket resembles the limitations of the hardware responsible for packet processing and subsequent transmission between network participants.

Said hardware is represented by an Ethernet-APL switch, which handles in- and outgoing packets via its internal packet processing hardware, as well as the data lines connected to the switch, transmitting those packets processed and forwarded by the switch.

The total packet capacity of APL switch and data line is resembled by the total volume of the water bucket. If the water bucket gets full, any subsequent packets '*spill over*' and get discarded, causing packet loss. Such an event usually is caused by the arrival of a multitude of incoming packets '*flooding*' the switch. This can happen via one or even multiple ports.

When downstream packets are processed according to Figure 3, packet flow starts at a single sender from the control/supervisory level transmitting packets via a $100\ Mbit/s$ connection to the APL switch and narrows down to multiple 10 Mbit/s connections, which forward all processed packets by the switch to its specific receiver at the field device level.

The result is a '*10-to-1*' change in transmission speed which is calculated as follows:

$$linkspeed_{ratio} = \frac{datarate_{\mathrm{in}}}{datarate_{\mathrm{out}}} \tag{1}$$

$$linkspeed_{\mathrm{ratio,100-10\ Mbit/s}} = \frac{100\ Mbit/s}{10\ Mbit/s} = 10 \tag{2}$$

When upstreaming packets, according to Figure 4, are processed, the packet flow starts at multiple senders from the control/supervisory level transmitting packets via a $10\ \mathrm{Mbit/s}$ connection, and narrows down to a single $10\ \mathrm{Mbit/s}$ connection, forwarding all processed packets to its specific receiver at the control/supervisory level.

While the change in transmission speed for downstreaming packets is limited to a '*10-to-1*' ratio, the change in transmission speed can get amplified by receiving data via multiple ports:

$$linkspeed_{\mathrm{ratio,16\ldots24-100\ Mbit/s}} = \frac{16\ldots24 \cdot 10\ Mbit/s}{100\ Mbit/s} = 1{,}6 \ldots 2{,}4 \tag{3}$$

$$linkspeed_{\mathrm{ratio,16\ldots24-10\ Mbit/s}} = \frac{16\ldots24 \cdot 10\ Mbit/s}{100\ Mbit/s} = 16 \ldots 24 \tag{4}$$

If the packet processing hardware of the switch, at which the change in link speed is occurring, has either not enough overall capacity to receive the packet floods or if subsequent packets arrive at a faster speed/pace than the hardware is able to forward them, packet discards can occur.

Thus, packet discards occur when the number of packets exceed the limitations of the Ethernet switch (queues & buffer packet size) or its connections for data transfer (line datarate). Discarding can lead to faulty behavior of the plant and must be avoided, especially for real-time protocols.

1.2. Assignment

Based on the previously examined '*congestion loss*' problem in '*mixed link speed*' networks the following questions arise:
What happens if a larger number of packets are sent…

- … **downstream** at the same time, creating a **'*mixed traffic*'** scenario, at the APL switch?
- … **upstream** at the same time, creating a '***bursty traffic*'** scenario, at the APL switch?

> **Note:** The mixed traffic scenario describes packet load generated with multiple different traffic types with different packet casting, packet load, packet count and packet priority characteristics, sent by one or multiple senders to one receiver.
>
> The bursty traffic scenario describes packet load generated with a single traffic type but sent by multiple senders to one receiver.
>
> For more information regarding traffic and packet characteristics, refer to chapter 2.1.2.

For answering these questions, a test setup shall be defined and implemented that allows an analysis of the hardware's packet processing behavior. Based on the behavioral evaluation, conclusions can be drawn as to how great potential of packet loss is when using '*mixed link speed*' networks.

Therefore, the following tasks can be specified:

1) Definition of a test and measurement setup for packet processing analysis of an Ethernet switch operating in a '*mixed link speed*' network:
   a) The setup should implement a change in link speed from $100\ \mathrm{Mbit/s}$ to $10\ \mathrm{Mbit/s}$, according to Figure 1 and Figure 2.
   b) The setup should allow for the generation of packet data in down- and upstream direction with variable traffic type, packet casting, packet load, packet count and packet priority characteristics, send by one or multiple sending devices to the receiving switch.
   c) The setup should allow an analysis of the behavior of the switch, based on its measured packet processing by identifying received, forwarded, and discarded packets send from the switch.
2) Analysis and documentation of the impact caused by '*mixed link speed*' networks:
   a) Downstream: while handling multiple traffic sources, caused by different traffic types running in parallel in the network.
   b) Upstream: while handling information bursts caused by multiple traffic sources sending information simultaneously.
3) Description of which traffic scenarios can lead to packet loss of system relevant data when using '*mixed link speed*' networks.

1.3. Structure of the thesis

Based on the tasks described in the assignment, the following work packages can be derived:

1) Definition of a test and measurement setup for network analysis, regarding packet processing for downstream and upstream traffic in '*mixed link speed*' networks:

   a) Based on a **network simulation model** build with '*OMNet++*' [Omnet_01], consisting of the following parts:

      - Chunk-based packet generation with variable packet number, packet size, packet rate and packet priority
      - Packet processing based on simulating an Ethernet-APL switch at the physical layer (ISO/OSI Layer 1)
      - Packet measurement of received, forwarded, and discarded packets for in-depth analysis of packet processing inside the switch

> **Note**: By limiting the simulation model of the Ethernet switch to its physical layer, the simulation model created in '*OMNet++*' can work with chunk-based generated packets. Such packets consist of a random binary bit sequence to achieve a set amount of load per packet.
>
> However, even without any additional packet characteristics apart from the packet size, the packet number, packet rate and packet priority are still configurable by the use of appropriate simulation submodules that are used to condition the packets.
>
> For more information regarding the simulation model, refer to chapter 3.

   b) Based on a **hardware test setup** build with real devices, consisting of the following parts:

      - Frame-based (IPv4) packet generation with variable packet number, packet size, packet rate and packet priority, based on the frame generator tool '*Ostinato*'
      - Packet processing based on testing real Ethernet-APL switches as DUT ('*Device Under Test*') at the network layer (i.e., ISO/OSI layer 3)
      - Packet measurement of received, forwarded, and discarded packets for in-depth analysis of packet processing outside the switch, based on the frame analysis tool '*WireShark*'

> **Note**: While working with real Ethernet switches, only frame-based packet data is accepted by the device. Such packets contain a proper frame header, according to the ISO/OSI layer 2, also called data link layer. [SpuCha_01, p. 17] The frame header defines the traffic type, casting type, packet priority, and many other characteristics of a packet apart from its load.
>
> In addition to the MAC address, IPv4 generated packets additionally use the IP address to transmit frames from station to station, thus using the ISO/OSI layer 3, also called network layer. [Plixer_01]
>
> For more information regarding the hardware test setup, refer to chapter 4.

As described in the work packages above, first a simulation-based model of an APL switch is planned with the network simulation software '*OMNet++*', paired with its Ethernet framework '*INET*'. The highly modular and adaptable simulation build shall give an insight into the packet processing behavior of an Ethernet switch at its physical layer.

By understanding the specific behavior of the switch during packet processing, appropriate analyses can subsequently be performed on real devices. For this purpose, real APL switches are to be tested using various packet load scenarios.

2. State of the art

To describe the scope of this thesis, Ethernet and PROFINET terminology will be used. A brief explanation on a few select aspects of these two topics shall be given in the following pages to provide a basic understanding.

Chapter 2.1 addresses a few principles of Ethernet technology. This includes the definition of '*Ethernet*' in terms of its implementation, structure, architecture, and functionality as far as it is deemed necessary for this thesis.

Afterwards in chapter 2.1.2, a closer look is taken at common implementations of communication technologies in industrial environments, such as '*PROFINET*', as well as the newest extension to the latter, called '*Ethernet-APL*'. Afterwards, the reader should have a basic understanding of the role of the described technologies in an actual automation setup.

2.1. Basics of Ethernet technology

The term '*Ethernet*' envelopes a wide variety of topics from which a selected few will be explained in the following chapter. For a more detailed insight into Ethernet, a look into the relevant literature and standards is advised.

2.1.1. Fast (Switched) Ethernet

When talking about '*classical*' Ethernet in industrial environments, one typically means '*Fast Ethernet*', also called '*Switched Ethernet*' or '*Switched LAN*'. [KocR_01, p. 41 et seq.] [IEEE_8023_01, p. 22 et seq.]

Fast Ethernet can be operated in various topologies (star, tree, line, ring) and their combinations. Common topology examples in the process automation environment are ring structures, which branch into lines such as the daisy chain structure presented in Figure 1 and Figure 2 (chapter 1.1).

Fast Ethernet operates with Ethernet switches in FDX ('*Full Duplex Communication*'). This means that each network participant can communicate with each other in both transmission directions at the same time. This is achieved by having a pair of wires reserved for each direction. This way packets exchanged between stations do not occupy the connection line of one another, so that both participants can use the transmission medium simultaneously, enabling the parallel transmission of data.

Note: The Ethernet standard [IEEE_8023_02, p. 56 et seq.] uses the terminology '*station*' for all participants inside a network, thus using said terminology going forward.

2.1.2. Ethernet signals

Packets and Frames

Data carrying information which travels through a network at its core consists solely of binary information. To ensure that said data reaches its sender and not get lost on the way, while also maintaining its integrity, it gets enveloped in a '*Ethernet frame header*' (see Figure 5).

Note: The Ethernet standard specifies the commonly called '*Ethernet frame header*' with the terminology '*MAC frame*' which derives into three separate formats:

a) Basic frame: standard frame format for basic data transmissions

$$7\ \text{Byte Preamble} + 1\ \text{Byte SFD} + 18\ \text{Bytes Frame header} + 46 \dots 1500\ \text{Bytes Data}$$

b) Q-tagged frame: advanced frame functionality utilizing the '*Q-Tag*' for packet prioritization

7 Byte Preamble + 1 Byte SFD + 18 B Frame header + 46 … 1500 B Data + 4 B Q − Tag

c) Envelope frame: advanced frame functionality utilizing the '*Envelope Prefix/Suffix*' for protocol functions

7 Byte Preamble + 1 Byte SFD + 18 B Frame header + 46 … 1500 B Data

+2 … 482 B Enevelope Prefix/Suffix

All three frame types use the same Ethernet frame format. [IEEE_8023_03, p. 118 et seq.] [SpuCha_02, p.45 et seq.]

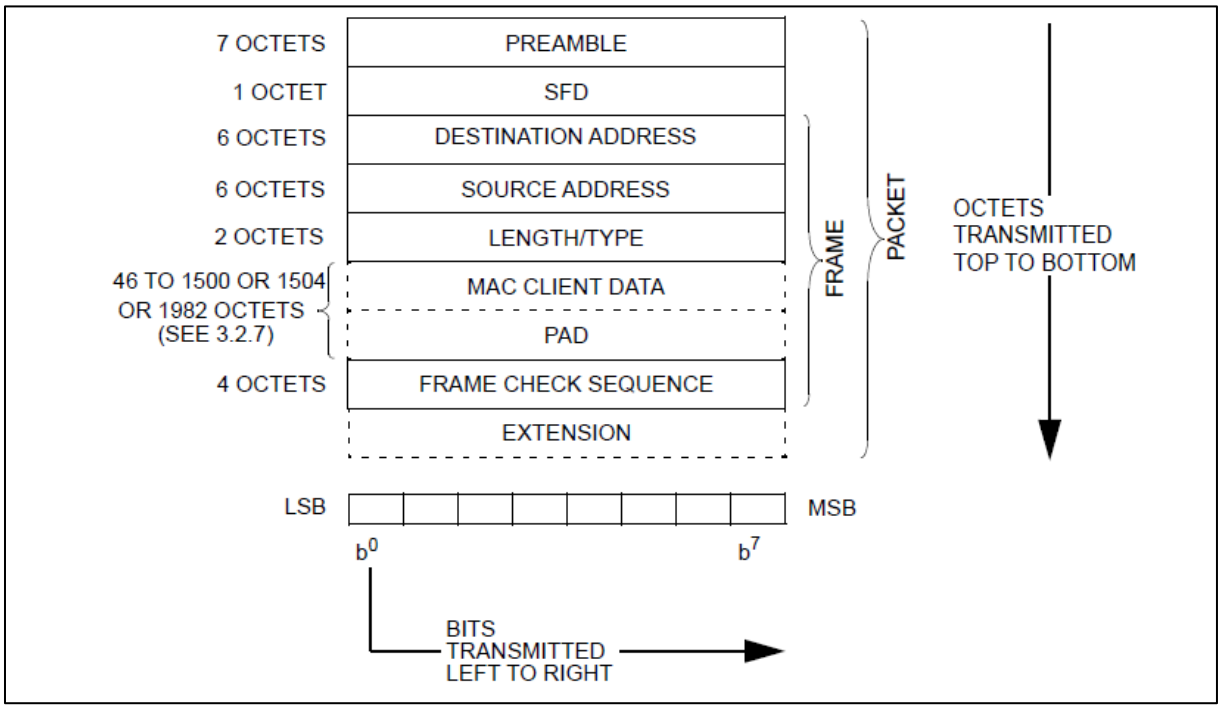

Figure 5: IEEE 802.3 packet and frame format [IEEE_8023_03, p. 118 et seq.]

To be able to send frames in between stations without errors, said frame additionally gets preceded by the '*Preamble*' and the '*Start Frame Delimiter*', which in its entirety is called a '*packet*'. [IEEE_8023_04, p. 98/118 et. seq.] Packets which are send in between stations are the result of the Media Access Control (MAC) frame by the Physical Layer (PHY) and are defined as a stream. [IEEE_8023_05, p. 105]

'*Frames*' are commonly used for describing data, that is transmitted at the '*data link layer*' (ISO/OSI Layer 2), whereas '*packets*' fulfill the same role at the '*network layer*' by additionally containing higher entity protocol functions residing inside the data field (ISO/OSI Layer 3). [SpuCha_03, p.31/302]

Note: In the context of this thesis, data is referred to using both terminologies '*packet*' and '*frame*'.

Layer 2 establishes the communication from station to station connected to the same network. The parts of the Ethernet standard that describe the frame format as well as the MAC ('*Media Access Control*') protocol belong to this layer. [IEEE_8021Q_01, p. 128 et seq.]

Layer 3 established the communication between stations across an internetwork, composed of number of interconnected network systems. [SpuCha_01, p. 17] Thus, this layer determines how data is

sent to the receiving device across multiple networks. It's responsible for packet forwarding, routing, and addressing by utilizing the IP address in addition to the MAC address of Layer 2. [Plixer_01]

Besides the stated separation of terminology of frame and packet according to the Ethernet standard one will often also find said terminologies for the intented purpose of the data contained inside the data filed of an Ethernet frame. Data within the data field which contains no higher entity functions (e.g. TCP/IP protocol) usually gets referred to as '*frame*' residing at the data link layer.

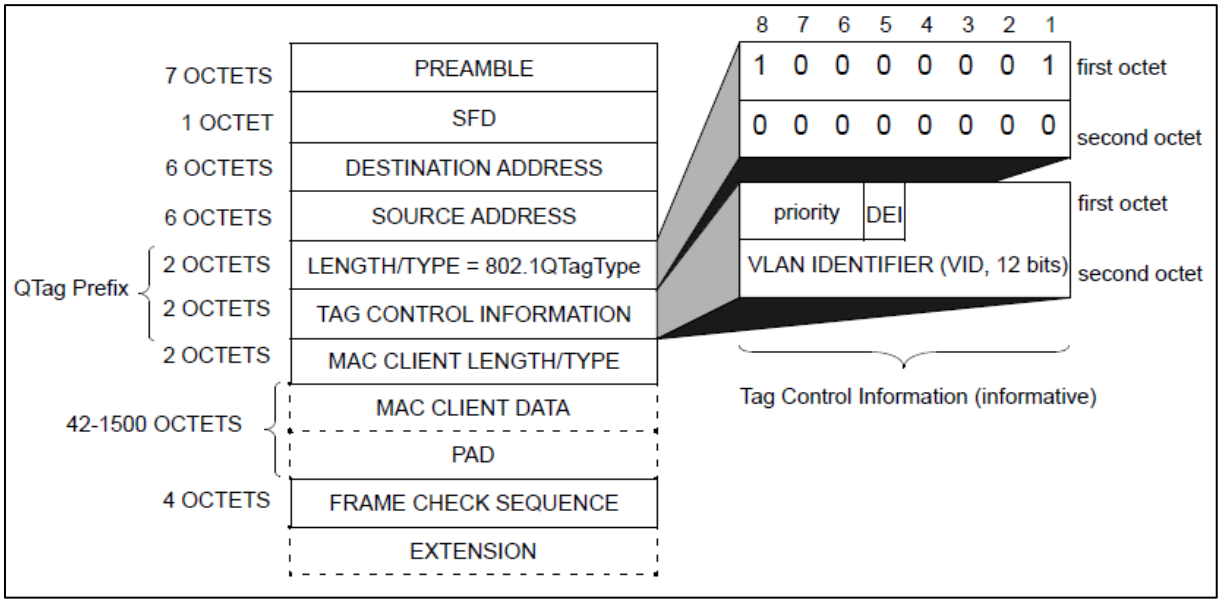

Figure 6: IEEE 802.1 basic frame with Q-Tag [IEEE_8021Q_02, p. 1907 et seq.]

Figure 6 shows the structure of a basic Ethernet frame with the added QTag Prefix above the MAC Client Data field. The illustrated frame is the common format used in automation networks.

The frame consists of multiple subsections, listed from top to bottom:

a) Preamble (7 Byte) – notifies the receiving station that a frame is starting and enables synchronization.

b) SFD ('*Start Frame Delimiter*', 1 Byte) – indicates the start of the destination MAC address.

c) Destination MAC (6 Byte) – identifies the receiving station.

d) Source MAC (6 Byte) – identifies the sending station.

e) QTag (VLAN) prefix (4 Byte) – carries the VLAN priority tag, handling packet prioritization specified by the IEEE 802.1Q.

f) Length/Ethertype (2 Byte) – defines the type of protocol inside the frame (e.g., IPv4 or IPv6).

g) Data and Pad (46 to 1500 Byte)) – contains the payload data. To fulfill the minimum required length of this filed padding data padding data can be added for transmission of smaller frames.

h) FCS ('Frame Check Sequence', 4 Byte) – contains the CRC ('Cyclic Redundancy Check') checksum sequence for detection of corrupted data.

i) IFG ('Inter Frame Gap', 12 Byte) – defines the minimum gap between the transmission of two subsequent frames.

Note: Packet-prioritization according to the VLAN/Q-tag plays an important role while handling incoming packets processed by the switch hardware.

For more information regarding packet priority, refer to chapter 4.2, 7.1 or A.3.2.

The total Ethernet frame size ranges from 64 to 1518 Byte. It includes the D-MAC (6 Byte), S-MAC (6 Byte), VLAN/Q-tag (4 Byte), Length/Ethertype (2 Byte) and the Data (46 to 1500 Byte) field.

For successful frame transmission PREAMB (7 Byte), SFD (1 Byte), FCS (4 Byte) and IFG (12 Byte) are added to the frame payload. Said transmission payload sums up to 24 Bytes but is not visible in the transferred frame length. Hence, the total frame length of one Ethernet frame ranges from 88 to 1542 Byte, resembling the packet payload one packet carries while travelling through the network.

Frame transmission (casting types)

The transmission of packets across the network is described below: [SpuCha_04, p.30 f. et seq.]

Ethernet uses a broadcast mechanism, where each frame that is transmitted on the data line is heard by every station sharing the transmission medium. Ethernet signals are transmitted from the Ethernet bridge port (PHY) of one station and sent over to every attached station. Due to the broadcast mechanism every station sharing the signal channel receives the same frame and reads the first bits of the signal and look at the second field, the destination address (D-MAC), of the frame. Said destination address of the frame then gets compared by the station with its 8 Byte source address (S-MAC), also called '*unicast address*', and any multicast address it has been enabled to recognize. If both addresses match the station continues to read the entire frame and delivers it to the networking software running inside that station. All other connected stations that do not have a matching address will stop reading the frame.

Besides unicasting the Ethernet delivery mechanism also supports multicasting. Instead of sending the same frames to multiple recipients, casting via a '*multicast address*' allows a single Ethernet frame to be received by a group of stations. Said multicast address simply gets added to the existing built-in unicast (physical) address of the station. This way a multicast group can be created. A single packet stream sent to the multicast address of a group will be received by all stations in that group. A multicast address containing only binary ones in its 6 Byte D-MAC address is defined as '*broadcast address*'. All Ethernet interfaces that see a frame transmitted with a broadcast address will read the frame and deliver it to the networking software running inside their station.

Protocols (high-level entities)

After looking at how frames are sent between stations, further insight shall be obtained about the data send in a frame: [SpuCha_04, p.30 f. et seq.]

Data exchanged between stations is carried in the data field of the Ethernet frame (see Figure 6) and is structured according to higher-level protocols. The high-level protocol information inside each frame is used to establish communication between applications running on stations connected to the network. Said high-level protocols, also referred to as high-level entities are independent of the Ethernet system. Ethernet LAN including its hardware and frames basically resemble a form of delivery for data being sent by applications using high-level protocols.

In other words, the Ethernet LAN doesn't care, which information it transmits according to the high-level protocol being carried in the data field of the Ethernet frame.

High-level protocol packets have their own addresses and data, embedded in the data field of the frame. This kind of arrangement is called '*encapsulation*'. Encapsulation enables independent systems, such as network protocols and Ethernet LANs, to work together. Using encapsulation, the Ethernet frame carries the network protocol packet by treating it as unknown data, placed into the data field of the Ethernet frame. Upon delivery of the Ethernet frame at the destination address, the network software running on the receiving station deals with the protocol packet extracted from the Ethernet's frame data field.

To get the protocol data inside an Ethernet frame to its intended receiving station, the high-level protocol software and the Ethernet system must interact to provide the correct destination address for the Ethernet frame. Therefore, each protocol packet uses its own mechanism to discover the Ethernet destination address of the station for which the packet is intended. Depending on the protocol in use said mechanism can vary.

Protocols using the IP ('*Internet Protocol*') are using the IP address to the discover the destination address of the receiving station in the network, to which the packet is intended to be transmitted. Examples for these protocol types are the connection-based TCP ('*Transmission Control Protocol*') and the non-connection-based UDP ('*User Datagram Protocol*'). TCP packets are sent in a unicast frame, whereas UDP packets use a multicast frame.

Note: The distinction between protocol types based on their connection-based behavior is another important factor. Whereas TCP operates in a connection-based manner, UDP on the other hand does not.

Connection-based protocols can discover packet information which was damaged or lost during the transmission between the sending and receiving station. Thereby, these protocols can retransmit any missing or corrupted information. Non-connection-based protocols, however, have no information about the status of its transmitted packets and thus cannot retransmit any packets if anything goes wrong.

Therefore, it is important to understand the connection characteristic of a protocol in terms of potential packet loss which is either recoverable or not.

However, when first trying to send an IP based protocol packet out to its intended receiver, the sending station is not aware of the IP address belonging to the receiving station. The ARP ('*Address Resolution Protocol*') protocol is used to discover the corresponding IP address of a station, requested via its MAC address. ARP packets are sent in a broadcast frame.

Aside from the use of IP based protocols, additional network protocols are used to fulfill applications needed, while operating an automation network. An example would be the operation of Fast Ethernet in combination with PROFINET as shown in Figure 1 and Figure 2. PROFINET has a wide variety of specific protocols such as DCP ('*Discovery and Configuration Protocol*') and RTC ('*Real Time Cyclic*'). For more information about PROFINET specific protocols, refer to [PNO_2712_01].

Note: When operating an automation network, the safe operation of time sensitive processes is a major factor.

For this reason, PROFINET protocol types are distinguished in RT ('*Real Time*') and NRT ('*Non Real Time*') applications.

RT applications usually control time sensitive processes and only allow a limited amount or even no delay, according to its respective conformance class. NRT applications, however, are not as time sensitive. (For more information about PROFINET conformance classes, refer to [PNO_8061_02, p. 61 et seq.])
Depending on the application type, packet prioritization plays a major role in ensuring the safe packet transmission of time sensitive data over its non-time sensitive counterpart.

For more information regarding packet priority, refer to chapter 4.2, 7.1 or A.3.2.

Thus, based on the protocol used for transmitting packet data via the Ethernet frame, there are different kinds of traffic, depending on the protocol's characteristics and applications.

Note: In the context of this thesis, the distinction of packet data, sent via its respective protocol type, is referred to using the term '*traffic type*'.

2.1.3. Ethernet switch

Switch architecture

The '*Ethernet Bridge*', more commonly known under the name '*Ethernet Switch*', is the basic building block for operating between network stations inside switched LANs.

Note: In the context of this thesis, '*bridge*' and '*switch*' are both used interchangeable to describe Ethernet bridges in the following chapters. For the rest of this thesis the term '*switch*' will be used.

'Ethernet switches perform their linking function by bridging Ethernet frames between Ethernet segments. To do this, they copy Ethernet frames from one switch port to another, based on the MAC addresses in the Ethernet frames.' [SpuCha_05, p.300]

An Ethernet switch consists of multiple hardware parts to ensure frame reception, processing and forwarding interconnecting different network stations with each other and ensuring data flow between them. An Ethernet switch therefore comprises at least one '*Bridge component*'. Said Bridge component includes the following parts: [IEEE_8021Q_03, p. 178 et seq.] See Figure 7.

- A '*MAC Relay*' entity,
- At least 2 ingress/egress ports (PHY)
- Higher-layer entities (i.e., STP ['*Spanning Tree Protocol*'])

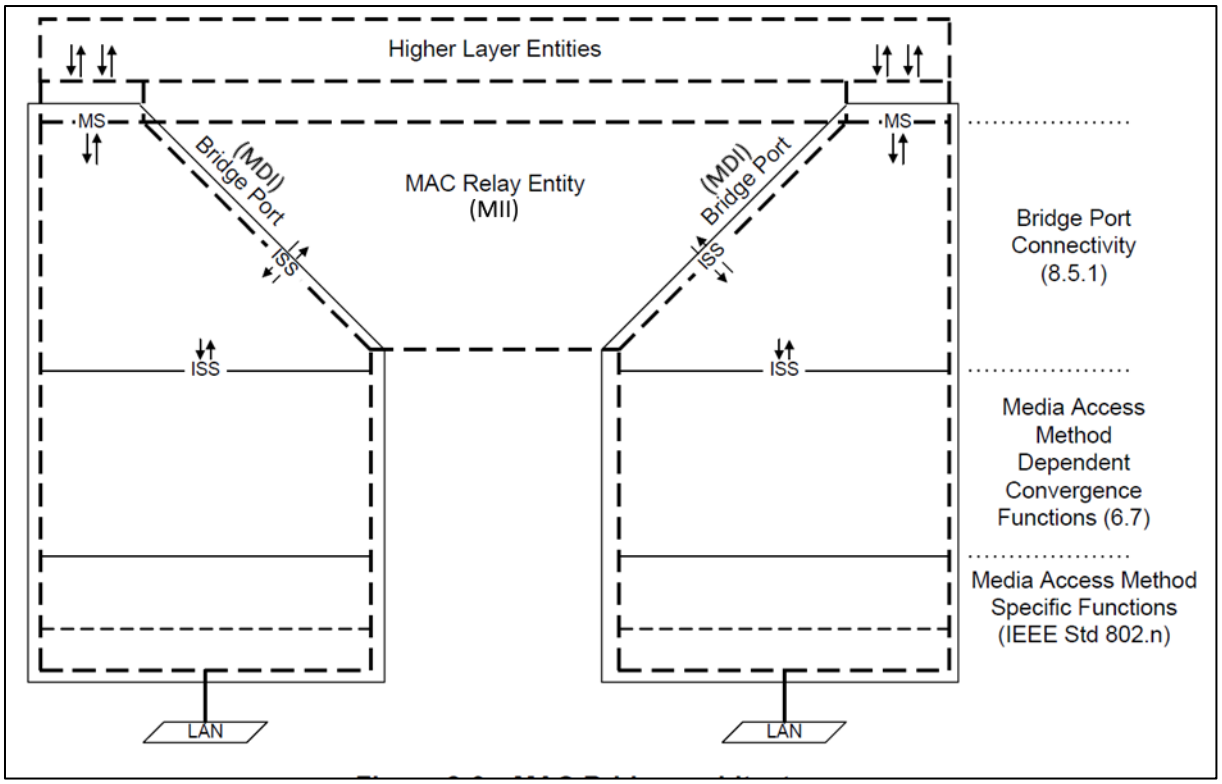

Figure 7: Ethernet Bridge architecture [IEEE_8021Q_04, p. 179 et seq.]

The '*MAC Relay*' of a Bridge, relays individual Ethernet frames between the separate MACs of the individual LAN stations, connected to its ports. In other words, the MAC Relay Entity is responsible for transmitting frames between the Bridge ports, which are connected to other stations of the network. Each station has its own MAC address, which is forwarded inside the Ethernet frame and used by the MAC Relay to correctly relay the frame between sender and receiver.

The Bridge ports, also called PHYs ('*Physical Layer*'), are responsible for the transmission of forwarded frames, processed by the MAC Relay. PHYs can have different types of media outputs (e.g., RJ45, SFP), therefore being classified as MDI ('*Media Dependent Interface*'). To ensure that the MAC Relay entity can access the PHY regardless of its media type, it utilizes an MII ('*Media Independent Interface*'). Thus, any type of PHY can be controlled by the MAC regardless of its used transmission medium. [SpuCha_06, p.109 et seq.]

Aside from the hardware specific entities at the physical layer, the Ethernet Bridge also possesses higher-level entities providing additional services aside from the general frame transmission. Said entities called LLC ('*Logical Link Control*') entities provide many services regarding the bridge management (e.g., prevention of forwarding loops, address learning, traffic filtering). [IEEE_8021Q_05, p. 180 et seq.]

Switch operation

As described in the previous section, an Ethernet switch links network stations with each other by relaying Ethernet frames between them. The processes that model the operation of a Bridge Port include: [IEEE_8021Q_05, p. 180 et seq.]

1) The Transmit and Receive Process…
   a) …receiving and transmitting frames from and to the attached LAN.
   b) …discarding frames on a received frame error, exceeding of limitation, based on filtering information or to preserve QoS ('*Quality of Service*') for other frames.

Note: The QoS describes the service availability of network operation based on frame information.

Higher prioritized packet data, which normally is essential to maintain proper network operation, is deemed more important than that with lower priority in terms of QoS.

Therefore, network operation according to QoS follows packet prioritization and prefers packet data carried by high-priority traffic, which in favor can mean the prolonged enqueuing or discarding of lower-priority frames.

c) …classifying received frames into VLANs, assigning each a VID value.

d) … selecting the traffic class and queuing of frames based on the VID.

Note: The VID ('*VLAN identifier*'), which is embedded into the '*Tag Control Information*' field of an Ethernet frame (see Figure 6 [chapter 2.1.2]), holds the entire VLAN/Q-tag information needed by the Bridge to perform frame transmission according packet prioritization.

For more information regarding packet priority, refer to chapter 4.2, 7.1 or A.3.2.

e) …delivering and accepting frames to and from the MAC Relay Entity and LLC Entities.

2) The Forwarding Process…

a) …enforcing loop-free active topologies for all frames.

Note: Ethernet by design requires, that only a single packet transmission path may exist between any two stations.

If not, the danger exists that switches with multiple interswitch connections can create loop paths in the network.

On said loop, packets will be forwarded endlessly, causing a packet overload over time. [SpuCha_04, p.30 f. et seq.] [SpuCha_07, p.308 et seq.]

b) …filtering frames using their VID and destination MAC addresses.

c) …forwarding received frames that are to be relayed to other Bridge Ports.

d) …observing and learning source addresses of frames received on each Port and updating the FDB ('*Forwarding Database*').

Note: To forward frames from one port to another the Ethernet switch needs to know the MAC addresses of its connected stations.

The switch does so by initial reading out the MAC address of all frames it sees on each port, running them in promiscuous mode. [IEEE_8023_06, p. 139 et seq.] Afterwards, each learned source address is saved in a table of source addresses, also called FDB.

By initially generating and subsequently maintaining the FDB, the switch knows, which station is connected to which port, based on the source address.

When a frame is received at the switch, it does not have to be send out to all connected ports, because the switch knows at which ports the sending and receiving station are located.

Therefore, each received frame is filtered by the switch based on the FDB to make packet forwarding decisions. [SpuCha_08, p.303 et seq.]

The process between receival and subsequent forwarding of a packet is handled by the internal packet processing hardware of the switch. Said hardware consists of several modules that ensure the correct

and secure transmission of packets. The following modules usually form the packet processing hardware of a switch ordered in the manner how packets pass through the switch:

1) Buffer: Storages all incoming packets at the port regardless of priority in a shared memory buffer for subsequent classification.
2) Classifier: Classifies storage packets based on various packet parameters (e.g., MAC, IP, VID) and assigns them to their designated queue.
3) Queues: Additional memory separated into multiple queues per port ranging from 0…4 or 0…7, based on the packet priority, responsible for enqueuing packets according to their priority, prior to their subsequent forwarding.
4) Scheduler: Schedules enqueued packets for subsequent forwarding based on various packet parameters (e.g., MAC, IP, VID) utilizing a scheduling algorithm (e.g., Weighted-round-robin)
5) Server: Forwards scheduled packets with given hardware delay

Note: The internal packet processing structure and function has been analyzed based on simulation model briefly described in chapter 3.

The processing of each single packet by the hardware takes time. Said time, also called PPT ('*Packet Processing Time*'), consists of multiple delays caused by the individual parts of the processing hardware. Figure 8 shows the breakdown of the individual delays.

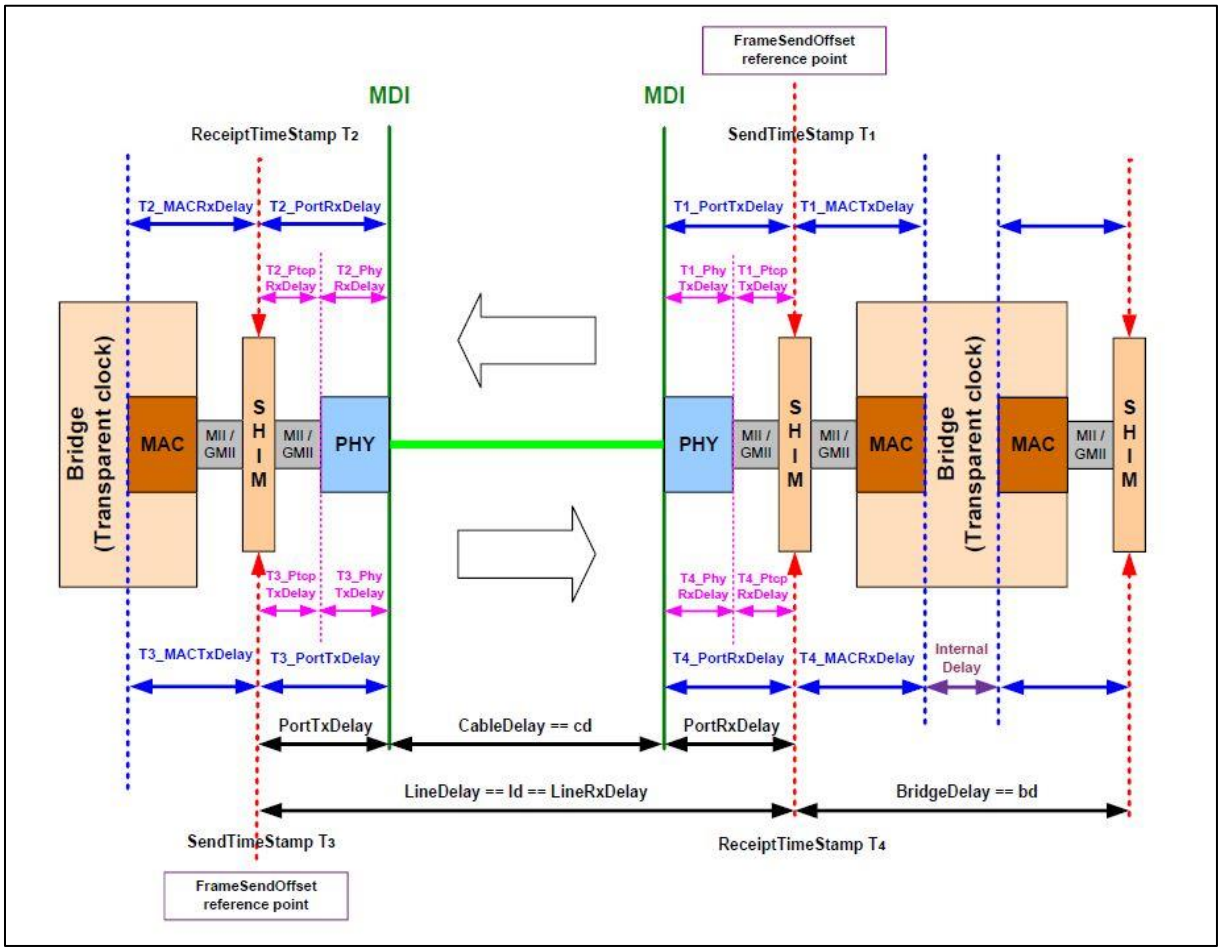

Figure 8: Ethernet Bridge hardware delays [PNO_2722_01, p. 163 et seq.]

a) '*Line delay*' describes the delay generated by transferring a packet from one station to another. It is the sum of '*Cable delay*' and '*Port(RX/TX) delay*'.

b) '*Bridge delay*' is the internal delay of the switch processing the received packet. It is the sum of '*MAC(RX/TX) delay*' and '*Internal delay*'.

Note: If the source port has a higher or equal transmission speed than the transmission port, cut-through delay is active. If the source port has a lower transmission speed than the transmission port, store & forward delay is active.

For simultaneous traffic bursts with multiple packets arriving at once at the designated egress port, only the packet arriving first gets forwarded via cut-through delay. All subsequent packets have to wait according to the first-in first-out (FIFO) principle and are therefore forwarded via store & forward delay.

c) '*Cable delay*' describes the datarate specific transmission delay of the cable transporting a packet from station to station depending on its size.

d) '*Propagation delay*' describes the length specific transmission delay of the cable transporting electric signals at approximately half the speed of light.

2.2. Communication in industrial environments

Communication in industrial based automation networks is implemented by utilizing established fieldbus transmission systems. While working with a '*mixed link speed*' network as shown in Figure 1 and Figure 2 two of said bus systems come into play.

2.2.1. PROFINET

PROFINET is an industrial Ethernet protocol. It is specified in [IEC_61158-5-10] and [IEC_61158-6-10].

2.2.2. Ethernet-APL

Ethernet-APL is another extension of the industrial Ethernet portfolio. Ethernet-APL is a new physical layer based on the Ethernet standard which is specified in [IEEE_8023CG_01] and [EAPL_PPS_01]. Ethernet-APL describes the transmission of data between two stations via a full-duplex SPE (Single Pair Ethernet) line. As such Ethernet-APL is a pure hardware-layer residing the first ISO/OSI layer which doesn't provide any higher layer functionality other than data transmission. More specifically Ethernet-APL is based on the '*10BASE-T1L*' Ethernet physical layer according to the '*IEE802.3cg*'. Said physical layer allows transmission rates of $10\ \mathrm{Mbit/s}$ over distances of $\leq 1000\ \mathrm{m}$.[EAPL__PPS_02, p.05]To utilize Ethernet-APL its SPE line gets implemented with APL capable devices. Both the transmission of pure data traffic by means of an APL field switch as well as the simultaneous power supply of connected network nodes via an APL power switch utilizing power over datalines (PoDL) are possible. [[EAPL_PPS_03, p.14] [IEEE_8023CG_02, p.86 et seq.]. Figure 1 and Figure 2 show the differences between industrial Ethernet (i.e. PROFINET) and Ethernet-APL.

Fast Ethernet cables used for PROFINET commonly uses Fast Ethernet line as standard. The Fast Ethernet cable (typically $100\ \mathrm{Mbit/s}$) consists of four wires, and communication is usually in full duplex mode at a maximum length of $100\ \mathrm{m}$ if copper medium is used.

The APL switch converts the $100\ \mathrm{Mbit/s}$ Fast Ethernet to the $10\ \mathrm{Mbit/s}$ Ethernet-APL signal. The physical connections of an APL switch are differentiated into trunks and spurs. The trunk is the '*main line*' of an APL network. It connects APL switches with each other, and comes both as unpowered and powered option. Unpowered trunks allow lengths up to $1.000\ m$. The maximum length of a powered trunk line depends on the number of supplied devices and the connection length between them. The

spur is used for connecting an APL switch with its APL field devices, which can be up to $200\ \mathrm{m}$ long. [PNO_EAPL_01]]

APL switches are divided into two different categories. APL power switches are used to deliver data traffic as well as the power supply for its subsequent connected APL field switches, which are supplied by a powered APL trunk line via PoDL. APL field switches are used for data traffic and can either be supplied auxiliary or with a powered APL trunk line. Comparing Figure 1 and Figure 2 shows that APL field switches can be operated at $100\ \mathrm{Mbit/s}$ Fast Ethernet or $10\ \mathrm{Mbit/s}$ Ethernet-APL depending on the switch's intended use of operation.

While being powered via an independent power supply the APL field switch can be operated at $100\ \mathrm{Mbit/s}$ utilizing Industrial Ethernet ports (refer to Figure 1). However if the field switch gets powered via a powered APL trunk line, APL trunk ports are used allowing transmission speeds up to 10 Mbit/s (refer to Figure 2).

'This concludes that Ethernet-APL is another physical layer for PROFINET. The user now has the choice of using the $100\ Mbit/s$ *Ethernet physical layer, an even higher transmission speed, or the* $10\ Mbit/s$ *Ethernet-APL physical layer.'* [PNO_8061_03, p. 124 et seq.]

3. Simulation test

The simulation test is intended to provide information as to how an Ethernet switch's internal packet process operates in detail. However, in the course of this thesis, focus has shifted from a simulation-based approach to hardware tests on real APL switches. Therefore, the following explanation is kept as short as possible and merely gives a general overview of the simulation approach.

The simulation-based test environment focuses on testing a simulated Ethernet switch at its physical layer (ISO/OSI Layer 1) as described in the assignment (see chapter 1.2). The test is based upon traffic generation and analysis via specific models generated inside the network simulation tool '*OMNet++*', along with its accompanied Framework '*INET*' for access to Ethernet applications. [Omnet_01] [Omnet_02] Modelling is achieved with modules of the INET '*Queuing Model*' library. [Omnet_03]

A more detailed understanding of each basic queuing module is given in the INET '*Queuing Tutorial*', with contains easy to understand practical examples. [Omnet_04] The structure, parameterization and measurement results of the simulation tests are described in detail in the presentation attached in the appendix (see chapter A.1).

4. Hardware test

As mentioned in the assignment of this thesis (see chapter 1.2), a test setup shall be defined to analyze the packet-processing behavior of real Ethernet-APL switches, representing the DUT ('*Device Under Test*'), operating in a '*mixed link speed*' network.

The purpose of the test setup is to determine if there is packet loss for system relevant data referring to time sensitive data such as real time traffic. In a real industrial network, such data loss could cause network problems and potentially jeopardize plant operation as a whole.

To assess the risk of packet loss, the hardware tests include various test scenarios aimed at finding answers to the following questions:

1) Are there any packet-throughput limits due to hardware restrictions (i.e., *BufferLength*, *QueueLength*, *DataRate* limit) that might cause packet discards if exceeded?
2) How does packet processing differ between simulation and hardware test, when testing mixed traffic in downstream direction and generating packet load based on the *'MinimumFrameMemory*' condition? (refer to chapter A.3.1)
3) How does the packet processing differ between simulation and hardware test, when testing with bursty traffic in upstream traffic direction and generating packet load based on the '*SimultaneousTrafficBurst*' condition? (refer to chapter A.4.1)
4) When gradually increasing netload exceeding the packet-throughput limits, what would be the threshold level at which packet discards happen also to prioritized packets?

Chapter 4 describes the overall testbed structure including the hardware and software used to perform the hardware tests.

Chapter 5 gives some insight into the hardware limitations of the DUTs, looking into total packet count and packet load processable by the APL switch hardware, as well as its connected data lines.

Chapter 6 shows how the best-practice '*mixed link speed*' network examples presented in chapter 1.1 can be turned into actual hardware testbed structures for performing hardware tests on the DUTs.

Chapter 7 summarizes the results of all measurements in a comprehensible and compact listed form.

Note: The detailed measurement results for the measurements conducted for this thesis have been moved to the appendix. There, each measurement is analyzed in detail, describing the packet-processing behavior of the tested APL switches figure by figure.

However, this level of detail is not necessary for understanding the essence of the measurement results and serves only for the understanding of the measurement evaluation.

4.1. Hardware test – Testbed overview

The hardware-based test environment revolves around evaluating a real Ethernet-APL switch as DUT. The test is based upon network traffic generation using the frame generator tool '*Ostinato*' in combination with traffic analysis using the frame measurement tool '*WireShark*'. [Osti_01] [WireSh_01] The measurement and analysis of the incoming and outgoing packets with the switch in between gives an insight into its packet-throughput behavior in the presence of network congestion. Figure *9* gives an overview of the hardware testbed that allows measurement and analysis of packet-throughput of the DUT.

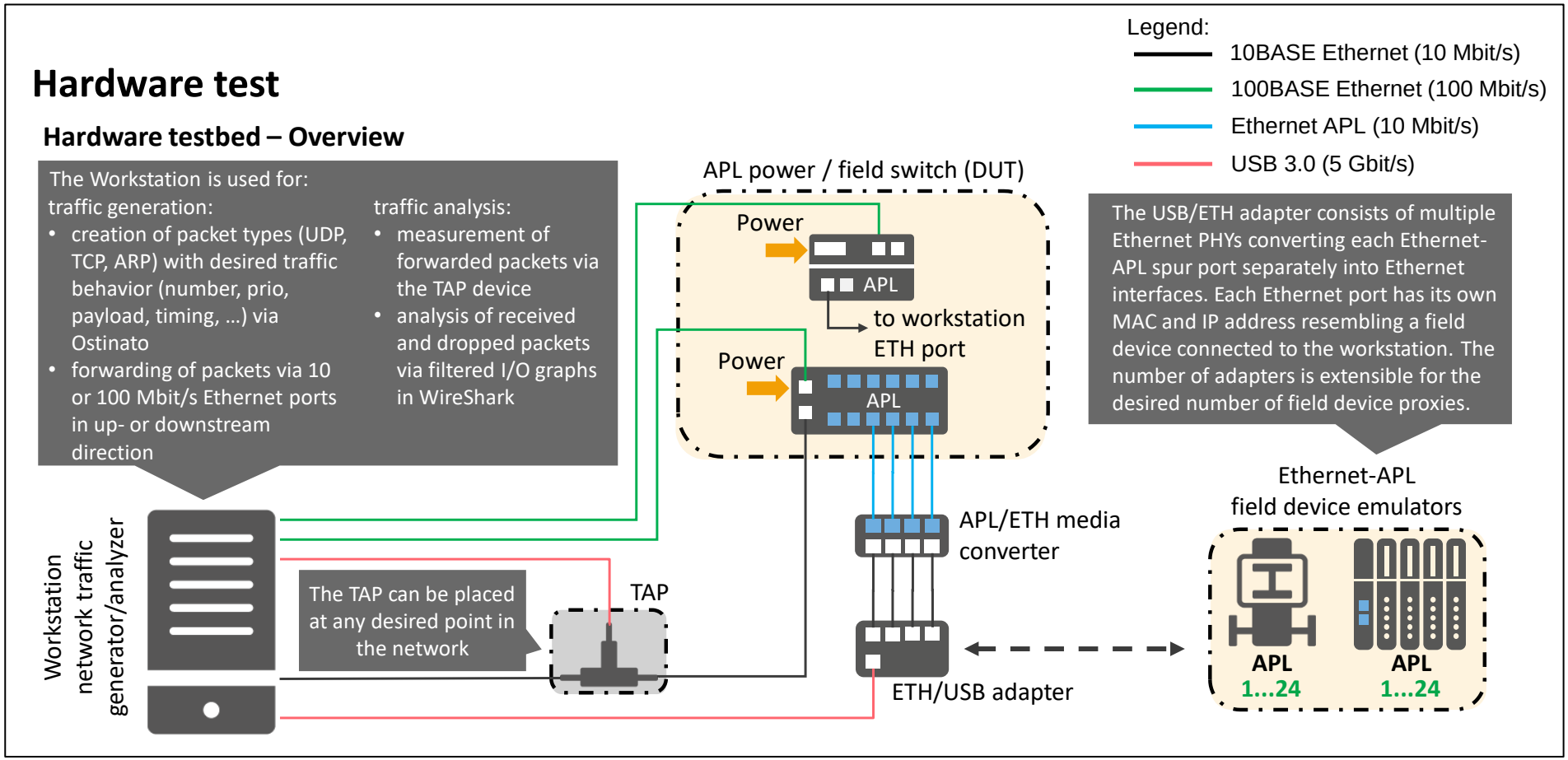

Figure 9: Hardware testbed for congestion loss-analysis (overview)

A workstation is used for traffic generation and traffic analysis. It can create different packet types in order to create the desired traffic scenarios and is also used for analysis of captured traffic via the TAP (‘*Test Access Point*’) device. The TAP device allows non-intrusive real-time measurements of the Ethernet traffic with datarates of up to $100\ \mathrm{Mbit/s}$. It can be placed at any desired place in the network for this purpose.

The APL switch used for testing can be directly connected to the workstation via its Fast Ethernet ports working at a user defined datarate of either 10 or $100\ \mathrm{Mbit/s}$.

Note: APL switches can utilize Fast Ethernet as well as Ethernet-APL trunk ports for data transmission in between network stations.

Fast Ethernet ports have the advantage of a higher data transmission commonly working at 100 Mbit/s instead of 10 Mbit/s as it is the case with an Ethernet-APL trunk port. However, using a powered Ethernet-APL trunk port substitutes the need of additional auxiliary power supplies for each APL switch connected to the APL trunk.

In case of the hardware testbed, the APL switches which have been provided for testing by 2 independent manufacturers are using Fast Ethernet ports for data transmission between stations and Ethernet-APL spur ports for the connection of APL field devices.

Regardless of the port type used, the difference in packet-throughput behavior of an APL switch is only fixed by its port datarate. Therefore, all measurement results based on the testbed structures presented in chapter 6 are viable by utilizing the correct port datarate setting regardless of the actual port type used.

During the runtime of this thesis, no real Ethernet-APL field devices have been provided for testing. To emulate the use of APL field devices, the workstation is connected to the Ethernet-APL egress spur ports of the APL switch. To connect the Ethernet-APL spur ports of the switch to ‘*regular*’ Ethernet ports of the workstation, an Ethernet/USB-Adapter in combination with an APL/ETH media converter are used. Six Ethernet/USB-Adapters are used in total. One ETH/USB adapter contains up to four separate Ethernet-PHYs (‘*Physical Layer*’), each having its own MAC and IP address while being connected to the workstation. Thus, up to 24 field devices can be emulated via connection of the APL switch spur ports to the workstation at a datarate of $10\ \mathrm{Mbit/s}$.

4.2. Hardware test – Test software

Table 1 shows the software tools used inside the hardware testbed for traffic generation, measurement, and analysis.

Table 1: Hardware testbed, software tools

 Netload generator	**IPv4 traffic generation using '*Ostinato*':** [Osti_01] ➢ Generation of desired packet types (RTC, DCP, ARP, OPC UA) manually or via import of existing packet capture file (PCAP) ➢ Each packet type called '*stream*' can be freely modified (MAC, IP, packet type [Length/EtherType], packet priority [VLAN], packet load, packet count, etc.)
 Netload monitor +  Netload analysis	**IPv4 traffic measurement using a Test Access Point (TAP) device (ProfiShark1G) in combination with network monitoring software '*WireShark*':** [ProTAP_01] [WireSh_01] ➢ Real-time measurement of received packets ➢ 2x ETH (1 Gbit/s) + 1x USB 3.0 (5 Gbit/s) interface ➢ 8 ns timestamp resolution ➢ Non-intrusive, transparent device (does not appear inside the analyzed subnetwork) ➢ Plug-in device to workstation via USB ➢ Separate network entity with its own MAC and IP address visible in WireShark ➢ Traffic analysis via PCAP capture-file directly generated in WireShark (e.g., via filtered conversation statistic, I/O Graphs, etc.)

Traffic generation tool: Ostinato is a frame-based traffic generator tool for emulating packet-load in network structures. It enables e.g. the generation of ARP and IP-based packet streams (UDP, TCP, ICMP). Additionally, the use of additional user defined packet types is possible by importing already existing PCAP files. Each packet stream can be modified in number, size and timing of packets to achieve the desired traffic scenario.

Note: Due to Ostinato being an IPv4/6-based frame generator, the traffic types used in the test are UDP, TCP and ARP packets. Alternatively, the use of PROFINET specific types such as RTC and DCP packets by importing them via a WireShark record file (PCAP) can be an additional option.

The analysis of the packet-throughput at the switches using WireShark is limited to the network layer (ISO/OSI layer 3, MAC+IP address). Thus, only the packet size and the number of packets from the respective sender/receiver are examined within the scope of the analysis and no higher-level protocol-specific properties, which only apply from Layer 4 and higher.

In other words, it is not relevant for the test results which task each specific protocol performs at higher service layers, but rather which casting type (unicast, multicast, broadcast), packet load, packet count and packet priority condition it holds.

The IEEE 802.1Q standard specifies up to 8 different traffic types that are divided into classes ranging from 0…7. [IEEE_8021Q_06, p. 1918 et seq.] The priority of each traffic type is not static and can be

freely defined by the user. However, common user priorities for PROFINET specific protocols are '6' for RTC traffic and '0' for IP (RPC via UDP) and DCP traffic. [PigRa_02, p. 64 et seq.] The user priorities '1…4' are commonly not used for PROFINET specific traffic. Additionally, IP traffic via OPC UA / TCP also has no fixed user priority and can be freely defined by the user.

Therefore, RTC ('*Real Time Cyclic*') unicast packets are emulated by UDP ('*User Data Protocol*') unicast packets and are set to user priority '6'.

DCP ('*Discovery and Configuration Protocol*') multicast packets are emulated by ARP ('*Address Resolution Protocol*') broadcast packets and are set to user priority '0'.

TCP ('*Transmission Control Protocol*') unicast packets are used in both standard and PROFINET networks and are set to user priority '4'.

For a more detailed description of the traffic parameters, refer to the measurement result summaries in chapter 7.

Traffic measurement tool: The TAP device enables real-time measurement of forwarded packets in WireShark PCAP files. The capture feeds can be monitored live or on-demand via WireShark.

Traffic analysis tool: WireShark allows capture and analysis of packet streams from all Ethernet interfaces connected to the workstation that the software is installed on. WireShark captures all frame-based packet types in a PCAP format and allows detailed traffic analysis via numerous filtering options as well as their visual representation in table or graph format.

5. Hardware test – DUT limitations & hardware delays

The internal switch hardware and its periphery is limited in its packet processing capacity and processing time. Exceeding these limitations can lead to packet loss and unstable packet forwarding behavior. Therefore, it is necessary to understand where such limitations exist in order to take them into account when trying to analyze packet processing behavior of the switch hardware accurately. Within the scope of this work, 2 ethernet switches A and B from different manufacturers are examined, whose hardware limitations are described in more detail below.

5.1. APL switch limitations, Manufacturer A

The internal packet processing of Manufacturer A is determined by two major factors. First, the packets arriving at the ingress port of the switch are stored in the internal packet buffer of the switch, which has a limited amount of storage. After that, the packets are classified and allocated to their respective internal packet queues which also have a limited amount of storage. Based on their priority, the packets inside the queues are then forwarded through the egress port.

Note: For more information regarding switch packet processing, see chapter 2.1.3 and 3.

The internal limitations of the APL switch from Manufacturer A are shown below.

5.1.1. 'bufferLength' limit, Manufacturer A

According to the manufacturer A, the total packet storage capacity is managed via one packet buffer, which gets shared between all of its managed Ethernet bridge ports (PHYs). It provides a combined storage capacity of 1800 packets, regardless of size.

This value has been verified by a measurement shown in Figure 10. The resulting '*bufferLength*' limit calculates as follows:

$$bufferLength_{\mathrm{A}} = 1.800\ x\ packet_{\mathrm{size}} \tag{5}$$

$$bufferLength_{\mathrm{A,min}} = bufferLength_{\mathrm{A}}\ x\ packet_{\mathrm{size,min}} \tag{6}$$

$$bufferLength_{\mathrm{A,min}} = 1.800\ x\ 64\ Byte\ = 115.200\ Byte \cong 112{,}5\ kByte \tag{7}$$

$$bufferLength_{\mathrm{A,max}} = bufferLength_{\mathrm{A}}\ x\ packet_{\mathrm{size,max}} \tag{8}$$

$$bufferLength_{\mathrm{A,max}} = 1.800\ x\ 1518\ Byte = 2.732.400\ Byte \cong 2{,}67\ MByte \tag{9}$$

5.1.2. 'queueLength' limit, Manufacturer A

According to the manufacturer, egress packet queue of the switch can handle up to 128 packets, regardless of size. This value has been verified by a measurement shown in Figure 11. The resulting '*queueLength*' limit calculates as follows:

$$queueLength_{\mathrm{A}} = 128 \cdot packet_{\mathrm{size}} \tag{10}$$

$$queueLength_{\mathrm{min}} = queueLength_{\mathrm{A}} \cdot packet_{\mathrm{size,min}} \tag{11}$$

$$queueLength_{\mathrm{min}} = 128 \cdot 64\ Byte = 8.192\ Byte = 8\ kByte \tag{12}$$

$$queueLength_{\mathrm{max}} = queueLength_{\mathrm{A}} \cdot packet_{\mathrm{size,max}} \tag{13}$$

$$queueLength_{\mathrm{max}} = 128 \cdot 1.518\ Byte = 194.304\ Byte = 189{,}75\ kByte \tag{14}$$

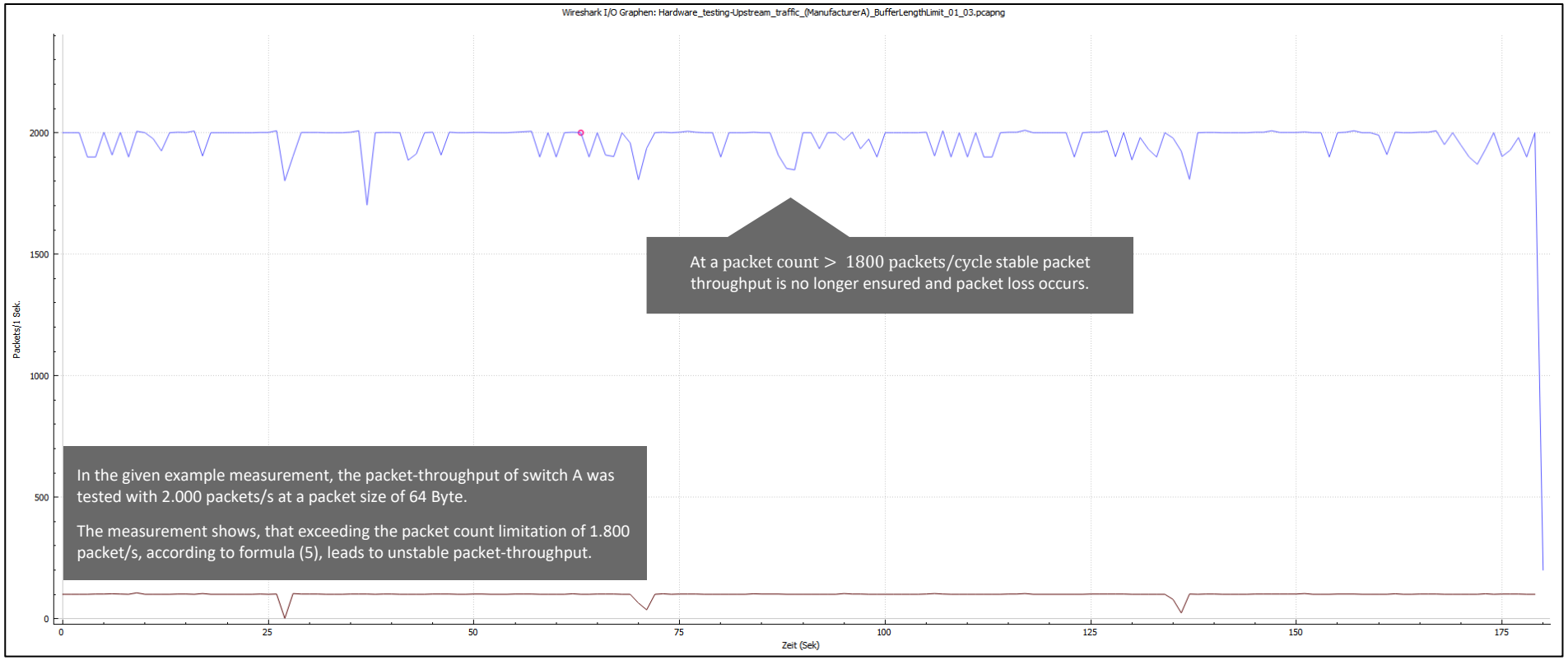

Figure 10: Packet loss due to exceeding '*bufferLength*' limit of the APL switch (Manufacturer A)

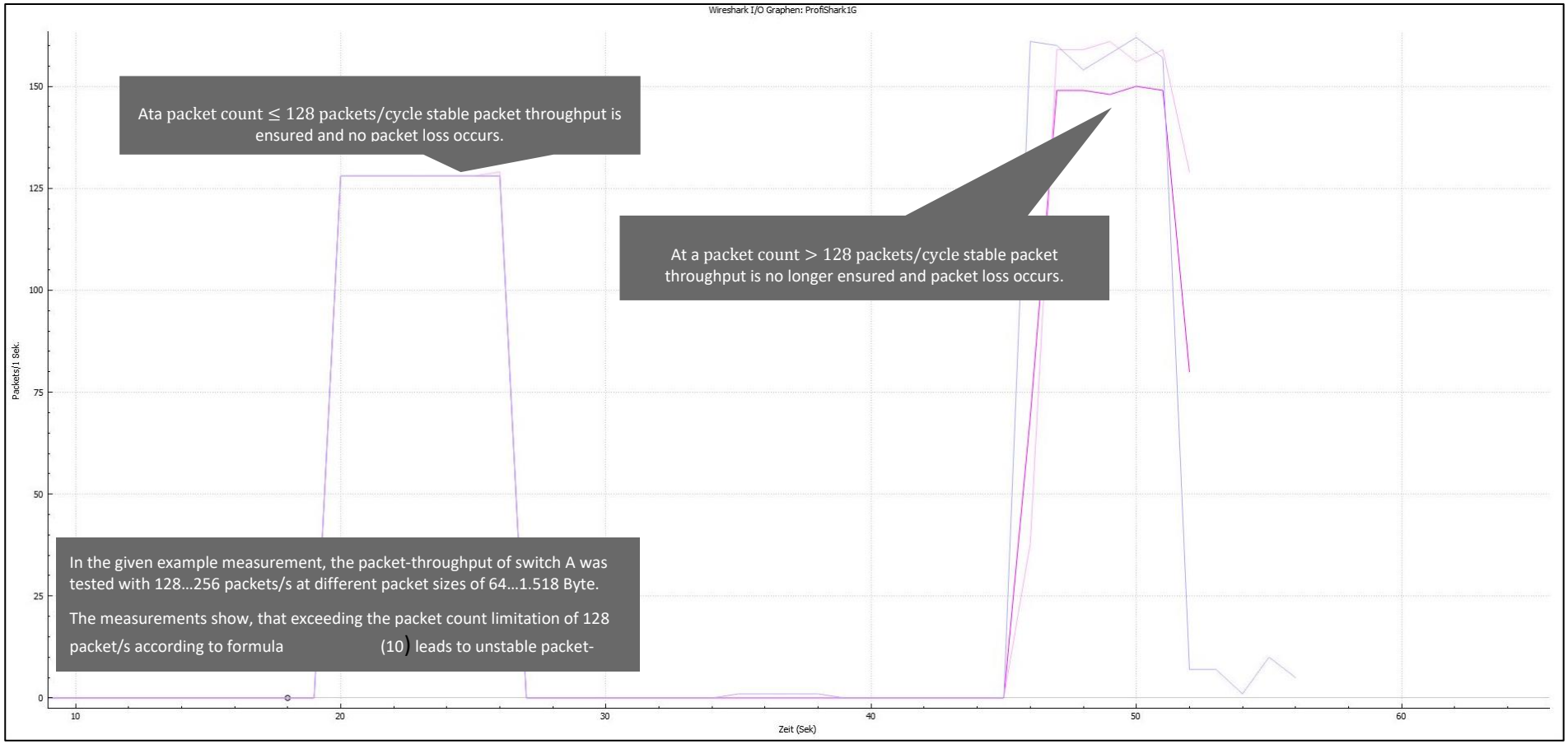

Figure 11: Packet loss due to exceeding '*queueLength*' limit of the APL switch (Manufacturer A)

5.2. APL switch limitations, Manufacturer B

The internal packet processing of Manufacturer B is fairly similar to that of Manufacturer A, however, the order in which packets are processed is different. With the switch from manufacturer B, the packets arriving at the ingress port of the switch are first classified based on their protocol casting-type (uni-, multi-, broadcast) as well as their designated user priority (VLAN tag). After that, the classified packets get allocated to their reserved buffer storage, which has a limited size. Based on their priority, the packets in the buffer allocations are subsequently forwarded through the egress port.

Note: For more information regarding switch packet processing, see chapter 2.1.3 and 3.

In the following, the internal limitations of the APL switch from Manufacturer B are discussed in some more detail.

5.2.1. 'bufferCount' limit, Manufacturer B

According to the manufacturer B, the total packet storage capacity is split into separate packet buffers for each managed Ethernet bridge port (PHY). Each has a maximum storage capacity of 1.024 packets.. Additionally, the total buffer size of 1.024 packets is limited to 256 Bytes per packet. This value has been verified by a measurement shown in Figure 12.

In addition, the storage capacity of up to 1.024 packets for each packet buffer, depends on the traffic type. Each traffic type has a specific number of reserved packets which can be stored inside a packet buffer, whereas higher prioritized packets get more reservation than others:

- Max. 512 packets for broad- and multicasting
- Max. 896 packets for unicasting
- Max. 1.024 packets for PROFINET RTC

The resulting '*bufferCount*' limit calculates as follows:

$$bufferCount_{\mathrm{B}} = 1.024 \cdot packet_{\mathrm{size}} \tag{15}$$

$$bufferCount_{\mathrm{B,min}} = bufferCount_{\mathrm{B}} \cdot packet_{\mathrm{size,min}} \tag{16}$$

$$bufferCount_{\mathrm{B,min}} = 1.024 \cdot 64\ Byte\ = 65.536\ Byte \cong 64\ kByte \tag{17}$$

$$bufferCount_{\mathrm{B,max}} = bufferCount_{\mathrm{B}} \cdot packet_{\mathrm{size,max}} \tag{18}$$

$$bufferCount_{\mathrm{B,max}} = 1.024 \cdot 256\ Byte\ = 262.144\ Byte \cong 256\ kByte \tag{19}$$

5.3. APL switch line limitation, Manufacturer A&B

Apart from its internal limitations, the switch is also limited in terms of its external periphery, namely the port connections which are connected via APL-spur or APL-trunk line. The following limitations exist for APL-lines in general and thus apply to the switches from both Manufacturer A and B.

5.3.1. 'dataRate' limit, Manufacturer A&B

The APL-trunk/spur line can handle traffic of 10 Mbit/s ≅1,19 MByte/s (see Figure 13).

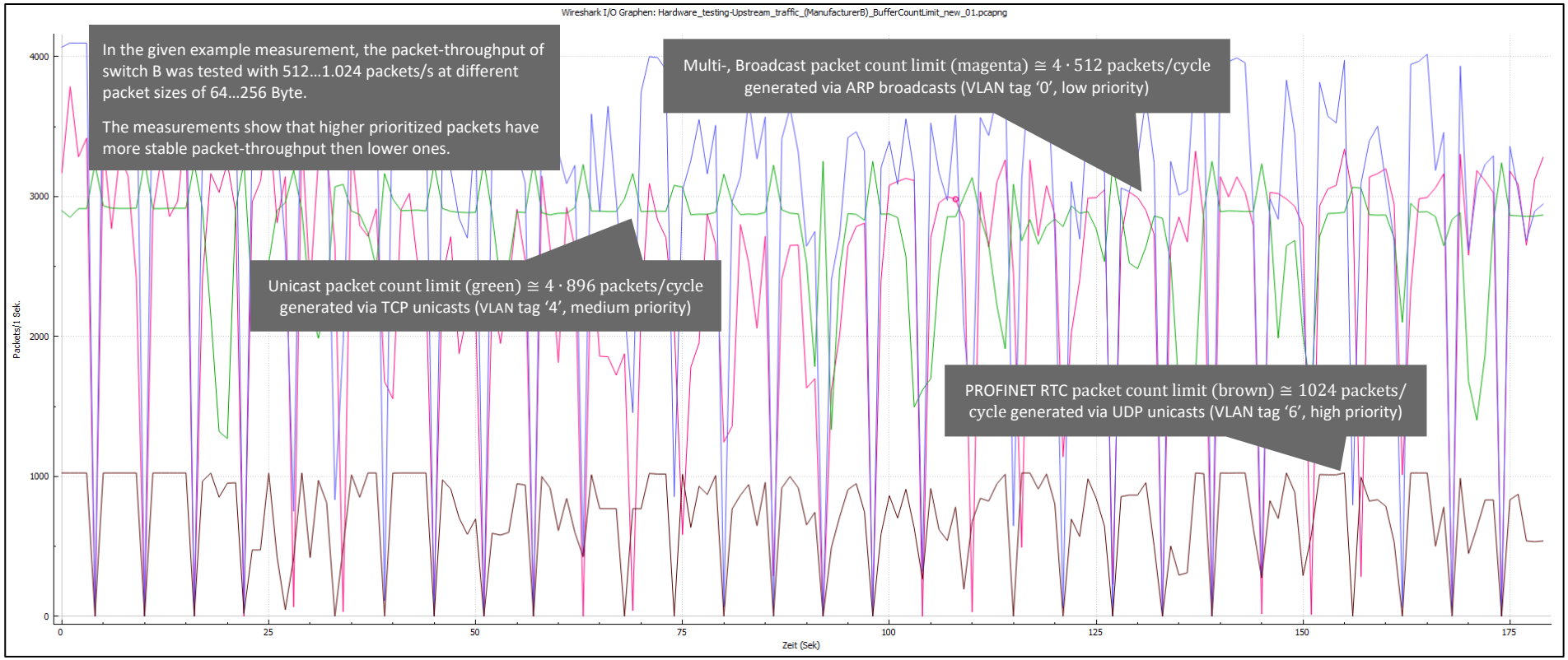

Figure 12: Packet loss due to exceeding '*bufferCount*' limit of the APL switch (Manufacturer B)

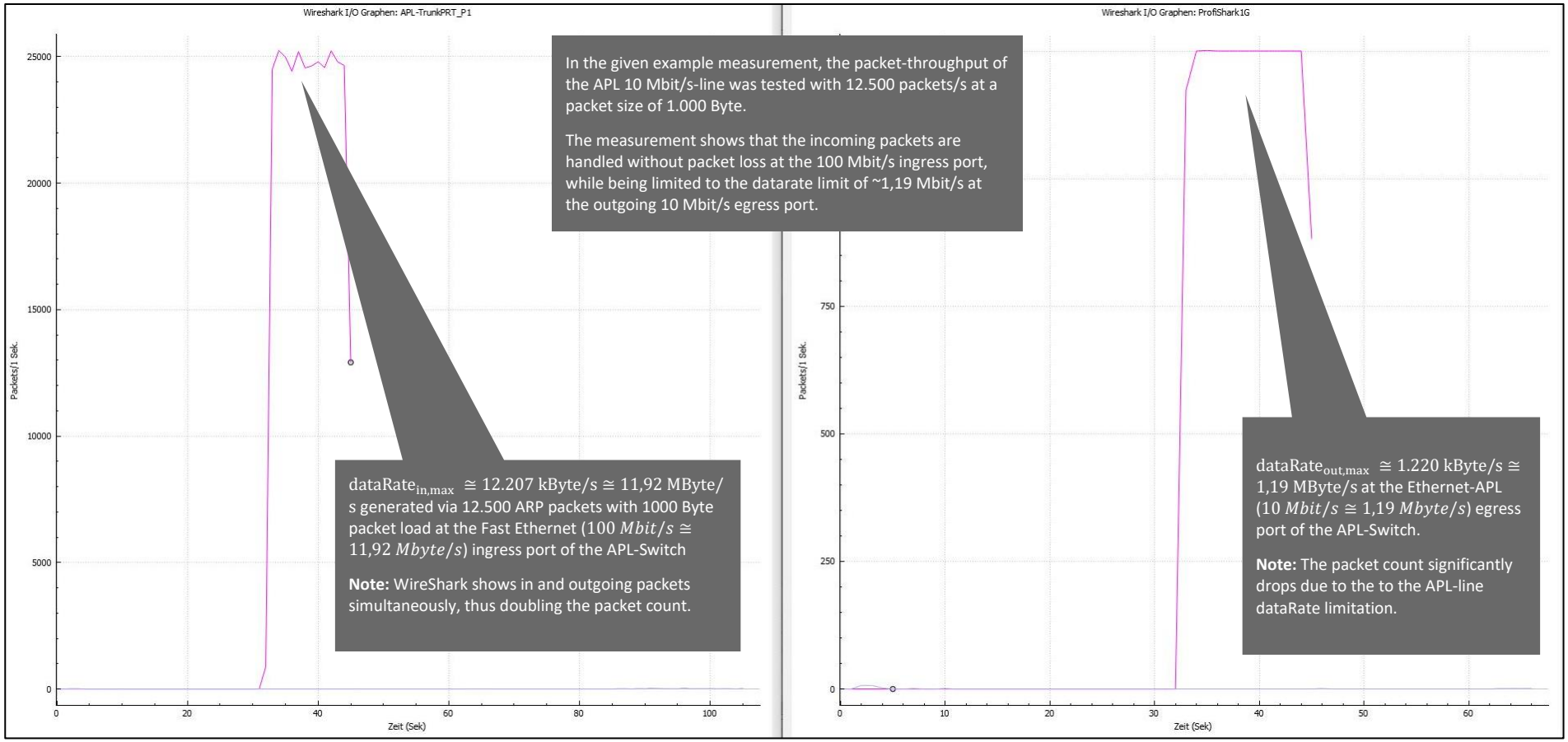

Figure 13: Packet loss due to exceeding '*dataRate*' limit of the APL switch 10 Mbit/s trunk/spur line (Manufacturer A&B)

5.4. APL switch hardware delays, Manufacturers A&B

According to the manufacturers, the following delays are apparent at the switch packet processing hardware: (see chapter 2.1.3)

Table 2: delay types of the packet processing hardware

delay type	value
bridge delay	a) $t_{\text{bridge,cut-through}}$: 4,5 µs (100 Mbit/s), 45 µs (10 Mbit/s) b) $t_{\text{bridge,store\&forward}}$: 6 … 125 µs (100 Mbit/s), 60 ….1250 µs (10 Mbit/s)
port delay	a) $t_{\text{port,RX}}$: < 3.2 µs b) $t_{\text{port,TX}}$: < 1,8 µs
cable delay	$t_{\text{cable}} = \frac{packet_{\text{size}}}{10\ Mbit/s}$ (20) *a)* $t_{\text{cable}}(packet_{\text{size,min}})$: 7,04 µs (100 $Mbit/s$), 70,4 µs (10 $Mbit/s$) b) $t_{\text{cable}}(packet_{\text{size,max}})$: 1,2336 ms (100 Mbit/s), 0,12336 ms (10 Mbit/s) $packet_{\text{size,min}} = 88\ Byte$ (see chapter 2.1.2) $packet_{\text{size,max}} = 1542\ Byte$ (see chapter 2.1.2)
propagation delay	$t_{\text{prop}} = \frac{l}{c_0/_2}$ (21) a) $t_{\text{prop}}(l_{\text{APL,trunk,max}})$: < 6.67 µs *b)* $t_{\text{prop}}(l_{\text{APL,spur,max}})$ < 1.33 µs $l_{\text{APL,trunk,max}} = 1000\ m\ per\ segment$ (see chapter 2.2.2) $l_{\text{APL,spur,max}} = 200\ m\ per\ segment$ (see chapter 2.2.2)

6. Hardware test –Testbed structures

The hardware measurements follow two separate best-practice examples for a typical mixed-speed network structure. This structure is based on a combination of industrial Ethernet with Ethernet-APL as shown in Figure 13 and Figure 14.

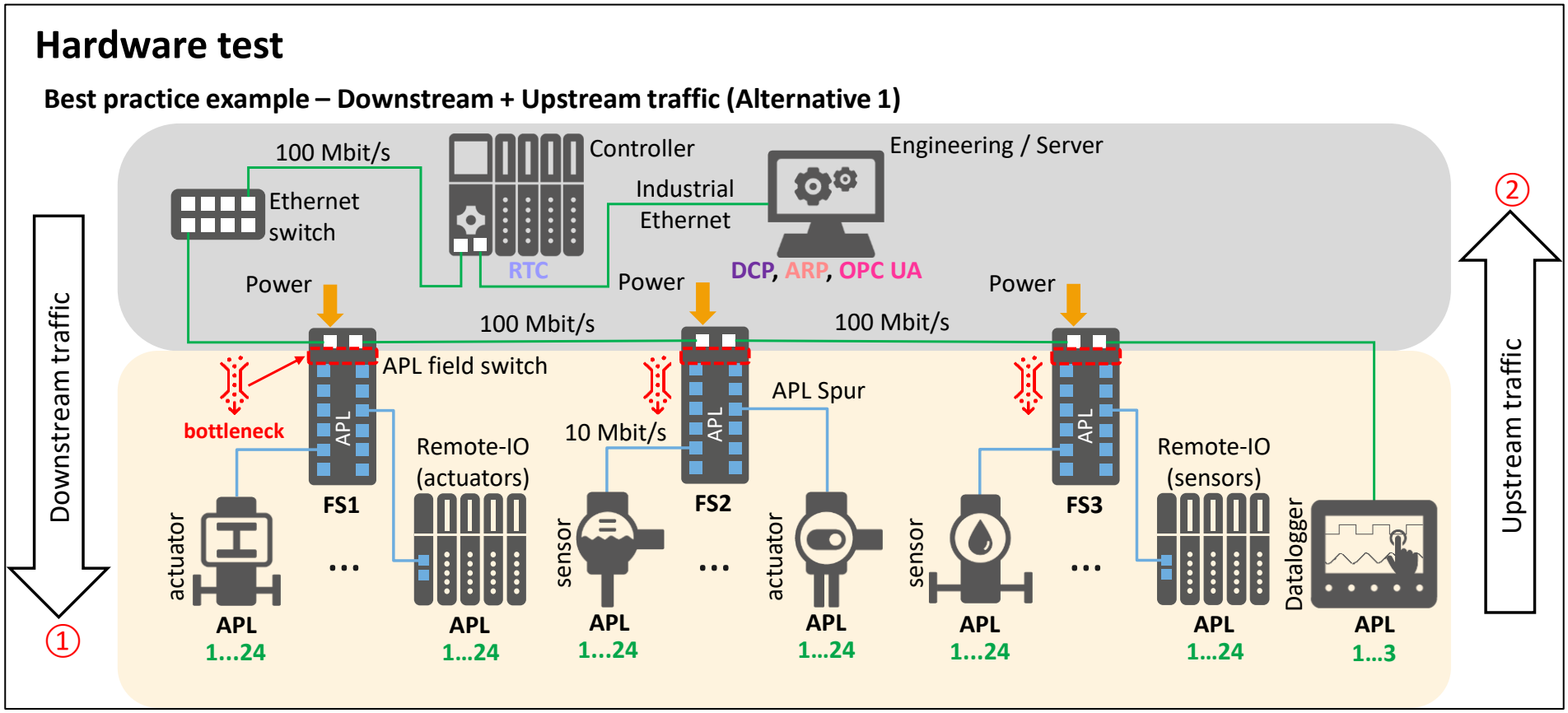

Figure 14: Mixed speed network including APL switches with industrial Ethernet (100 Mbits/s) connection (derived from [NieK_EAPL_01, p.57 et seq.])

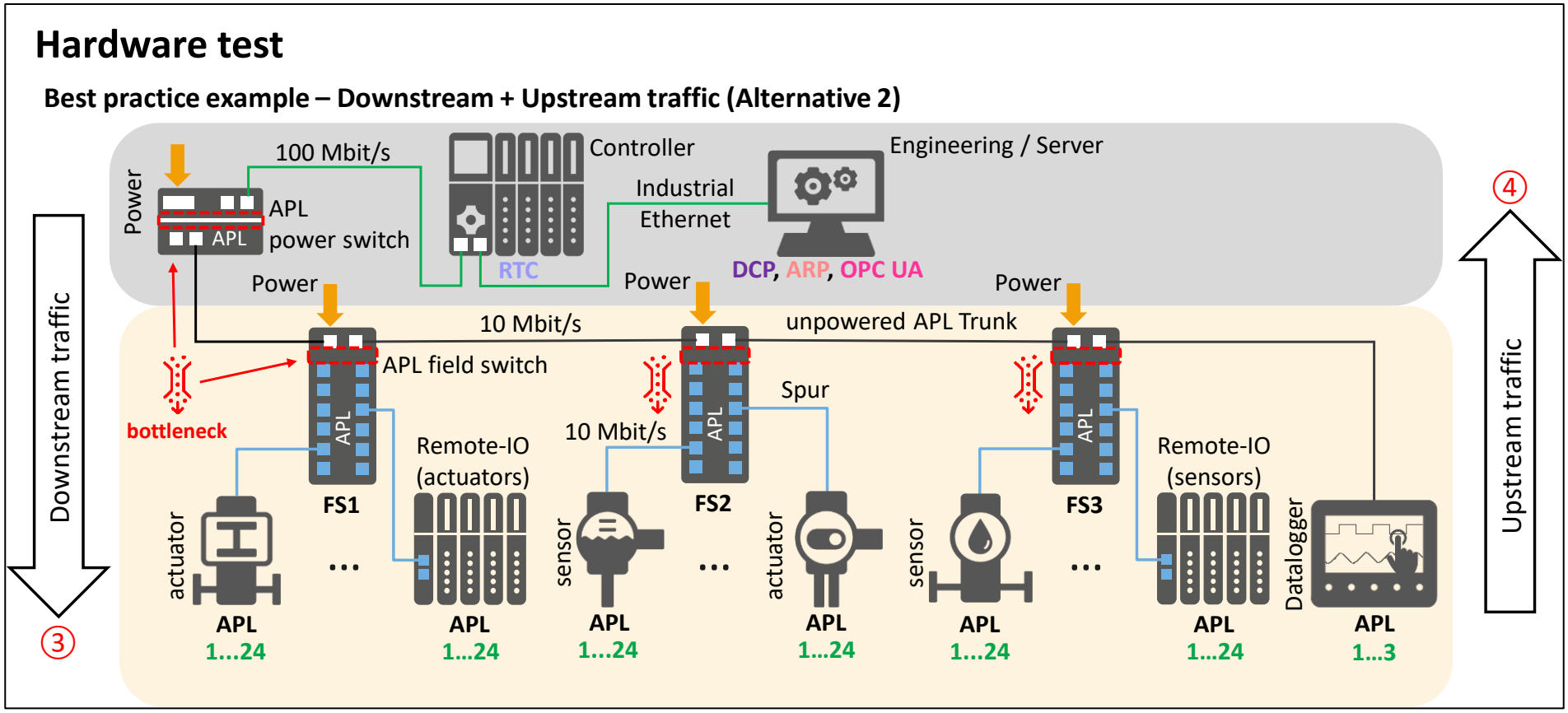

Figure 15: Mixed speed network including APL switches with Ethernet-APL (10 Mbit/s) connection (derived from [NieK_EAPL_01, p.57 et seq.])

Both network structures shown in Figure 14 and Figure 15 are common examples for the typical use of Ethernet-APL in combination with Fast industrial Ethernet. By combination of these physical layers, a unified Ethernet structure can be established, starting from the controller level and going all the way down to the field device level. When connecting the field device level that is using Ethernet-APL at 10 Mbit/s with the controller level that is using Fast industrial Ethernet at 100 Mbit/s, a bottleneck problem arises at the APL power and/or APL field switches which have to handle the transmission speed change.

For testing a mixed-speed network structure such as this, the test setup is divided into two separate setups for accurately analyzing the bottleneck in the downstream and upstream traffic direction respectively.

6.1. Hardware testbed - Downstream traffic analysis (Alternative 1)

Figure *16* shows the setup for accurate downstream traffic analysis. It is based on the network structure shown in Figure 14 (chapter 6), which in turn is based on the more general hardware test setup shown in Figure *9* (chapter 4.1).

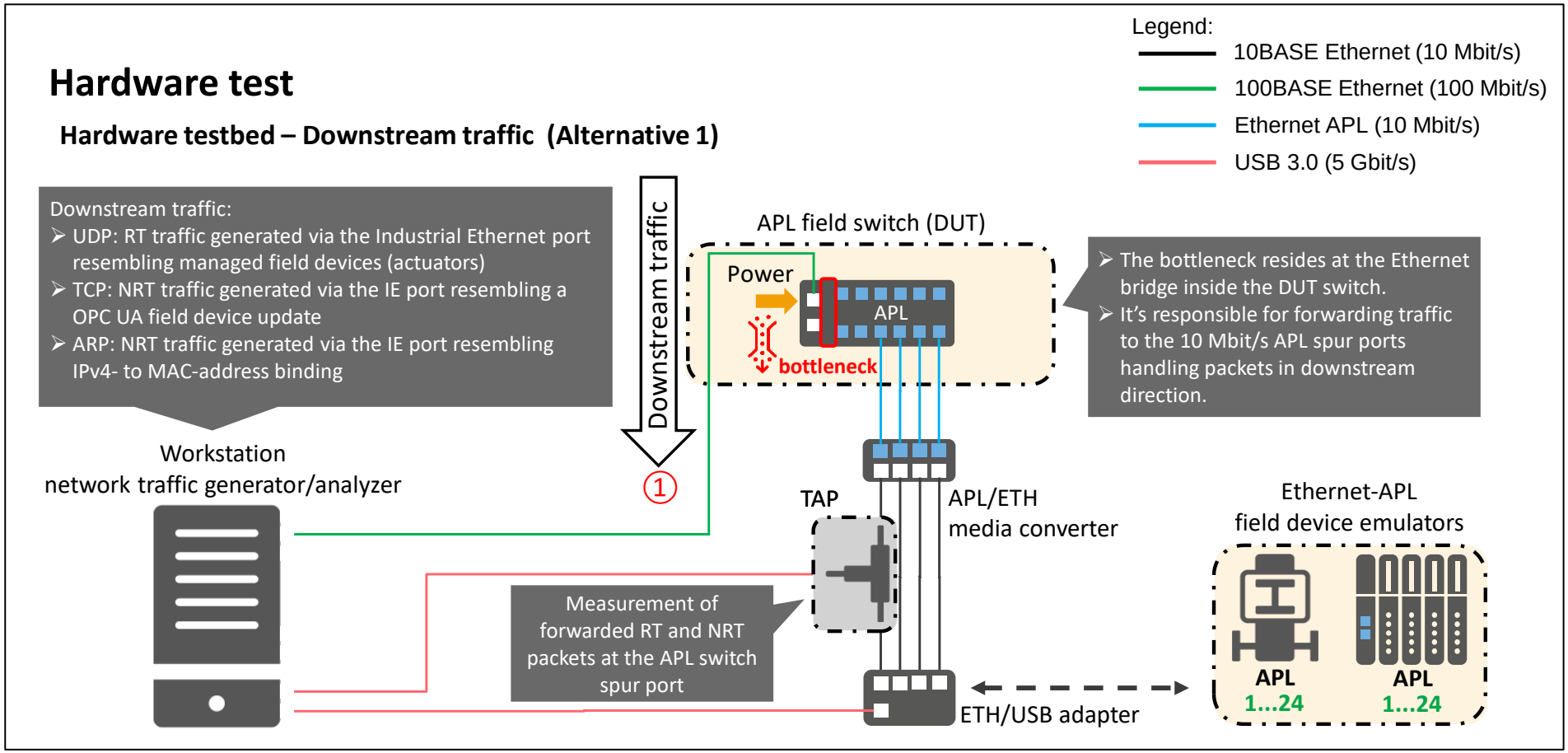

Figure 16: Hardware testbed for congestion loss-analysis of downstream traffic (Alternative 1)

The main purpose of downstream traffic analysis is to find out how the APL switch behaves in terms of packet handling for '*mixed traffic*'. Such traffic includes multiple different traffic types (i.e., ARP, UDP, TCP) that are received by the APL switch simultaneously. As the traffic is generated on the workstation using Ostinato, there is full control over packet load, packet count and packet priority.

The mixed traffic direction starts from the workstation sending packets down to the APL switch via one of its industrial Fast Ethernet egress ports. Then, traffic follows through one of the industrial Fast Ethernet ingress ports of the APL switch down to its multiple Ethernet-APL egress spur ports. Finally, the packets get forwarded from the Ethernet-APL egress spur ports of the APL switch to the Ethernet-APL ingress ports of the field device emulators.

In the test structure shown in Figure 16, the bottleneck resides at the Ethernet-APL field switch, which has to handle the change of transmission speed from the incoming 100 Mbit/s industrial Ethernet port to the multiple outgoing 10 Mbit/s Ethernet-APL spur ports.

In between the transmission-speed change from the incoming industrial Ethernet port to the outgoing Ethernet-APL spur ports, an additional internal Ethernet bridge handles packet forwarding at the physical layer. Considering traffic in downstream direction, this bridge has to handle the transition from one ingress queue to multiple egress queues, with a limited buffer size at its disposal.

Therefore, traffic will be analyzed in downstream direction between the workstation that is sending packets via the 100 Mbit/s Fast Ethernet port of the field switch, down to each respective field device emulators connected to the 10 Mbit/s Ethernet-APL spur ports. For traffic measurement, the TAP device is placed between the APL switch and one of the field device spur line connections to capture all traffic that is being forwarded.

6.2. Hardware testbed - Upstream traffic analysis (Alternative 1)

Figure 17 shows the setup for accurate upstream traffic analysis. It is based on the network structure shown in Figure 14 (chapter 6), which in turn is based on the more general hardware test setup shown in Figure *9* (chapter 4.1).

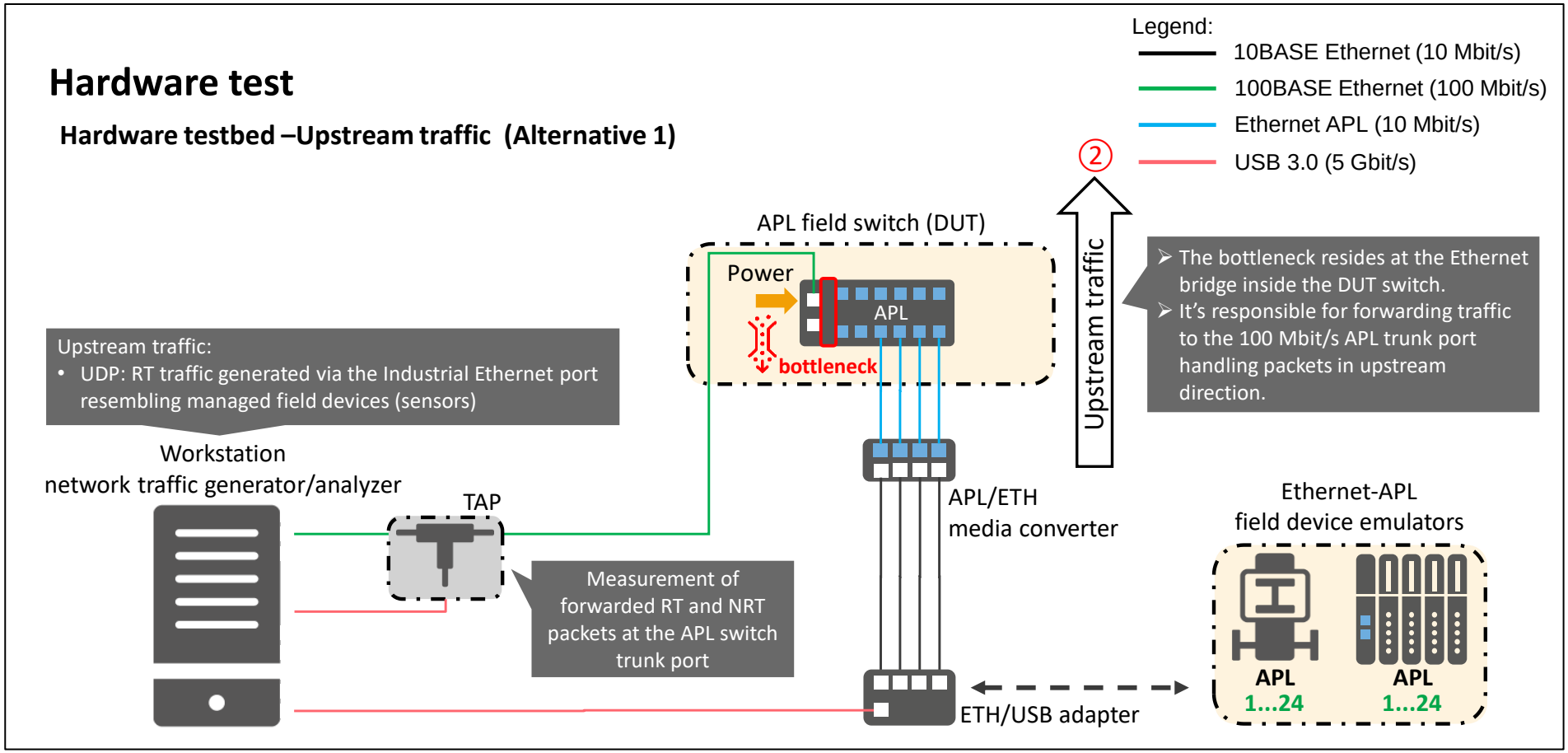

Figure 17: Hardware testbed for congestion loss-analysis of upstream traffic (Alternative 1)

Figure 17 shows the packet processing behavior of the APL switch while being stressed with bursty traffic in upstream direction.

The main goal of upstream traffic analysis is to find out, how the APL switch behaves when handling '*bursty traffic*'. Such traffic only includes one traffic type (i.e., UDP) that is sent to the APL switch by multiple senders at the same point in time. It is generated on the field device emulators using Ostinato, which allows full control over packet load, packet count and packet priority.

The bursty traffic direction starts from the field device emulators sending packets up to the APL switch via their Ethernet-APL egress spur ports. Traffic then follows through the Ethernet-APL ingress spur ports of the APL switch, up to one of the industrial Fast Ethernet egress ports. In the last step, the packets get forwarded from the industrial Fast Ethernet egress port of the APL switch to one industrial Fast Ethernet ingress port of the workstation.

In this particular test structure, the bottleneck resides at the Ethernet-APL field switches, which have to handle the change of transmission speed from multiple incoming 10 Mbit/s Ethernet-APL spur ports to one outgoing 100 Mbit/s Fast Ethernet port. In a bursty traffic scenario simultaneous incoming data streams at multiple spur ports may outweigh the higher egress port speed, which can cause packet queuing and overlapping of subsequent packet stream cycles.

In other words, the outgoing transmission speed may not be high enough to process all packets of a reoccurring traffic burst caused by multiple incoming packet stream, which can lead to potential overflow of the packet buffer inside the switch hardware.

The change of transmission speed between ingress and egress ports is handled by the internal Ethernet bridge of the switch. It's responsible for packet forwarding at the physical layer. Considering traffic in upstream direction, this bridge must handle the transition of multiple ingress queues to one egress queue with a limited buffer size.

Therefore, traffic will be analyzed in upstream direction between the field device emulators simultaneously sending packets via the 10 Mbit/s Ethernet-APL spur ports of the APL field switch up to the workstation connected via the 100 Mbit/s Fast Ethernet port.

For traffic measurement, the TAP device is placed between the APL switch and the workstation line connection in order to capture all traffic forwarded by the APL field switch originally generated by the field devices.

Note: With the help of the existing test setup, there is only a limited possibility to generate stable network loads distributed over multiple Ethernet interfaces.

This is due to hardware-related limitations, such as jitter during the transmission of the individual data packets along the individual network adapters (USB/ETH ↔ ETH/APL). There are also software-related limitations, such as missing synchronization of the individual interfaces due to the lack of real-time capability between Windows and Ostinato.

Thus, in the present tests with distributed network load over multiple network interfaces, packet-throughput instabilities occur with increasing traffic payload, which has to be considered during evaluation.

6.3. Hardware testbed – Downstream traffic analysis (Alternative 2)

Figure 18 shows the setup for accurate downstream traffic analysis. It is based on the network structure shown in Figure 15 (chapter 6), which in turn is based on the more general hardware test setup shown in Figure *9* (chapter 4.1).

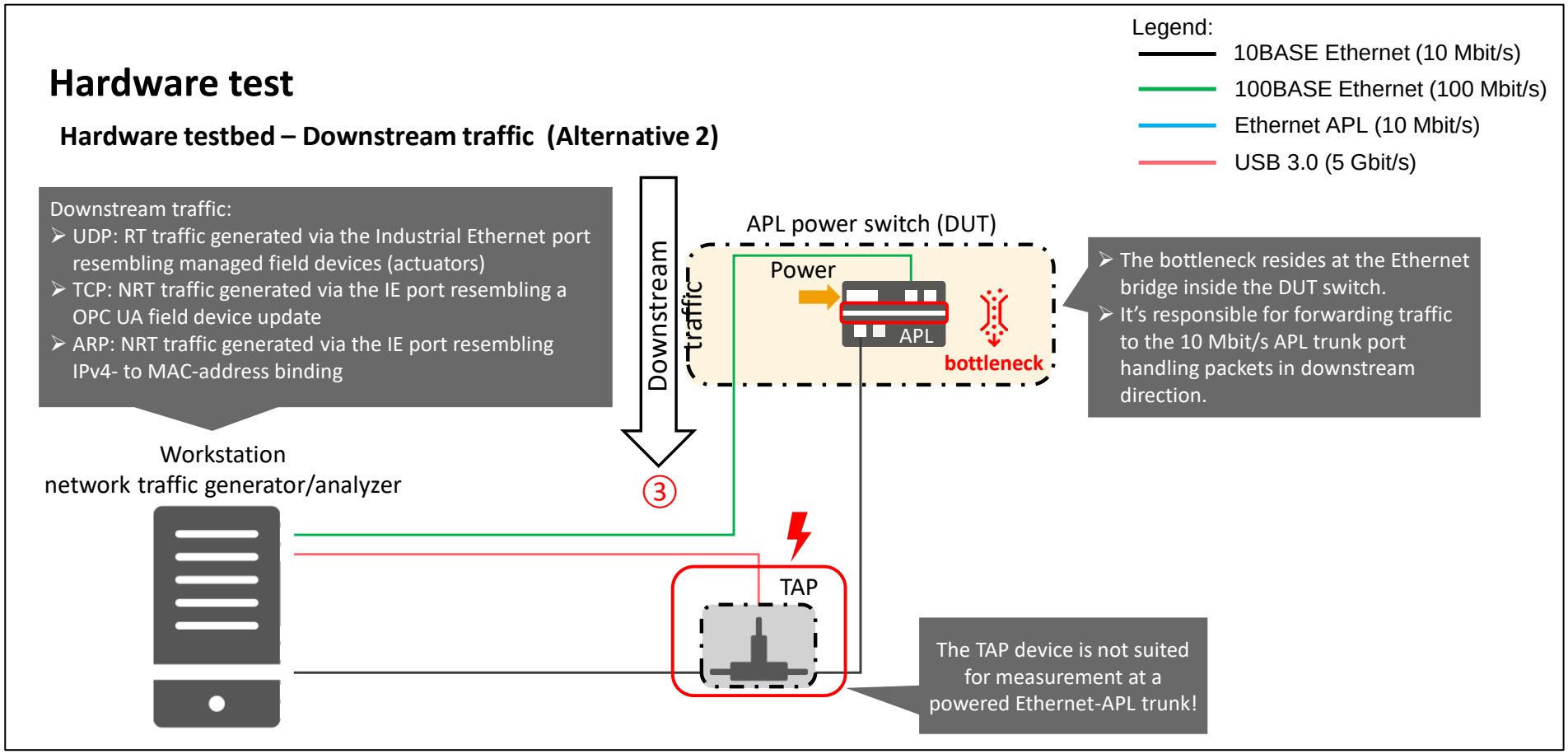

Figure 18: Hardware testbed for congestion loss-analysis of downstream traffic (Alternative 2)

Note: Due to the unavailability of a powered trunk APL/ETH media converter, downstream traffic analysis at the powered Ethernet-APL trunk port was not possible.

6.4. Hardware testbed – Upstream traffic analysis (Alternative 2)

Figure 19 depicts the setup for accurate upstream traffic analysis. It is based on the network structure shown in Figure 15 (chapter 6), which in turn is based on the more general hardware test setup shown in Figure *9* (chapter 4.1).

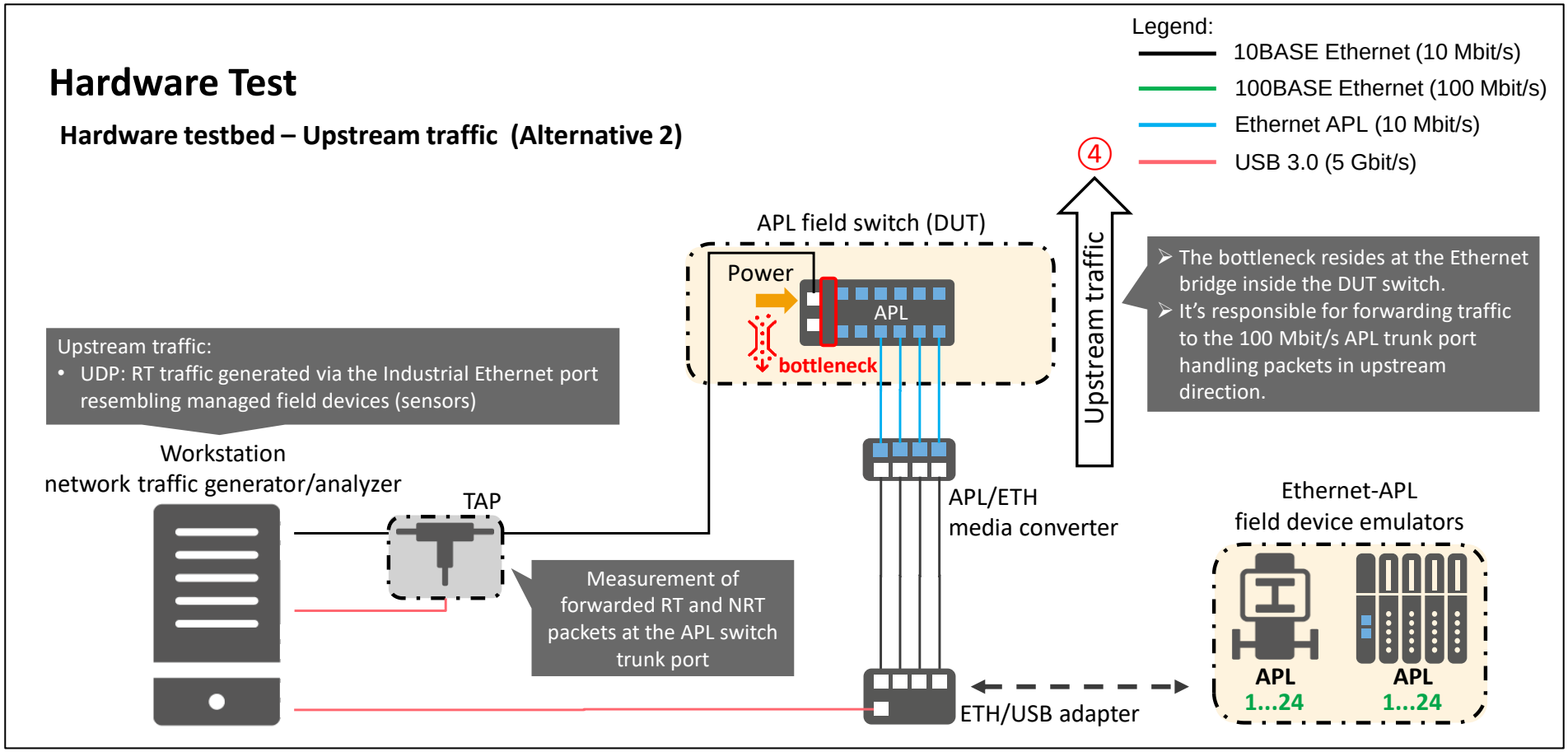

Figure 19: Hardware testbed for congestion loss-analysis of upstream traffic (Alternative 2)

Note: The upstream traffic analysis shown in Figure 19 follows the same measurement principles as applied in chapter 6.3. The only difference is the change in transmission speed at the Fast Ethernet egress port of the APL switch from 100 Mbit/s to 10 Mbit/s.

6.5. Hardware testbed – Interpretation of measurement results

The hardware measurements have been conducted by analyzing the packet-throughput behavior of the APL switches in terms of packet loss under various packet load conditions. As the data processing inside the switch cannot be measured, analysis is restricted to packet-throughput at fixed intervals (packets/second) rather than a '*packet-by-packet*' analysis. This means that all packets that enter and leave the APL switch at its ingress and egress ports are measured. The difference between incoming and outgoing packets at the switch then gives insight about potential packet loss.

To understand the analysis based on packet-throughput, a quick overview shall be given based on a simple example measurement result.

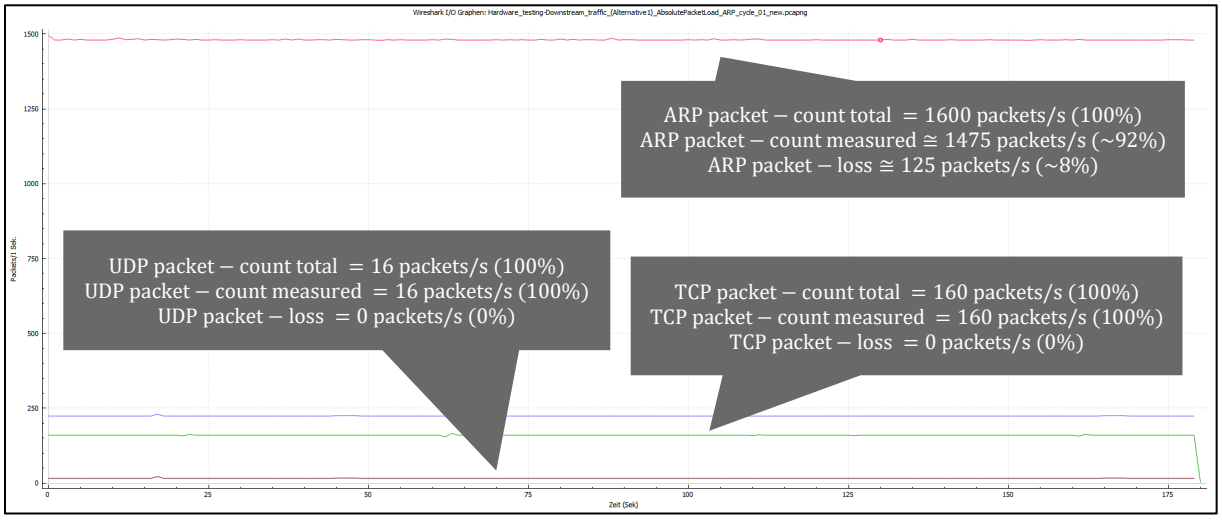

Figure 20: Example packet-throughput measurement based on packet count for each traffic-type

A measurement typically consists of multiple recorded traffic types showing their packet count over a fixed interval (i.e., packets/second). The various traffic types are color coded. ARP traffic is depicted in magenta. TCP traffic is shown in green. And UDP traffic is divided into two colored lines. Blue stands for the total amount of UDP traffic recorded across all field devices whereas brown only shows the UDP traffic from one field device.

The analysis of packet-throughput behavior of the APL switch is based on the number of incoming packets which must be processed inside the switch hardware and subsequently forwarded without packet loss and without significant delays. The labels in Figure 20 show this very information by stating the total amount of packets received at the ingress ports of the switch as well as the number of packets subsequently forwarded out of the switch at its egress ports. The difference between incoming and outgoing packet count is the number of packets dropped along the way indicating packet loss.

This information is necessary to understand the appropriate packet processing behavior of the switch in certain traffic scenarios and will be provided throughout all following measurements.

Disclaimer: Prior to looking at the measurement results, summarizing the previous chapters, a measurement is composed of:

- Generated packets from the workstation by means of "emulated field devices" using Ostinato (see chapter 4.1 and 4.2)

Said packets are structured by:

- Uni-; Multi-, Broadcast frames (see chapter 2.1.2)
- Packet types and packet priority (see chapter 2.1.2, 2.1.3 and 4.2)

7. Hardware test – Measurement results, summary

The following chapter summarizes all measurements conducted in chapter A.3 to A.5. To show briefly whether the packet processing behavior of the tested APL switches was within expectations, a color-coded system has been implemented. Each color refers to a specific packet loss threshold used to grade the success rate of the packet forwarding process.

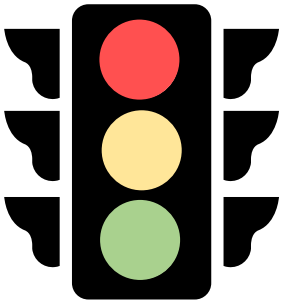

Table 3: packet loss severity

RED LIGHT: More than a quarter of all total packets get lost. $(\textbf{packet loss} > 25\%)$
YELLOW LIGHT: Less than a quarter of all packets get lost. $(\textbf{packet loss}: 1 \dots 25\%)$
GREEN LIGHT: Less than a hundredth of all total packets get lost. $(\textbf{packet loss}: < 1\%)$

Downstream and upstream traffic for Manufacturer A is summarized in Table *4* to Table *6*.
Downstream and upstream traffic for Manufacturer B is summarized in Table *7* to Table *8*.

Note: The packet loss stated in Table 3, which is highlighted in different colors to indicate severity, should only give the reader a basic guidance as to when the packet-throughput behavior of a specific traffic type begins to deteriorate. Depending on the traffic type, packet loss has a different reason and impact on network functionality.

Because connection-based traffic types such as TCP can keep track of lost information and resend it, packet loss is less impactful. However, this is quite different for non-connection-based traffic types such as RTC (here replaced by UDP traffic) for which no such mechanisms for retransmission exist. Therefore, packet loss not only depends on the percentage of packets lost in total but rather the acceptable threshold of discardable packets for each traffic type.

Note: Each column of Table 4 to Table 9 resembles a different traffic type with its own packet priority notified by the VLAN tag. Each table row resembles one measurement for each of the stated traffic types. For a deeper understanding of the stated measurement methods, each table row is referenced to a detailed measurement explanation found in the appendix (see chapter A.3…A.5).

Disclaimer: The packet-load parameters shown in Table 4 to Table 9 have been used in in consultation with the manufacturers as well as in accordance with the PROFINET specific robustness requirements. [PNO_7302_01]

7.1. Hardware test – Downstream traffic summary (Alternative 1), Manufacturer A

Table 4: Downstream traffic summary (Alternative 1), Manufacturer A

Packet processing	UDP real-time data single actuator	TCP IP field device update	ARP request
@ *'MinimumFrameMemory'* (see chapter A.3.1)	TPP[1]: 88 Byte PUP[2] (VLAN): 6 (RTC) PCT[3]: 62,5 ms PCC[4]: 24 · 1 packets / cycle FPC[5]: ~2 kByte/cycle PCS[6]: 384 packets/s FPS[7]: ~33 kByte/s **packet loss**: **0**%	TPP: 1542 Byte PUP (VLAN): 4 (OPC UA) PCT: 500 ms PCC: 80 packets / cycle FPC: ~120 kByte/cycle PCS: 160 packets/s FPS: ~241 kByte/s **packet loss**: **0**%	TPP: 88 Byte PUP (VLAN): 0 (DCP) PCT: 125 ms PCC: 100 packets / cycle FPC: ~9 kByte/cycle PCS: 800 packets/s FPS: ~69 kByte/s **packet loss**: **0**%
@ decreasing ARP cycle time (see chapter A.3.2.1)	see row 1 /column 1 **packet loss**: **~0** / **~0** / **~0**%	see row 1 /column 2 **packet loss**: **~0** / **~0** / **~0**%	PCT: 62,5/ 31,25/ 15,625 ms PCS: 1.600/ 3.200/ 6.400 packets/s FPS: ~138/ ~275/ ~550 kByte/cycle **packet loss**: **~8** / **~16** / **~18**%
@ increasing ARP packet count (see chapter A.3.2.2)	see row 1 /column 1 **packet loss**: **~0** / **~0** / **~0**%	see row 1 /column 2 **packet loss**: **~0** / **~0** / **~0**%	PCC: 200/ 400/ 800 packets / cycle FPC: ~17/ ~34/ ~69 kByte/cycle PCS: 1.600/ 3.200/ 6.400 packets/s FPS: ~138/ ~275/ ~550 kByte/cycle **packet loss**: **~25** / **~46** / **~82**%
@ decreasing TCP cycle time (see chapter A.3.2.3)	see row 1 /column 1 **packet loss**: **~0** / **~0** / **~0**%	PCT: 125 ms / 62,5 ms / 31,25 ms PCS: 640/ 1.280/ 2.560 packets/s FPS: ~964/ ~1.928/ ~3.855 kByte/cycle **packet loss**: **~0** / **~38** / **~69**%	see row 1 /column 2 **packet loss**: **0**/ **100**/ **100**%
@ increasing TCP packet count (see chapter A.3.2.4)	see row 1 /column 1 **packet loss**: **~0** / **~0** / **~6**%	PCC: 320/ 640/ 1.280 packets / cycle PCS: 640/ 1.280/ 2.560 packets/s FPS: ~964/ ~1.928/ ~3.855 kByte/cycle **packet loss**: **~50** / **~70** / **~81**%	see row 1 /column 3 **packet loss**: **15**/ **15**/ **31**%

[1] TPP: total packet payload
[2] PUP: packet user priority
[3] PCT: packet cycle time
[4] PCC: packet count per cycle
[5] FPC: total frame payload per cycle
[6] PCS: packet count per second
[7] FPS: total frame payload per second

Summary (Table 4): Table 4 shows the packet processing behavior of the APL switch from Manufacturer A, while being stressed with mixed traffic in downstream direction, according to chapter 6.1.

The expected behavior of the switch is that in congestion situations, the high priority packets will continue to be transported and when needed, the lower priority packets will be dropped.

While handling incoming packet streams consisting of multiple different traffic types, packet handling inside the internal packet buffer according to packet prioritization becomes the most significant factor for packet loss. In other words, packet loss is highly dependent on the priority of the arriving packets and should be handled accordingly by the packet processing of the APL switch.

The following traffic type user priorities (VLAN tags) have been set for testing the packet handling behavior of the APL switches: (see also chapter 4.2 and A.3.2)

- UDP (VLAN#: 6) highest priority
- TCP (VLAN#: 4) medium priority
- ARP(VLAN#: 0) no priority

Each incoming packet is validated and handled according to its designated user priority. If the packet count of a specific traffic type surpasses the internal hardware limitations of the switch, packet prioritization still upholds the manageable packet count of packets with higher priority before handling the ones with lower priority. This ensures upholding packet streams with high priority even in bursty traffic scenarios that would otherwise pose a risk of potential packet loss.

The packet loss information in column 1 shows, that the APL switch always tries to uphold system relevant UDP traffic, which is the desired packet prioritization behavior according to IEEE 802.1Q. Additionally, ARP and TCP traffic also show no packet loss at the '*MinimumFrameMemory*' condition. This condition defines the minimal buffer memory size needed to prevent packet loss in non-real-time traffic due to interference with real-time traffic. The '*MinimumFrameMemory*' buffer size is calculated by multiplying the number of Ethernet ports, that are working at a fixed datarate, with the '*MaxPortBlockingTime*' of the egress ports, which are delaying queued non-real-time traffic due to packet prioritization of higher prioritized real-time traffic. [PNO_2722_02, p. 400 et seq.]

While increasing ARP and TCP traffic above the '*MinimumFrameMemory*' condition, packet loss for these traffic types may occur. However, this is the desired behavior and happens due to the internal hardware limitations of the APL switch, as discussed earlier in chapter 5.1. While processing packet loads that exceed the internal hardware limitations of the switch, packets are still processed according to packet prioritization, which leads to packet discards in lower priority traffic types, as is highlighted in rows 2 to 5.Only by continuous increase of TCP traffic with high packet loads per packet, packet prioritization begins to struggle and shows signs of UDP packet loss in row 5. However, considering a real industrial network environment, this test case is not very likely. Whether or not TCP packet loss becomes possible depends on the number and interval of TCP packet bursts, managed by flow control mechanisms. If these aspects are properly considered, packet loss may be avoided.

7.2. Hardware test – Upstream traffic summary (Alternative 1), Manufacturer A

Table 5: Upstream traffic summary (Alternative 1), Manufacturer A

Packet processing	UDP real-time data single sensor
@ *'SimultaneousTrafficBurst'* (see chapter A.4.1)	TPP8: 88 Byte PUP9 (VLAN): 6 (RTC) PCT10: 62,5 ms PCC11: 4 · 1 / 8 · 1 / 16 · 1 / 24 · 1 packets / cycle FPC[12]: 352 / 704 / 1.408 / 2.112 Byte/cycle PCS[13]: 64 / 128 / 256 / 384 packets/s FPS[14]: ~6 /~11 /~22 /~33 kByte/s ***packet loss***: **0**% / **0**% / **0**% / ~**3**%
@ decreasing UDP cycle time (see chapter A.4.2.1)	PCT: 31,25 ms / 15,625 ms / 7,8125 ms PCS: 768/ 1.536/ 3.072 packets/s FPS: ~66/ ~132/ ~264 kByte/cycle **packet loss**: ~**1**% / ~**3**% / ~**6**%
@ increasing UDP packet count (see chapter A.4.2.2)	PCC: 24 · 32/ 24 · 64/ 24 · 128 packets / cycle PCS: 12.288/ 24.576/ 49.152 packets/s FPS: ~1.056/ ~2.112/ ~4.224 kByte/cycle **packet loss**: ~**4**% / ~**3**% / ~**4**%

[8] TPP: total packet payload
[9] PUP: packet user priority
[10] PCT: packet cycle time
[11] PCC: packet count per cycle
[12] FPC: total frame payload per cycle
[13] PCS: packet count per second
[14] FPS: total frame payload per second

Summary (Table 5): Table 5 shows the packet processing behavior of the APL switch from Manufacturer A, while being stressed with bursty traffic in upstream direction using the setup described in chapter 6.2.

The expected behavior of the switch is that in congestion situations, the maximum amount of high priority packets will continue to be transported, while further increase of packets in a shorter timeframe will increase the amount of dropped packets.

While forwarding packet streams coming in via multiple ingress ports through only one egress port, there are two key factors that determine packet loss. One is the size of the internal packet buffer storing the incoming packets. The other is the PPT ('*Packet Processing Time*') required for forwarding the packets one-by-one (FIFO). In other words, packet loss is highly dependent on incoming and outgoing transmission speed as well as the internal storage capacity of the APL switch.

The packet loss information in column 1 shows that system relevant UDP traffic is not always transferred without packet loss. However, the losses are not related to a faulty behavior of the switch, but rather to the missing synchronization of the emulated field devices in Ostinato. While testing with more than 16 to 24 field devices, all generating packet load via the workstation at the same time, an increasing packet-throughput instability for UDP traffic is becoming noticeable. This unintended behavior originates from missing synchronization between the emulated field devices in Ostinato. Unfortunately, Ostinato showed significant performance problems when managing more than 8 Ethernet interfaces. When sending packets through each interface at the same time, strong fluctuations in packet-throughput could be observed (see chapter 6.2).

Therefore, the packet loss information shown in row 1 is not caused by the switch itself and thus indicates that the APL-Switch is able to fulfil the *'SimultaneousTrafficBurst*' condition. Said condition refers to the unlikely but possible event of multiple field devices sending out their system relevant RTC packets (represented here by UDP traffic) simultaneously to the ingress ports of the connected APL switch, at a cycle time of 64 ms. Simultaneously occurring packet bursts can typically be avoided by a correct network setup using accurate packet control by the controller, i. e. regulating its field device responses in a homogenous distribution across one entire cycle. However, when considering realistic packet load scenarios, simultaneous packet bursts may occur and thus must be taken into consideration.

Further increasing the UDP traffic beyond the *'SimultaneousTrafficBurst*' condition results in increased packet loss, as is shown by the results in row 2 to 3. Regardless of said increase in packet loss due to the field device emulation, the APL switch manages to uphold all incoming UDP packets without problems. Additionally, it should be noted that this test case might not be particularly likely in a real industrial network environment. Whether or not it becomes an issue depends on the number and interval of packet bursts which in turn depend on the cycle time and number of field devices used in the setup. If these aspects are properly considered, packet loss could be avoided.

7.3. Hardware test – Upstream traffic summary (Alternative 2), Manufacturer A

Table 6: Upstream traffic summary, Manufacturer A

Packet processing	UDP real-time data single sensor
@ *'SimultaneousTrafficBurst'* (see chapter A.5.1)	TPP15: 88 Byte PUP16 (VLAN): 6 (RTC) PCT17: 62,5 ms PCC18: 4 · 1 / 8 · 1 / 16 · 1 / 24 · 1 packets / cycle FPC[19]: 352 / 704 / 1.408 / 2.112 Byte/cycle PCS[20]: 64 / 128 / 256 / 384 packets/s FPS[21]: ~6 /~11 /~22 /~33 kByte/s ***packet loss***: **0**% / **0**% / **0**% / ~**2**%
@ decreasing UDP cycle time (see chapter A.5.2.1)	PCT: 31,25 ms / 15,625 ms / 7,8125 ms PCS: 768/ 1.536/ 3.072 packets/s FPS: ~66/ ~132/ ~264 kByte/cycle **packet loss**: ~**5**% / ~**5**% / ~**6**%
@ increasing UDP packet count (see chapter A.5.2.2)	PCC: 24 · 32/ 24 · 64/ 24 · 128 packets / cycle PCS: 12.288/ 24.576/ 49.152 packets/s FPS: ~1.056/ ~2.112/ ~4.224 kByte/cycle **packet loss**: ~**2**% / ~**40**% / ~**70**%

[15] TPP: total packet payload
[16] PUP: packet user priority
[17] PCT: packet cycle time
[18] PCC: packet count per cycle
[19] FPC: total frame payload per cycle
[20] PCS: packet count per second
[21] FPS: total frame payload per second

Summary (Table *6*): Table *6* shows the packet processing behavior of the APL switch from Manufacturer A, while being stressed with bursty traffic in upstream direction, according to chapter 6.4.

The expected behavior of the switch is that in congestion situations, the maximum amount of high priority packets will continue to be transported, while further increase of packets in a shorter timeframe will increase the amount of dropped packets. When comparing the results of the two upstream measurements shown in Table *5* and Table *6*, one may observe that decreasing the transmission speed from $100\ \mathrm{Mbit/s}$ to $10\ \mathrm{Mbit/s}$ on the outbound egress port results in a sharp rise in packet losses. The same issues with synchronization between field device emulators described in chapter 6.2 are also occurring here. Therefore, the packet loss information in row 1 as well as row 2 is not related to a faulty behavior of the switch.

However, aside from this explainable portion of packet loss, further increase of UDP traffic as listed in row 3, shows a significant increase in packet loss. This packet loss happens due to the combined packet load of UDP traffic exceeding the '*dataRate*' limit of the $10\ \mathrm{Mbit/s}$ APL trunk line *(*$\sim 1.19\ \mathrm{MByte/s}$).

The likelihood of UDP traffic reaching packet loads as described in the above test scenarios might not be very high when looking at a real industrial network. However, it still needs to be considered for prevention of possible system-relevant packet loss. Therefore, the number and interval of packet bursts, which in turn depend on the cycle time and number of field devices used in the setup, need to be taken into account while operating the network, similar to the previous measurement summary of Table *5* (chapter 7.2).

7.4. Hardware test – Downstream traffic summary (Alternative 1), Manufacturer B

Table 7: Downstream traffic summary, Manufacturer B

Packet processing	UDP real-time data single actuator	TCP IP field device update	ARP request
@ *'MinimumFrameMemory'* (see chapter A.3.1)	TPP[22]: 88 Byte PUP[23] (VLAN): 6 (RTC) PCT[24]: 62,5 ms PCC[25]: 24 · 1 packets / cycle FPC[26]: ~2 kByte/cycle PCS[27]: 384 packets/s FPS[28]: ~33 kByte/s **packet loss**: **0**%	TPP: 1542 Byte PUP (VLAN): 4 (OPC UA) PCT: 500 ms PCC: 80 packets / cycle FPC: ~120 kByte/cycle PCS: 384 packets/s FPS: ~241 kByte/s **packet loss**: **0**%	TPP: 88 Byte PUP (VLAN): 0 (DCP) PCT: 125 ms PCC: 100 packets / cycle FPC: ~9 kByte/cycle PCS: 800 packets/s FPS: ~69 kByte/s **packet loss**: **0**%
@ decreasing ARP cycle time (see chapter A.3.2.1)	see row 1 /column 1 **packet loss**: **~0 / ~0 / ~0**%	see row 1 /column 2 **packet loss**: **~0 / ~0 / ~0**%	PCT: 62,5/ 31,25/ 15,625 ms PCS: 1.600/ 3.200/ 6.400 packets/s FPS: ~138/ ~275/ ~550 kByte/cycle **packet loss**: **~0 / ~0 / ~0**%
@ increasing ARP packet count (see chapter A.3.2.2)	see row 1 /column 1 **packet loss**: **~0 / ~0 / ~0**%	see row 1 /column 2 **packet loss**: **~0 / ~0 / ~46%**	PCC: 200/ 400/ 800 packets / cycle FPC: ~18/ ~35/ ~70 kByte/cycle PCS: 1.600/ 3.200/ 6.400 packets/s FPS: ~138/ ~275/ ~550 kByte/cycle **packet loss**: **~0 / ~0 / ~0**%
@ decreasing TCP cycle time (see chapter A.3.2.3)	see row 1 /column 1 **packet loss**: **0/ 0/ ~10**%	PCT: 125 ms / 62,5 ms / 31,25 ms PCS: 640/ 1.280/ 2.560 packets/s FPS: ~964/ ~1.928/ ~3.855 kByte/cycle **packet loss**: **~0 / ~39 / ~70**%	see row 1 /column 2 **packet loss**: **0/ 3/ 6**%
@ increasing TCP packet count (see chapter A.3.2.4)	see row 1 /column 1 **packet loss**: **~0 / ~10 / ~23**%	PCC: 320/ 640/ 1.280 packets / cycle PCS: 640/ 1.280/ 2.560 packets/s FPS: ~964/ ~1.928/ ~3.855 kByte/cycle **packet loss**: **~49 / ~70 / ~81**%	see row 1 /column 3 **packet loss**: **~0 / ~0 / ~16**%

[22] TPP: total packet payload
[23] PUP: packet user priority
[24] PCT: packet cycle time
[25] PCC: packet count per cycle
[26] FPC: total frame payload per cycle
[27] PCS: packet count per second
[28] FPS: total frame payload per second

Summary (Table 7): Table 7 shows the packet processing behavior of the APL switch from Manufacturer B while being stressed with mixed traffic in downstream direction, according to chapter 6.1.

The expected behavior of the switch is that in congestion situations, the maximum amount of high priority packets will continue to be transported, while further increase of packets in a shorter timeframe will increase the amount of dropped packets. A direct comparison of Table 4 and Table 7 shows that the APL switch from Manufacturer B handles downstream traffic quite similarly to that from Manufacturer A. The stable packet-throughput of system relevant UDP traffic in column 1 shows that the APL switch from Manufacturer B also tries to maintain high-priority traffic by preventing packet loss.

By gradual increase of ARP and TCP traffic as shown in rows 2 to 5, packet loss in these traffic types occurs in a similar pattern as was observed in chapter 6.1. Additionally, the packet prioritization between TCP and ARP traffic sometimes fails. More specifically, while working at the hardware packet processing limit of the APL switch, TCP traffic shows packet loss prior to ARP traffic, even though TCP has the higher priority.

When increasing TCP traffic even further and with high packet loads per packet, packet prioritization starts to fail completely and shows signs of UDP packet loss in row 5. However, when reviewing these results, one must consider that such packet loads might not be very likely to occur in a real industrial network environment.

7.5. Hardware test – Upstream traffic summary (Alternative 1), Manufacturer B

Table 8: Upstream traffic summary, Manufacturer B

Packet processing	UDP real-time data single sensor
@ *'SimultaneousTrafficBurst'* (see chapter A.4.1)	TPP[29]: 88 Byte PUP[30] (VLAN): 6 (RTC) PCT[31]: 62,5 ms PCC[32]: 1 · 4 / 2 · 4 / 4 · 4 / 6 · 4 packets/cycle FPC[33]: 352 / 704 / 1.408 / 2.112 Byte/cycle PCS[34]: 64 / 128 / 256 / 384 packets/s FPS[35]: ~6 /~11 /~22 /~33 kByte/s ***packet loss***: **0**% / **0**% / **0**% / **0**%
@ndecreasing UDP cycle time (see chapter A.4.2.1)	PCT: 31,25 ms / 15,625 ms / 7,8125 ms PCS: 768/ 1.536/ 3.072 packets/s FPS: ~66/ ~132/ ~264 kByte/cycle **packet loss**: **0** / **0** / **0**%
@ increasing UDP packet count (see chapter A.4.2.2)	PCC: 24 · 128/ 24x256 / 512 packets / cycle PCS: 8.192 / 16.384 / 32.768 packets/s FPS: ~704 / ~1.408 / ~2.816 kByte/cycle **packet loss**: **0** / **0** / **0**%

[29] TPP: total packet payload
[30] PUP: packet user priority
[31] PCT: packet cycle time
[32] PCC: packet count per cycle
[33] FPC: total frame payload per cycle
[34] PCS: packet count per second
[35] FPS: total frame payload per second

Summary (Table 8): Table 8 shows the packet processing behavior of the APL switch from Manufacturer B, while being stressed with bursty traffic in upstream direction, according to chapter 6.1.

The expected behavior of the switch is that in congestion situations, the maximum amount of high priority packets will continue to be transported, while further increase of packets in a shorter timeframe will increase the amount of dropped packets.

The packet loss information in column 1 to 5 shows that the APL switch of Manufacturer B can maintain stable-packet-throughputs for high-priority UDP traffic in all measurements. This is regardless of gradual traffic increases and while working at a datarate of $100\ \mathrm{Mbit/s}$ at the egress port.

The direct comparison of Table *5* and Table 8 shows that the APL switch of Manufacturer B did not experience any stability issues regarding UDP packet throughout, as it was the case in the test with Manufacturer A. This might be because testing was only possible with a limited amount of field devices (1...4 instead of 1…24) in comparison to Manufacturer A, thus not provoking synchronization issues between the field device emulators handled via Ostinato.

7.6. Hardware test – Upstream traffic summary (Alternative 2), Manufacturer B

Table 9: Upstream traffic summary, Manufacturer B

Packet processing	UDP real-time data single sensor
@ *'SimultaneousTrafficBurst'* (see chapter A.5.1)	TPP[36]: 88 Byte PUP[37] (VLAN): 6 (RTC) PCT[38]: 62,5 ms PCC[39]: 1 · 4 / 2 · 4 / 4 · 4 / 6 · 4 packets/cycle FPC[40]: 352 / 704 / 1.408 / 2.112 Byte/cycle PCS[41]: 64 / 128 / 256 / 384 packets/s FPS[42]: ~6 /~11 /~22/~33 kByte/s ***packet loss***: **0**% / **0**% / **0**% / **0**%
@ decreasing UDP cycle time (see chapter A.5.2.1)	PCT: 31,25 ms / 15,625 ms / 7,8125 ms PCS: 768/ 1.536/ 3.072 packets/s FPS: ~66 / ~132 / ~264 kByte/cycle **packet loss**: ~**0**% / ~**4**% / **63**%
@ increasing UDP packet count (see chapter A.5.2.2)	PCC: 4 · 32 / 4 · 64 / 4 · 128 packets/cycle PCS: 2.048 / 4.096 / 8.192 packets/s FPS: ~176 / ~352 / ~704 kByte/s **packet loss**: ~**0**% / ~**0**% / ~**43**%

[36] TPP: total packet payload
[37] PUP: packet user priority
[38] PCT: packet cycle time
[39] PCC: packet count per cycle
[40] FPC: total frame payload per cycle
[41] PCS: packet count per second
[42] FPS: total frame payload per second

Summary (Table 9): Table 9 shows the packet processing behavior of the APL switch from Manufacturer B, while being stressed with bursty traffic in upstream direction, according to chapter 6.4.

The expected behavior of the switch is that in congestion situations, the maximum amount of high priority packets will continue to be transported, while further increase of packets in a shorter timeframe will increase the amount of dropped packets.

When comparing the results of the two upstream measurements shown in Table 8 and Table 9, one may observe that decreasing the transmission speed from $100\ \mathrm{Mbit/s}$ to $10\ \mathrm{Mbit/s}$ on the egress port results in a sharp rise in packet losses.

By flooding the APL switch hardware with packets sent via multiple ingress ports, severe packet loss as well as shutdown of ingress ports of the APL switch are possible side effects. This behavior was observed with the ingress ports all working at the same link speed as the single egress port responsible for forwarding packets. Said issues become visible when increasing the UDP traffic beyond the *'SimultaneousTrafficBurst*' condition, resulting in increased packet loss, as is shown by the results in row 2…3. This leads to the conclusion, that '*10 Mbit/s –to-10 Mbit/s*' upstream connections should be avoided to prevent packet loss.

However, generating packet load up to or even beyond the *'SimultaneousTrafficBurst*' condition might again not be very likely in a real industrial network environment. The probability of packet-loss according to the above measurements needs to be questioned in a similar matter as described in the measurement summary of Table *5* (chapter 7.2).

8. Conclusion and Outlook

The purpose of this thesis was to investigate the potential of packet loss in '*mixed link speed*' networks that are formed by the combined use of industrial Ethernet and Ethernet-APL working at different datarates. Running parts of the network at different datarates leads to a '*congestion loss*' problem, which can lead to limitations in packet-throughput and consequently to potential packet loss of system-relevant data. Such packet loss can jeopardize the behavior of the network which controls the automation processes of the plant and therefore must be avoided at all costs.

Various tests have been performed to determine whether there are packet losses when operating the available APL switches in a '*mixed link speed*' network or not.

The measurements showed that the APL switches were capable of performing successful packet processing without packet losses, based on testing the *'MinimumFrameMemory*' and the '*SimultaneousTrafficBurst*' conditions, which reflect the requirements set for down- and upstreaming packet data according to the PROFINET robustness requirements (refer to chapter 7 or A.3.1, A.4.1 and A.5.1).

Even by further increase of packet load beyond the hardware limitations, both APL switches managed to maintain correct packet processing most of the time, which surpassed the expectations (refer to chapter 7 or A.3.2, A.4.2 and A.5.2).

Only a few specific packet overload scenarios lead to packet processing problems in terms of high-priority packet loss due to missing packet prioritization or exceeding packet processing limitations. However, said overload scenarios are deemed very unlikely in a real automation network.
(refer to chapter 7.1, 7.4 and 7.6 or A.3.2.4, A.5.2.1 and A.5.2.2)

To conclude the analysis of the conducted measurements, it can be said that the potential of packet loss in the '*mixed link speed*' networks tested is relatively low, posing no threat to correct network function.

This indicates that the future implementation of '*mixed link speed*' networks is possible in terms of packet utilization if the equipment is designed correctly.

Due to issues with the test setup regarding the synchronization of packet generation, some measurements showed packet loss, without faulty packet processing behavior of the tested APL switches (refer to chapter 7.2 and 7.3 or A.4 and A.5). These issues only allowed a limited analysis of packet processing behavior in certain test cases and thus leaves room for improvement.

A possible improvement of the test setup could for example include the use of '*Raspberry PIs*' (RasPI), replacing the field device emulators controlled via the frame generator software Ostinato. The RasPIs can generate packets via a packet driver software utilizing the API of the microcomputer. The exact synchronization of packets can be ensured by signal-triggering at the binary inputs of the RasPI, which can be conditioned to wait for a signal change (rising/falling signal edge) generated by a signal generator.

V. Bibliography

chronological order

[EAPL_PPS_01]
[EAPL__PPS_02, p.05]
[EAPL_PPS_03, p.14]

PROFIBUS Nutzerorganisation e.V., ODVA, FieldComm Group, and OPC Foundation, '*Port Profile Specification - Ethernet-APL*', Version 1.0, June 2021,
Hyperlink: https://www.profibus.com/download/port-profile-specification-ethernet-apl [accessed 09.10.2022]

[IEEE_8023CG_01]
[IEEE_8023CG_02, p.86 et seq.]

IEEE Computer Society, IEEE 802.3CG, '*IEEE Standard for Ethernet Amendment 5: Physical Layers Specifications and Management Parameters for 10 Mb/s Operation and Associated Power Delivery over a Single Balanced Pair of Conductors*', November 2019,
Hyperlink: https://standards.ieee.org/ieee/802.3cg/7308/
[accessed 09.10.2022]

[PNO_EAPL_01]]

PROFIBUS Nutzerorganisation e.V., '*Whitepaper - Ethernet-APL*', June 2021, Hyperlink: https://www.ethernet-apl.org/document/ethernet-to-the-field-whitepaper/ [accessed 23.08.2022]

[NieK_EAPL_01, p.57 et seq.]

Karl-Heinz Niemann, '*Engineering Guideline - Ethernet-APL*', Version 1.11, December 2021,
Hyperlink: https://www.ethernet-apl.org/document/ethernet-apl-engineering-guidelines/ [accessed 31.08.2022]

[PNO_8061_01, p. 164 et seq.]
[PNO_8061_02, p. 61 et seq.]
[PNO_8061_03, p. 124 et seq.]

PROFIBUS Nutzerorganisation e.V., '*Guideline or PROFINET*', Version 147, April 2022, File name: PROFINET_Guideline_8061_V147 _Apr22, Hyperlink:
https://www.profibus.com/download/profinet-installation-guidelines [accessed 09.10.2022]

[SpuCha_01, p. 17]
[SpuCha_02, p.45 et seq.]
[SpuCha_03, p.31/302]
[SpuCha_04, p.30 f. et seq.]
[SpuCha_05, p.300]
[SpuCha_06, p.109 et seq.]
[SpuCha_07, p.308 et seq.]
[SpuCha_08, p.303 et seq.]

Charles E. Spurgeon' '*Ethernet the definitive guide*', 2. Edition, 2014, Reilly Media, Inc. (publisher), ISBN: 978-1-449-36184-6

[Plixer_01]

Plixer, '*Network layers explained*':
https://www.plixer.com/blog/network-layers-explained/
[accessed 29.08.2022]

[KocR_01, p. 41 et seq.]

Ricarda Koch, '*Kommunikationsnetze in der Automatisierungstechnik*', 2019, Siemens AG,
Publicis Pixelpark (publisher), ISBN: 978-3-89578-441-5

[IEEE_8023_01, p. 22 et seq.] [IEEE_8023_02, p. 56 et seq.] [IEEE_8023_03, p. 118 et seq.] [IEEE_8023_04, p. 98/118 et. seq.] [IEEE_8023_05, p. 105] [IEEE_8023_06, p. 139 et seq.]	IEEE Computer Society, IEEE 802.3, 'IEEE Standard for Ethernet', June 2018, Hyperlink: https://standards.ieee.org/ieee/802.3/7071/ [accessed 09.10.2022]
[IEEE_8021Q_01, p. 128 et seq.] [IEEE_8021Q_02, p. 1907 et seq.] [IEEE_8021Q_03, p. 178 et seq.] [IEEE_8021Q_04, p. 179 et seq.] [IEEE_8021Q_05, p. 180 et seq.] [IEEE_8021Q_06, p. 1918 et seq.]	IEEE Computer Society, IEEE 802.1Q, '*IEEE Standard for Local and Metropolitan Area Networks - Bridges and Bridged Networks*', May 2018, Hyperlink: https://standards.ieee.org/ieee/802.1Q/6844/ [accessed 09.10.2022]
[PNO_2712_01]	PROFIBUS Nutzerorganisation e.V., '*Application Layer services for decentralized periphery – Technical Specification for PROFINET*', Version 24, 2021, File name: PN-AL-services_2712_V24MU2_Apr21, Hyperlink: https://www.profibus.com/download/profinet-specification [accessed 09.10.2022]
[PNO_2722_01, p. 163 et seq.] [PNO_2722_02, p. 400 et seq.]	PROFIBUS Nutzerorganisation e.V., '*Application Layer protocol for decentralized periphery – Technical Specification for PROFINET*', Version 24, 2021' File name: PN-AL-protocol_2722_V24MU2_Apr21, Hyperlink: https://www.profibus.com/download/profinet-specification [accessed 09.10.2022]
[IEC_61158-5-10]	IEC- International Electrotechnical Commission, IEC_61158-5-10:2019: '*Industrial communication networks - Fieldbus specifications - Part 5-10: Application layer service definition - Type 10 elements*', 2019 Hyperlink: https://webstore.iec.ch/publication/64836. [accessed 20.01.2023]
[IEC_61158-6-10]	IEC- International Electrotechnical Commission, IEC 61158-6-10:2019: '*Industrial communication networks - Fieldbus specifications - Part 6-10: Application layer protocol specification - Type 10 elements*', 2019 Hyperlink: https://webstore.iec.ch/publication/59893. [accessed 20.01.2023]
[PigRa_01, p. 72] [PigRa_02, p. 64 et seq.]	Raimond Pigan, '*Automatisieren mit PROFINET*', 2. Edition, 2008, Siemens AG, Publicis Corporate Publishing (publisher), ISBN: 978-3-89578-293-0
[Omnet_01]	OMNeT, '*OMNeT ++ simulation environment*', Hyperlink: https://omnetpp.org/ [accessed 27.08.2022]

[Omnet_02] OMNeT, '*INET Framework*', Hyperlink: https://inet.omnetpp.org/ [accessed 27.08.2022]

[Omnet_03] OMNeT, '*INET User's Guide – Queueing Model*', Hyperlink: https://inet.omnetpp.org/docs/users-guide/ch-queueing.html [accessed 27.08.2022]

[Omnet_04] OMNeT, '*INET Tutorials – Queueing Tutorial*', Hyperlink: https://inet.omnetpp.org/docs/tutorials/queueing/doc/index.html [accessed 23.08.2022]

[Osti_01] Ostinato, '*Ostinato – User Guide*', Hyperlink: https://userguide.ostinato.org/ [accessed 27.08.2022]

[ProTAP_01] ProfiTAP, '*ProfiShark – Capture Tool*', Hyperlink: https://www.profitap.com/profishark-1g/ [accessed 27.08.2022]

[WireSh_01] WireShark, '*WireShark – User Guide*': https://www.wireshark.org/docs/wsug_html_chunked/ [accessed 27.08.2022]

[PNO_7302_01] PROFIBUS Nutzerorganisation e.V., '*Netload Robustness - Guideline for PROFINET*', Draft, March 2022, File name: PN-Netload-Robustness_7302_d20_Mar22, Hyperlink: https://www.profibus.com/download/profinet-netload-robustness-for-security-guideline-former-security-level-1-netload [accessed 09.10.2022]

VI. List of figures

VII. List of tables

VIII. Appendix A

All listed files are accessible via the '*Academic Cloud*' WebServer:
Hyperlink: https://sync.academiccloud.de/index.php/s/eOVhIseCA4IMi2x
Password: IpPDbEAPL_010922

A.1 Simulation testing – file overview

folder directory	data	description
Simulation test	Simulation_Test_Model_Structure&Results.pptx	Presentation about Ethernet switch packet processing behavior at the Physical Layer based on a simulation model built I OMNet++
*\Omnet++ \projects \APL_Downstreaming-basic_model-mixed-cuttrough.7z	*.project	OMNet++ project file, Ethernet Switch handling mixed downstream flow of packets
*\simulations	downstream_mixed.ini	INI-file for parametrization of the simulation model submodule functions
	downstream_mixed.ned	NED-file representing the simulation model build with the OMNet++/INET framework
	mixed-burst.anf	ANF-file holding the captured packet data flow after simulation run-through
*\Omnet++ \projects	*.project	OMNet++ project file, Ethernet Switch handling

\APL_Upstreaming-basic_model-burst-shaped-cuttrough.7z		shaped upstream flow of packets
*\simulations	upstream_burst_shaped.ini	INI-file for parametrization of the simulation model submodule functions
	upstream_burst_shaped.ned	NED-file representing the simulation model build with the OMNet++/INET framework
	burst-shaped.anf	ANF-file holding the captured packet data flow after simulation run-through
*\Omnet++ \projects \APL_Upstreaming-basic_model-burst-unshaped-cuttrough.7z	*.project	OMNet++ project file, Ethernet Switch handling unshaped upstream flow of packets
*\simulations	upstream_burst_unshaped.ini	INI-file for parametrization of the simulation model submodule functions
	upstream_burst_unshaped.ned	NED-file representing the simulation model build with the OMNet++/INET framework
	burst-unshaped.anf	ANF-file holding the captured packet data flow after simulation run-through
*\Omnet++ \captures	*.PNG	Screenshots

A.2 Hardware test – file overview

folder directory	data	description
Hardware test \Testaufbau \Messungen \Hardware limitations \Manufacturer A	Hardware_testing-Upstream_traffic_(ManufacturerA)_BufferLengthLimit_01-04.ossn	Ostinato measurement data of Manufacturer A conducted in 0
*\Messdaten	Hardware_testing-Upstream_traffic_(ManufacturerA)_BufferLengthLimit_01_01.pcapng	WireShark measurement data of Manufacturer A conducted in 0
	Hardware_testing-Upstream_traffic_(ManufacturerA)_BufferLengthLimit_01_02.pcapng	
	Hardware_testing-Upstream_traffic_(ManufacturerA)_BufferLengthLimit_01_03.pcapng	
	Hardware_testing-Upstream_traffic_(ManufacturerA)_BufferLengthLimit_01_04.pcapng	
*\org	*.PCAPNG	raw WireShark measurement data repository
* \Bilder	*.JPG	Screenshots
Hardware test \Testaufbau \Messungen \Hardware limitations \Manufacturer B	Hardware_testing-Upstream_traffic_(ManufacturerB)_BufferCountLimit_01.ossn	Ostinato measurement data of Manufacturer B conducted in 5.2.1
	Hardware_testing-Upstream_traffic_(ManufacturerB)_BufferCountLimit_02.ossn	
	Hardware_testing-Upstream_traffic_(ManufacturerB)_BufferCountLimit_03.ossn	
*\Messdaten	Hardware_testing-Upstream_traffic_(ManufacturerB)_BufferCountLimit_new_01.pcapng	WireShark measurement data of Manufacturer B conducted in 5.2.1
	Hardware_testing-Upstream_traffic_(ManufacturerB)_BufferCountLimit_new_02.pcapng	
	Hardware_testing-Upstream_traffic_(ManufacturerB)_BufferCountLimit_new_03.pcapng	
*\org	*.PCAPNG	raw WireShark measurement data repository
* \Bilder	*.JPG	Screenshots
Hardware test \Testaufbau \Messungen \Downstream	Switch_Port_Summary.JPG	Hardware setting of APL switch ports of Manufacturer A for measurements conducted in chapter A.3

\ManufacturerA \Hardware settings	TAP_Port_Summary.JPG	Hardware setting of TAP device ports for measurements conducted in chapter A.3
	workstation_Port_Summary.JPG	Hardware setting of workstation ports for measurements conducted in chapter A.3
Hardware test \Testaufbau \Messungen \Downstream \ManufacturerA	Hardware_testing-Downstream_traffic_(ManufacturerA)_MinimumFrameMemory_01.ossn	Ostinato measurement data of Manufacturer A conducted in chapter A.3.1
	Hardware_testing-Downstream_traffic_(ManufacturerA)_MinimumFrameMemory_02.ossn	
	Hardware_testing-Downstream_traffic_(ManufacturerA)_AbsolutePacketLoad_ARP_cycle_01.ossn	Ostinato measurement data of Manufacturer A conducted in chapter A.3.2.1
	Hardware_testing-Downstream_traffic_(ManufacturerA)_AbsolutePacketLoad_ARP_cycle_02.ossn	
	Hardware_testing-Downstream_traffic_(ManufacturerA)_AbsolutePacketLoad_ARP_cycle_03.ossn	
	Hardware_testing-Downstream_traffic_(ManufacturerA)_AbsolutePacketLoad_ARP_burst_01.ossn	Ostinato measurement data of Manufacturer A conducted in chapter A.3.2.2
	Hardware_testing-Downstream_traffic_(ManufacturerA)_AbsolutePacketLoad_ARP_burst_02.ossn	
	Hardware_testing-Downstream_traffic_(ManufacturerA)_AbsolutePacketLoad_ARP_burst_03.ossn	
	Hardware_testing-Downstream_traffic_(ManufacturerA)_AbsolutePacketLoad_TCP_cycle_01.ossn	Ostinato measurement data of Manufacturer A conducted in chapter A.3.2.3
	Hardware_testing-Downstream_traffic_(ManufacturerA)_AbsolutePacketLoad_TCP_cycle_02.ossn	
	Hardware_testing-Downstream_traffic_(ManufacturerA)_AbsolutePacketLoad_TCP_cycle_03.ossn	
	Hardware_testing-Downstream_traffic_(ManufacturerA)_AbsolutePacketLoad_TCP_burst_01.ossn	Ostinato measurement data of Manufacturer A conducted in chapter A.3.2.4
	Hardware_testing-Downstream_traffic_(ManufacturerA)_AbsolutePacketLoad_TCP_burst_02.ossn	
	Hardware_testing-Downstream_traffic_(ManufacturerA)_AbsolutePacketLoad_TCP_burst_03.ossn	
*\Messdaten	Hardware_testing-Downstream_traffic_(ManufacturerA)_MinimumFrameMemory_01_new.pcapng	WireShark measurement data of Manufacturer A conducted in chapter A.3.1
	Hardware_testing-Downstream_traffic_(ManufacturerA)_MinimumFrameMemory_02_new.pcapng	
	Hardware_testing-Downstream_traffic_(ManufacturerA)_AbsolutePacketLoad_ARP_cycle_01_new.pcapng	WireShark measurement data of Manufacturer A conducted in chapter A.3.2.1
	Hardware_testing-Downstream_traffic_(ManufacturerA)_AbsolutePacketLoad_ARP_cycle_02_new.pcapng	
	Hardware_testing-Downstream_traffic_(ManufacturerA)_AbsolutePacketLoad_ARP_cycle_03_new.pcapng	

	Hardware_testing-Downstream_traffic_(ManufacturerA)_AbsolutePacketLoad_ARP_burst_01_new.pcapng	WireShark measurement data of Manufacturer A conducted in chapter A.3.2.2
	Hardware_testing-Downstream_traffic_(ManufacturerA)_AbsolutePacketLoad_ARP_burst_02_new.pcapng	
	Hardware_testing-Downstream_traffic_(ManufacturerA)_AbsolutePacketLoad_ARP_burst_03_new.pcapng	
	Hardware_testing-Downstream_traffic_(ManufacturerA)_AbsolutePacketLoad_TCP_cycle_01_new.pcapng	WireShark measurement data of Manufacturer A conducted in chapter A.3.2.3
	Hardware_testing-Downstream_traffic_(ManufacturerA)_AbsolutePacketLoad_TCP_cycle_02_new.pcapng	
	Hardware_testing-Downstream_traffic_(ManufacturerA)_AbsolutePacketLoad_TCP_cycle_03_new.pcapng	
	Hardware_testing-Downstream_traffic_(ManufacturerA)_AbsolutePacketLoad_TCP_burst_01_new.pcapng	WireShark measurement data of Manufacturer A conducted in chapter A.3.2.4
	Hardware_testing-Downstream_traffic_(ManufacturerA)_AbsolutePacketLoad_TCP_burst_02_new.pcapng	
	Hardware_testing-Downstream_traffic_(ManufacturerA)_AbsolutePacketLoad_TCP_burst_03_new.pcapng	
*\org	*.PCAPNG	raw WireShark measurement data repository
*\Bilder	*.JPG	Screenshots
Hardware test \Testaufbau \Messungen \Downstream \ManufacturerB \Hardware settings	Switch_Port_Summary.JPGa	Hardware setting of APL switch ports of Manufacturer B for measurements conducted in chapter A.3
	TAP_Port_Summary.JPG	Hardware setting of TAP device ports for measurements conducted in chapter A.3
	workstation_Port_Summary.JPG	Hardware setting of APL switch ports of Manufacturer A for

		measurements conducted in chapter A.3
Hardware test \Testaufbau \Messungen \Downstream \ManufacturerB	Hardware_testing-Downstream_traffic_(ManufacturerB)_MinimumFrameMemory_01.ossn	Ostinato measurement data of Manufacturer B conducted in chapter A.3.1
	Hardware_testing-Downstream_traffic_(ManufacturerB)_MinimumFrameMemory_02.ossn	
	Hardware_testing-Downstream_traffic_(ManufacturerB)_AbsolutePacketLoad_ARP_cycle_01.ossn	Ostinato measurement data of Manufacturer B conducted in chapter A.3.2.1
	Hardware_testing-Downstream_traffic_(ManufacturerB)_AbsolutePacketLoad_ARP_cycle_02.ossn	
	Hardware_testing-Downstream_traffic_(ManufacturerB)_AbsolutePacketLoad_ARP_cycle_03.ossn	
	Hardware_testing-Downstream_traffic_(ManufacturerB)_AbsolutePacketLoad_ARP_burst_01.ossn	Ostinato measurement data of Manufacturer B conducted in chapter A.3.2.2
	Hardware_testing-Downstream_traffic_(ManufacturerB)_AbsolutePacketLoad_ARP_burst_02.ossn	
	Hardware_testing-Downstream_traffic_(ManufacturerB)_AbsolutePacketLoad_ARP_burst_03.ossn	
	Hardware_testing-Downstream_traffic_(ManufacturerB)_AbsolutePacketLoad_TCP_cycle_01.ossn	Ostinato measurement data of Manufacturer B conducted in chapter A.3.2.3
	Hardware_testing-Downstream_traffic_(ManufacturerB)_AbsolutePacketLoad_TCP_cycle_02.ossn	
	Hardware_testing-Downstream_traffic_(ManufacturerB)_AbsolutePacketLoad_TCP_cycle_03.ossn	
	Hardware_testing-Downstream_traffic_(ManufacturerB)_AbsolutePacketLoad_TCP_burst_01.ossn	Ostinato measurement data of Manufacturer B conducted in chapter A.3.2.4
	Hardware_testing-Downstream_traffic_(ManufacturerB)_AbsolutePacketLoad_TCP_burst_02.ossn	
	Hardware_testing-Downstream_traffic_(ManufacturerB)_AbsolutePacketLoad_TCP_burst_03.ossn	
*\Messdaten	Hardware_testing-Downstream_traffic_(ManufacturerB)_MinimumFrameMemory_01_new.pcapng	WireShark measurement data of Manufacturer B conducted in chapter A.3.1
	Hardware_testing-Downstream_traffic_(ManufacturerB)_MinimumFrameMemory_02_new.pcapng	
	Hardware_testing-Downstream_traffic_(ManufacturerB)_AbsolutePacketLoad_ARP_cycle_01_new.pcapng	WireShark measurement data of Manufacturer B conducted in chapter A.3.2.1
	Hardware_testing-Downstream_traffic_(ManufacturerB)_AbsolutePacketLoad_ARP_cycle_02_new.pcapng	
	Hardware_testing-Downstream_traffic_(ManufacturerB)_AbsolutePacketLoad_ARP_cycle_03_new.pcapng	
	Hardware_testing-Downstream_traffic_(ManufacturerB)_AbsolutePacketLoad_ARP_burst_01_new.pcapng	WireShark measurement data of Manufacturer B conducted in chapter A.3.2.2
	Hardware_testing-Downstream_traffic_(ManufacturerB)_AbsolutePacketLoad_ARP_burst_02_new.pcapng	

	Hardware_testing-Downstream_traffic_(ManufacturerB)_AbsolutePacketLoad_ARP_burst_03_new.pcapng	
	Hardware_testing-Downstream_traffic_(ManufacturerB)_AbsolutePacketLoad_TCP_cycle_01_new.pcapng	WireShark measurement data of Manufacturer B conducted in chapter A.3.2.3
	Hardware_testing-Downstream_traffic_(ManufacturerB)_AbsolutePacketLoad_TCP_cycle_02_new.pcapng	
	Hardware_testing-Downstream_traffic_(ManufacturerB)_AbsolutePacketLoad_TCP_cycle_03_new.pcapng	
	Hardware_testing-Downstream_traffic_(ManufacturerB)_AbsolutePacketLoad_TCP_burst_01_new.pcapng	WireShark measurement data of Manufacturer B conducted in chapter A.3.2.4
	Hardware_testing-Downstream_traffic_(ManufacturerB)_AbsolutePacketLoad_TCP_burst_02_new.pcapng	
	Hardware_testing-Downstream_traffic_(ManufacturerB)_AbsolutePacketLoad_TCP_burst_03_new.pcapng	
*\org	*.PCAPNG	raw WireShark measurement data repository
* \Bilder	*.JPG	Screenshots
Hardware test \Testaufbau \Messungen \Upstream \Fast Ethernet trunk (100 Mbit) \ManufacturerA \Hardware settings	Switch_Port_Summary.JPG	Hardware setting of APL switch ports of Manufacturer A for measurements conducted in chapter A.4
	TAP_Port_Summary.JPG	Hardware setting of TAP device ports for measurements conducted in chapter A.4
	workstation_Port_Summary.JPG	Hardware setting of workstation ports for measurements conducted in chapter A.4
Hardware test \Testaufbau \Messungen	Hardware_testing-Upstream_traffic_(ManufacturerA)_SimultanousTrafficBurst_01-04.ossn	Ostinato measurement data of Manufacturer A conducted in chapter A.4.1

\Upstream \Fast Ethernet trunk (100 Mbit) \ManufacturerA	Hardware_testing-Upstream_traffic_(ManufacturerA)_AbsolutePacketLoad_UDP_cycle_01.ossn	Ostinato measurement data of Manufacturer A conducted in chapter A.4.2.1
	Hardware_testing-Upstream_traffic_(ManufacturerA)_AbsolutePacketLoad_UDP_cycle_02.ossn	
	Hardware_testing-Upstream_traffic_(ManufacturerA)_AbsolutePacketLoad_UDP_cycle_03.ossn	
	Hardware_testing-Upstream_traffic_(ManufacturerA)_AbsolutePacketLoad_UDP_burst_01.ossn	Ostinato measurement data of Manufacturer A conducted in chapter A.4.2.2
	Hardware_testing-Upstream_traffic_(ManufacturerA)_AbsolutePacketLoad_UDP_burst_02.ossn	
	Hardware_testing-Upstream_traffic_(ManufacturerA)_AbsolutePacketLoad_UDP_burst_03.ossn	
*\Messdaten	Hardware_testing-Upstream_traffic_(ManufacturerA)_SimultaneousTrafficBurst_01.pcapng	WireShark measurement data of Manufacturer A conducted in chapter A.4.1
	Hardware_testing-Upstream_traffic_(ManufacturerA)_SimultaneousTrafficBurst_02.pcapng	
	Hardware_testing-Upstream_traffic_(ManufacturerA)_SimultaneousTrafficBurst_03.pcapng	
	Hardware_testing-Upstream_traffic_(ManufacturerA)_SimultaneousTrafficBurst_04.pcapng	
	Hardware_testing-Upstream_traffic_(ManufacturerA)_AbsolutePacketLoad_UDP_cycle_01.pcapng	WireShark measurement data of Manufacturer A conducted in chapter A.4.2.1
	Hardware_testing-Upstream_traffic_(ManufacturerA)_AbsolutePacketLoad_UDP_cycle_02.pcapng	
	Hardware_testing-Upstream_traffic_(ManufacturerA)_AbsolutePacketLoad_UDP_cycle_03.pcapng	
	Hardware_testing-Upstream_traffic_(ManufacturerA)_AbsolutePacketLoad_UDP_burst_01.pcapng	WireShark measurement data of Manufacturer A conducted in chapter A.4.2.2
	Hardware_testing-Upstream_traffic_(ManufacturerA)_AbsolutePacketLoad_UDP_burst_02.pcapng	
	Hardware_testing-Upstream_traffic_(ManufacturerA)_AbsolutePacketLoad_UDP_burst_03.pcapng	
*\org	*.PCAPNG	raw WireShark measurement data repository
* \Bilder	*.JPG	Screenshots
Hardware test \Testaufbau \Messungen \Upstream \Fast Ethernet trunk (100 Mbit) \ManufacturerB \Hardware settings	Switch_Port_Summary.JPG	Hardware setting of APL switch ports of Manufacturer B for measurements conducted in chapter A.4
	TAP_Port_Summary.JPG	Hardware setting of TAP device ports for measurements conducted in chapter A.4
	workstation_Port_Summary.JPG	Hardware setting of workstation ports for measurements conducted in chapter A.4
Testaufbau	Hardware_testing-Upstream_traffic_(ManufacturerB)_SimultanousTrafficBurst_01.ossn	

\Messungen \Upstream \Fast Ethernet trunk (100 Mbit) \ManufacturerB	Hardware_testing-Upstream_traffic_(ManufacturerB)_SimultanousTrafficBurst_02.ossn	Ostinato measurement data of Manufacturer B conducted in chapter A.4.1
	Hardware_testing-Upstream_traffic_(ManufacturerB)_SimultanousTrafficBurst_03.ossn	
	Hardware_testing-Upstream_traffic_(ManufacturerB)_SimultanousTrafficBurst_04.ossn	
	Hardware_testing-Upstream_traffic_(ManufacturerB)_AbsolutePacketLoad_UDP_cycle_01.ossn	Ostinato measurement data of Manufacturer B conducted in chapter A.4.2.1
	Hardware_testing-Upstream_traffic_(ManufacturerB)_AbsolutePacketLoad_UDP_cycle_02.ossn	
	Hardware_testing-Upstream_traffic_(ManufacturerB)_AbsolutePacketLoad_UDP_cycle_03.ossn	
	Hardware_testing-Upstream_traffic_(ManufacturerB)_AbsolutePacketLoad_UDP_burst_01.ossn	Ostinato measurement data of Manufacturer B conducted in chapter A.4.2.2
	Hardware_testing-Upstream_traffic_(ManufacturerB)_AbsolutePacketLoad_UDP_burst_02.ossn	
	Hardware_testing-Upstream_traffic_(ManufacturerB)_AbsolutePacketLoad_UDP_burst_03.ossn	
*\Messdaten	Hardware_testing-Upstream_traffic_(ManufacturerB)_SimultaneousTrafficBurst_01.pcapng	WireShark measurement data of Manufacturer B conducted in chapter A.4.1
	Hardware_testing-Upstream_traffic_(ManufacturerB)_SimultaneousTrafficBurst_02.pcapng	
	Hardware_testing-Upstream_traffic_(ManufacturerB)_SimultaneousTrafficBurst_03.pcapng	
	Hardware_testing-Upstream_traffic_(ManufacturerB)_SimultaneousTrafficBurst_04.pcapng	
	Hardware_testing-Upstream_traffic_(ManufacturerB)_AbsolutePacketLoad_UDP_cycle_01.pcapng	WireShark measurement data of Manufacturer B conducted in chapter A.4.2.1
	Hardware_testing-Upstream_traffic_(ManufacturerB)_AbsolutePacketLoad_UDP_cycle_02.pcapng	
	Hardware_testing-Upstream_traffic_(ManufacturerB)_AbsolutePacketLoad_UDP_cycle_03.pcapng	
	Hardware_testing-Upstream_traffic_(ManufacturerB)_AbsolutePacketLoad_UDP_burst_01.pcapng	Hardware setting of APL switch ports of Manufacturer B for measurements conducted in chapter A.4
	Hardware_testing-Upstream_traffic_(ManufacturerB)_AbsolutePacketLoad_UDP_burst_02.pcapng	
	Hardware_testing-Upstream_traffic_(ManufacturerB)_AbsolutePacketLoad_UDP_burst_03.pcapng	
*\org	*.PCAPNG	raw WireShark measurement data repository
* \Bilder	*.JPG	Screenshots
Hardware test \Testaufbau \Messungen \Upstream \Fast Ethernet trunk (10 Mbit) \ManufacturerA \Hardware settings	Switch_Port_Summary.JPG	Hardware setting of APL switch ports of Manufacturer A for measurements conducted in chapter A.5
	TAP_Port_Summary.JPG	Hardware setting of TAP device ports for measurements conducted in chapter A.5

	workstation_Port_Summary.JPG	Hardware setting of workstation ports for measurements conducted in chapter A.5
Hardware test \Testaufbau \Messungen \Upstream \unpowered APL trunk (10Mbit) \ManufacturerA	Hardware_testing-Upstream_traffic_(ManufacturerA)_SimultanousTrafficBurst_01-04.ossn	Ostinato measurement data of Manufacturer A conducted in chapter A.5.1
	Hardware_testing-Upstream_traffic_(ManufacturerA)_AbsolutePacketLoad_UDP_cycle_01.ossn	Ostinato measurement data of Manufacturer A conducted in chapter A.5.2.1
	Hardware_testing-Upstream_traffic_(ManufacturerA)_AbsolutePacketLoad_UDP_cycle_02.ossn	
	Hardware_testing-Upstream_traffic_(ManufacturerA)_AbsolutePacketLoad_UDP_cycle_03.ossn	
	Hardware_testing-Upstream_traffic_(ManufacturerA)_AbsolutePacketLoad_UDP_burst_01.ossn	Ostinato measurement data of Manufacturer A conducted in chapter A.5.2.2
	Hardware_testing-Upstream_traffic_(ManufacturerA)_AbsolutePacketLoad_UDP_burst_02.ossn	
	Hardware_testing-Upstream_traffic_(ManufacturerA)_AbsolutePacketLoad_UDP_burst_03.ossn	
*\Messdaten	Hardware_testing-Upstream_traffic_(ManufacturerA)_SimultaneousTrafficBurst_01.pcapng	WireShark measurement data of Manufacturer A conducted in chapter A.5.1
	Hardware_testing-Upstream_traffic_(ManufacturerA)_SimultaneousTrafficBurst_02.pcapng	
	Hardware_testing-Upstream_traffic_(ManufacturerA)_SimultaneousTrafficBurst_03.pcapng	
	Hardware_testing-Upstream_traffic_(ManufacturerA)_SimultaneousTrafficBurst_04.pcapng	
	Hardware_testing-Upstream_traffic_(ManufacturerA)_AbsolutePacketLoad_UDP_cycle_01.pcapng	WireShark measurement data of Manufacturer A conducted in chapter A.5.2.1
	Hardware_testing-Upstream_traffic_(ManufacturerA)_AbsolutePacketLoad_UDP_cycle_02.pcapng	
	Hardware_testing-Upstream_traffic_(ManufacturerA)_AbsolutePacketLoad_UDP_cycle_03.pcapng	
	Hardware_testing-Upstream_traffic_(ManufacturerA)_AbsolutePacketLoad_UDP_burst_01.pcapng	WireShark measurement data of Manufacturer A conducted in chapter A.5.2.2
	Hardware_testing-Upstream_traffic_(ManufacturerA)_AbsolutePacketLoad_UDP_burst_02.pcapng	
	Hardware_testing-Upstream_traffic_(ManufacturerA)_AbsolutePacketLoad_UDP_burst_03.pcapng	
*\org	*.PCAPNG	raw WireShark measurement data repository
* \Bilder	*.JPG	Screenshots
Hardware test \Testaufbau \Messungen \Upstream	Switch_Port_Summary.JPG	Hardware setting of APL switch ports of Manufacturer B for measurements conducted in chapter A.5

\Fast Ethernet trunk (10 Mbit) \ManufacturerB \Hardware settings	TAP_Port_Summary.JPG	Hardware setting of TAP device ports for measurements conducted in chapter A.5
	workstation_Port_Summary.JPG	Hardware setting of workstation ports for measurements conducted in chapter A.5
Hardware test \Testaufbau \Messungen \Upstream \unpowered APL trunk (10Mbit) \ManufacturerB	Hardware_testing-Upstream_traffic_(ManufacturerB)_SimultanousTrafficBurst_01.ossn	Ostinato measurement data of Manufacturer B conducted in chapter A.5.1
	Hardware_testing-Upstream_traffic_(ManufacturerB)_SimultanousTrafficBurst_02.ossn	
	Hardware_testing-Upstream_traffic_(ManufacturerB)_SimultanousTrafficBurst_03.ossn	
	Hardware_testing-Upstream_traffic_(ManufacturerB)_SimultanousTrafficBurst_04.ossn	
	Hardware_testing-Upstream_traffic_(ManufacturerB)_AbsolutePacketLoad_UDP_cycle_01.ossn	Ostinato measurement data of Manufacturer B conducted in chapter A.5.2.1
	Hardware_testing-Upstream_traffic_(ManufacturerB)_AbsolutePacketLoad_UDP_cycle_02.ossn	
	Hardware_testing-Upstream_traffic_(ManufacturerB)_AbsolutePacketLoad_UDP_cycle_03.ossn	
	Hardware_testing-Upstream_traffic_(ManufacturerB)_AbsolutePacketLoad_UDP_burst_01.ossn	Ostinato measurement data of Manufacturer B conducted in chapter A.5.2.2
	Hardware_testing-Upstream_traffic_(ManufacturerB)_AbsolutePacketLoad_UDP_burst_02.ossn	
	Hardware_testing-Upstream_traffic_(ManufacturerB)_AbsolutePacketLoad_UDP_burst_03.ossn	
*\Messdaten	Hardware_testing-Upstream_traffic_(ManufacturerB)_SimultaneousTrafficBurst_01.pcapng	WireShark measurement data of Manufacturer B conducted in chapter A.5.1
	Hardware_testing-Upstream_traffic_(ManufacturerB)_SimultaneousTrafficBurst_02.pcapng	
	Hardware_testing-Upstream_traffic_(ManufacturerB)_SimultaneousTrafficBurst_03.pcapng	
	Hardware_testing-Upstream_traffic_(ManufacturerB)_SimultaneousTrafficBurst_04.pcapng	
	Hardware_testing-Upstream_traffic_(ManufacturerB)_AbsolutePacketLoad_UDP_cycle_01.pcapng	WireShark measurement data of Manufacturer B conducted in chapter A.5.2.1
	Hardware_testing-Upstream_traffic_(ManufacturerB)_AbsolutePacketLoad_UDP_cycle_02.pcapng	
	Hardware_testing-Upstream_traffic_(ManufacturerB)_AbsolutePacketLoad_UDP_cycle_03.pcapng	
	Hardware_testing-Upstream_traffic_(ManufacturerB)_AbsolutePacketLoad_UDP_burst_01.pcapng	WireShark measurement data of Manufacturer B conducted in chapter A.5.2.2
	Hardware_testing-Upstream_traffic_(ManufacturerB)_AbsolutePacketLoad_UDP_burst_02.pcapng	
	Hardware_testing-Upstream_traffic_(ManufacturerB)_AbsolutePacketLoad_UDP_burst_03.pcapng	
*\org	*.PCAPNG	raw WireShark measurement data repository
* \Bilder	*.JPG	Screenshots

A.3 Hardware test – Downstream traffic (Alternative 1), Measurement results, Manufacturers A&B

Disclaimer: This chapter documents traffic measurement results in full detail and is meant to provide extensive background information on the results summarized in chapter 7.1 and 7.4. However, this level of detail is not necessary for understanding the essence of said results and serves only for the profound understanding of the measurement evaluation.

The following hardware measurement analyze the downstream traffic packet-throughput behavior of the tested APL switches according to Figure *16* (chapter 6.1).

Note: A detailed description for better understanding and interpreting the measurement result figures is presented in chapter 6.5

A.3.1 Packet-throughput analysis - Packet processing @ '*MinimumFrameMemory*' condition

The given Ethernet-APL switches handle packet data for up to 24 field devices sent via the Ethernet-APL spur ports in both packet stream directions. In addition to field device data, many other traffic types are also managed via the switches, which can result in large amounts of data. For handling such large amounts of incoming data of different traffic types, the packet buffer inside the switch must be able to process a minimum number of packets without discarding anything, to avoid packet loss.

The minimum buffer size needed for avoiding packet loss is determined by the '*MinimumFrameMemory*' condition that defines a value $128\,\mathrm{kByte}$, calculated for 24 ports at $10\,\mathrm{Mbit/s}$ transmission speed per port. [PNO_2722_02, p. 400 et seq.]

A hardware test shall validate whether the switches manage to fulfil the *'MinimumFrameMemory*' condition at their given hardware limitations (see chapter 5).

The packet processing behavior of the switches is deemed successful if stable packet-throughput without packet losses, is ensured for all traffic types.

Note: While analyzing the packet processing of a switch, packet load as well as packet count must be considered for accurate validation of potential packet loss.

For example, packet loss can already occur at packet loads smaller than $128\,\mathrm{kByte}$ by sending more than 128 packets to one packet queue of the switch from Manufacturer A, thereby exceeding its '*queueLength*' limit and provoking packet loss. Said packets might only have a packet load of $64\,\mathrm{Byte}$ per packet, resulting in a total load of $8\,\mathrm{kByte}$, which is much smaller than the packet load according to the '*MinimumFrameMemory*' condition.

Thus, the measurement results of the conducted test scenarios will be analyzed based on packet load as well as packet count.

The following packet load parameters stated in Table 10 have been used for testing:

Table 10: Downstream traffic analysis (Alternative 1), Manufacturers A&B – Traffic parameters

	UDP real-time data single actuator		TCP IP field device update	ARP request	
number of field devices	24 (Manufacturer A) 4 (Manufacturer B)		–	–	
user priority	**6 (RTC)**		**4**	**0**	
total packet payload **(TPP)**	46 Byte (data) + 42 Byte (framing/transmission) = 88 Byte		1.500 Byte (data) + 42 Byte (framing/transmission) = 1.542 Byte	46 Byte (data) + 42 Byte (framing/transmission) = 88 Byte	
packet cycle time **(PCT)**	62,5 ms		500 ms	250 ms / 125 ms	
Packet Count per Cycle **(PCC)**	1 · 24 packets / cycle (Manufacturer A)	6 · 4 packets / cycle (Manufacturer B)	100 packets / cycle	80 packets / cycle	
Packet data Payload per Cycle **(PPC)**	24 packets / cycle · 46 Byte = 1.104 Byte/cycle **(~1, 08 kByte/cycle)**		80 packets / cycle · 1.500 Byte = 120.000 Byte/cycle **(~117, 2 kByte/cycle)**	100 packets / cycle · 46 Byte = 4.600 Byte/cycle **(~4, 5 kByte/cycle)**	
total frame payload per cycle **(FPC)**	24 packets/cycle · 88 Byte = 2.112 Byte/cycle **(~2, 1 kByte/cycle)**		80 p/c · 1.542 Byte = 123.360 Byte/c **(~120, 5 kByte/c)**	100 p/c · 88 Byte = 8.800 Byte/c **(~8, 6 kByte/cycle)**	
Packet Count per Second **(PCS)**	$\frac{24 \cdot 1 \frac{\text{packets}}{\text{cycle}}}{62{,}5 \text{ ms}}$ **= 384 packets/s**		$\frac{80 \frac{\text{packets}}{\text{cycle}}}{500 \text{ ms}}$ **= 160 packets/s**	$\frac{100 \frac{\text{packets}}{\text{cycle}}}{250 \text{ ms}}$ **= 400 packets/s**	$\frac{100 \frac{\text{packets}}{\text{cycle}}}{125 \text{ ms}}$ **= 800 packets/s**
packet data payload per second **(PPS)**	384 packtes/s · 46 Byte = 17.664 Byte/s **(~17, 3 kByte/s)**		160 packets/s · 1.500 Byte = 240.000 Byte/s **(~234, 4 kByte/s)**	400 packets/s · 46 Byte = 18.400 Byte/s **(~18, 0 kByte/s)**	800 packets/s · 46 Byte = 36.800 Byte/s **(~35, 9 kByte/s)**
total frame payload per second **(FPS)**	384 p/s · 88 Byte = 33.792 Byte/s **(~33 kByte/s)**		160 p/s · 1.542 Byte = 246.720 Byte/s **(~240, 9 kByte/s)**	400 p/s · 88 Byte = 35.200 Byte/s **(~34, 3 kByte/s)**	800 p/s · 88 Byte = 70.400 Byte/s **(~68, 8 kByte/s)**

Note: Due to the different number of ETH/APL media converters available for testing Manufacturer A and B, the number of field device emulators and thus the distribution of PCC ('*Packet Count per Cycle*') may vary. However, the total sum of UDP traffic stays the same for both.

The combined PPC (‘*Packet data Payload per Cycle*’) of all traffic types stated in Table 10 shows that up to $128\ \mathrm{kByte}$ of data is generated every $500\ \mathrm{ms}$ by overlapping traffic types meeting the ‘*MinimumFrameMemory*’ test condition. The following figures show the packet-throughput behavior of the switch under the traffic parameters stated in Table 10.

Figure 21 and Figure 22 show the packet-throughput behavior of Manufacturer A in downstream direction. Figure 23 and Figure 24 illustrate the packet-throughput behavior of Manufacturer B in downstream direction.

The magenta line resembles ARP traffic at ~ 400 to $800\ \mathrm{packets/s}$. The green line represents TCP traffic $\sim 160\ \mathrm{packets/s}$. The blue line shows the entire UDP traffic captured, send by the workstation out to all field device emulators at ~ 192 to $256\ \mathrm{packets/s}$. The brown line shows the UDP traffic from one field device emulator at $16\ \mathrm{packets/s}$ for Manufacturer A and $96\ \mathrm{packets/s}$ for Manufacturer B.

Note: The total packet count of recorded UDP traffic in downstream direction only resembles around half of all field device emulators sending packets.

The TAP device which was used for traffic measurement was placed directly in between one APL switch spur port and one field device emulator (see Figure 16). However, the TAP device also measured other UDP traffic which was sent to neighboring field devices. This is due to the crosstalk nature of UDP traffic (see chapter 2.1.2). Besides measuring the packet count of one field device, the purpose of additional capturing the total UDP traffic is for better detection of stability issues, regarding packet-throughput and potential packet loss.

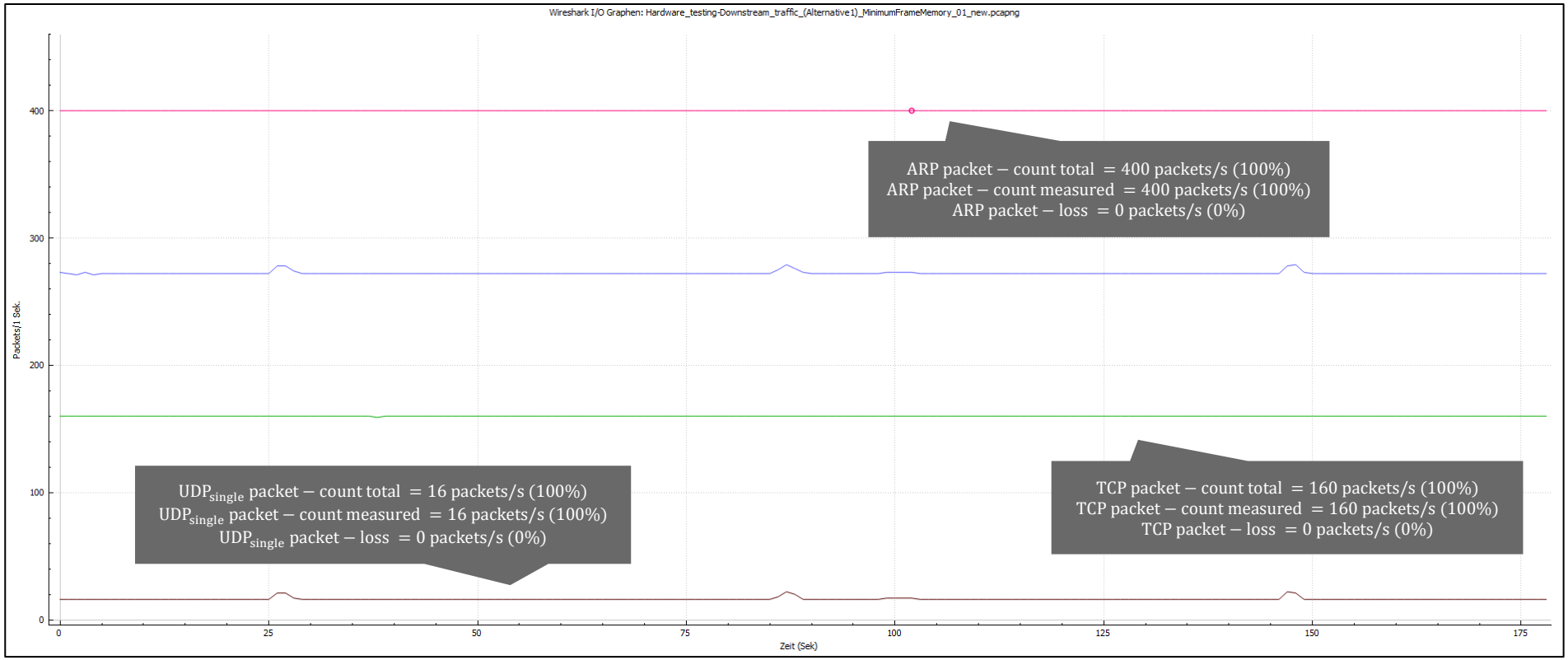

Figure 21: Downstream traffic analysis @ ***'MinimumFrameMemory'*** **condition (Alternative 1), Manufacturer A** - Measurement results (100 ARP packets / 250 ms)

Figure 21 shows that no traffic type experiences any packet loss at a PCS ('*Packet Count per Second*') of 400 packets/s for APL traffic. Hence, all packets of every traffic type have been forwarded successfully by the APL switch.

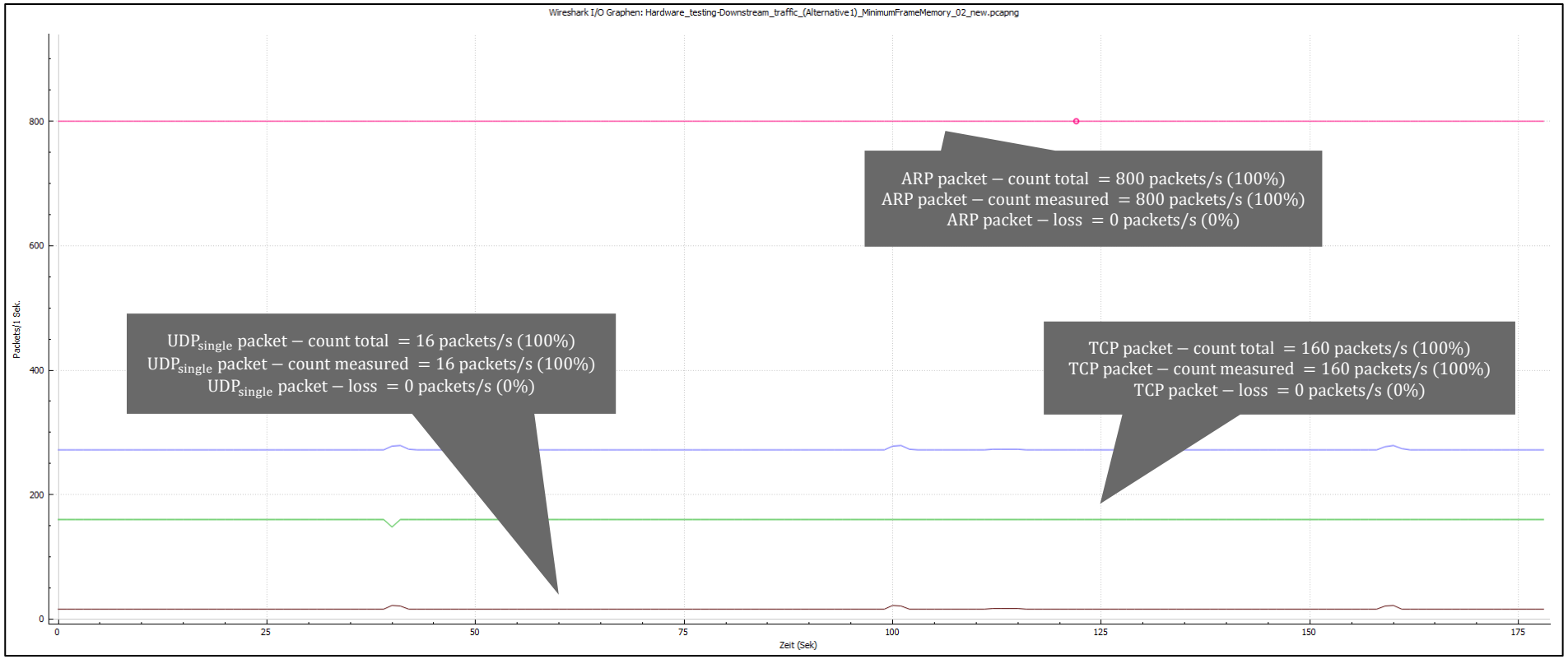

Figure 22: Downstream traffic analysis @ ***'MinimumFrameMemory'* condition (Alternative 1), Manufacturer A** - Measurement results (100 ARP packets / 125 ms)

Figure 22 shows that by doubling the PCS ('*Packet Count per Second*') for ARP-traffic from ~400 packets/s to ~800 packets/s, packet loss still does not occur for any traffic type. Hence, all packets of every traffic type have been forwarded successfully by the APL switch.

Summary (Figure 21 and Figure 22): The measurements show that stable packet-throughput of high-priority traffic, by gradual increase of ARP traffic up to ~800 packets/s, at a total packet load of 128 kByte, according to the '*MinimumFrameMemory*' condition, is always ensured.

The total packet count is 204 packets/cycle and consists of the PCC ('*Packet Count per Cycle*') of all traffic types, stated in Table 10. This value is below the '*bufferLength*' limit of the APL switch (1.800 packets/cycle). Therefore, no packet discarding occurs in the packet buffer of the APL switch (see chapter 5.1.1). Apart from being limited to a fixed number of total packets inside the packet buffer, the APL switch of Manufacturer A has a similar limit regarding its packet queues. If the APL switch does not forward incoming packets fast enough, subsequent packets of the next cycle can overlap with packets already queued. This

can then result in potential packet discards if the packet queues continuously getting filled until they reach their capacity limit, based on the '*queueLength*' limit (see chapter 5.1.2). Therefore, the PPT ('*Packet Processing Time*') needs to be reviewed, regarding potential packet discarding inside the packet queues.

The following PPTs have been calculated with the help of the hardware delay time stated in Table 2 (chapter 5.4) and the traffic parameters stated in Table 10.

$$PPT_{\text{type}} = x_{\text{packets,type}} \cdot \left(t_{\text{bridge}} + t_{\text{port}} + t_{\text{cable}} + t_{\text{prop}}\right) \quad (22)$$

Note: The store and forward bridge delay is calculated, based on the datarate of the internal Ethernet bridge of the APL switch, handling the transition between its Fast Ethernet ingress port, working at $100\ \text{Mbit/s}$, and Ethernet-APL egress spur port datarate working at $10\ \text{Mbit/s}$.

$$PPT_{\text{UDP}} = 24 \cdot \left(6\ \mu s + \frac{88\ Byte}{10\ Mbit/s} + 4\ \mu s + 1{,}33\ \mu s\right) \cong 2\ ms \quad (23)$$

$$\rightarrow (PPT_{\text{total}} = PPT_{\text{UDP}} = 2\ ms) < T_{UDP} = 62{,}5\ ms \quad (24)$$

Note: UDP traffic with the highest priority of '6' is processed first. By staying below the packet refresh rate of $62{,}5\ \text{ms}$ according to the UDP cycle time (the PPT stays below said rate), it can be ensured that there is no packet loss due to packet overlapping.

$$PPT_{\text{TCP}} = 80 \cdot \left(125\ \mu s + \frac{1542\ Byte}{10\ Mbit/s} + 4\ \mu s + 1{,}33\ \mu s\right) \cong 109\ ms \quad (25)$$

$$\rightarrow\ (PPT_{\text{total}} = 2 \cdot PPT_{\text{UDP}} + PPT_{\text{TCP}} = 113\ ms) < T_{\text{TCP}} = 500\ ms \quad (26)$$

Note: TCP traffic with the second highest priority of '4' gets processed second. The PPT for forwarding all TCP packets of one cycle also stays below its refresh-rate of $500\ \text{ms}$ also ensuring no packet loss.

While processing TCP traffic at a total PPT of $113\ \text{ms}$, other traffic types are being continuously send out based on their cycle time. Thus, UDP traffic arrives a total of two times at a refresh rate of $62{,}5\ \text{ms}$, while TCP packets are being processed. Due to UDP traffic being prioritized over TCP traffic said UDP traffic gets processed in between the processing of TCP traffic. Hence, the total PPT for TCP traffic increases by the PPT of two UDP cycles.

$$PPT_{\text{ARP}} = 100 \cdot \left(6\ \mu s + \frac{88\ Byte}{10\ Mbit/s} + 4\ \mu s + 1{,}33\ \mu s\right) \cong 8\ ms \quad (27)$$

$$\rightarrow\ (PPT_{\text{total}} = 2 \cdot PPT_{\text{UDP}} + PPT_{\text{TCP}} + PPT_{\text{ARP}} = 121\ ms) < T_{\text{ARP}} = 125\ ms \quad (28)$$

The calculated PPT for each respective traffic type shows that the switch hardware is fast enough in processing and forwarding all incoming packets of one cycle, thus preventing packet overlapping inside the packet queues. Additionally, the respective packet count of each traffic type stated by the PCC ('*Packet Count per Cycle*'), according to Table 10, stays below 128 packets/cycle.

Therefore, the packet count always stays below the '*queueLength*' limit of the APL switch (128 packets/cycle) so that no packet discarding happens in the packet queues of the APL switch (see chapter 5.1.2). In conclusion, the APL switch of Manufacturer A shows the desired packet processing behavior and fulfills the packet-throughput requirements according to Table 10.

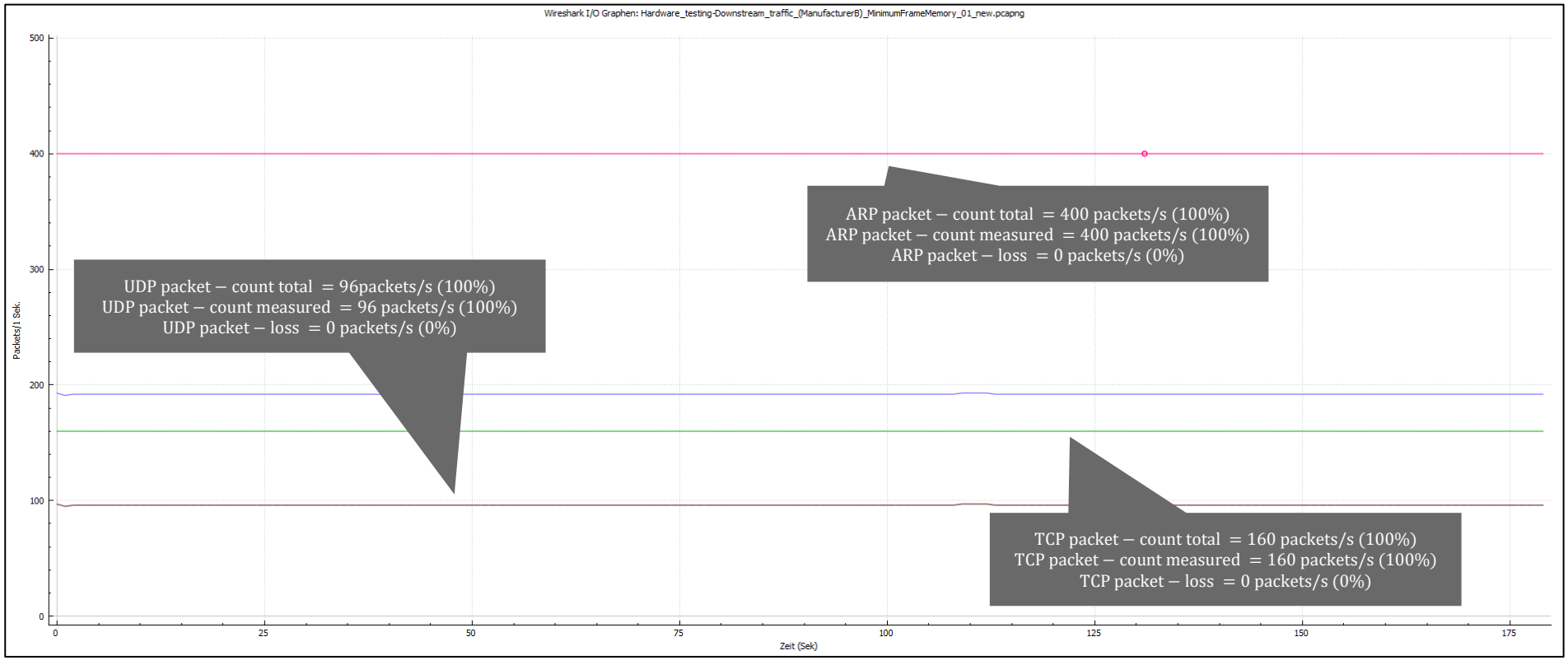

Figure 23: Downstream traffic analysis @ ***'MinimumFrameMemory'* condition (Alternative 1), Manufacturer B** - Measurement results (100 ARP packets / 250 ms)

Figure 23 shows that no traffic type experiences any packet loss at a PCS ('*Packet Count per Second*') of 400 packets/s for APL traffic. Hence, all packets of every traffic type have been forwarded successfully by the APL switch.

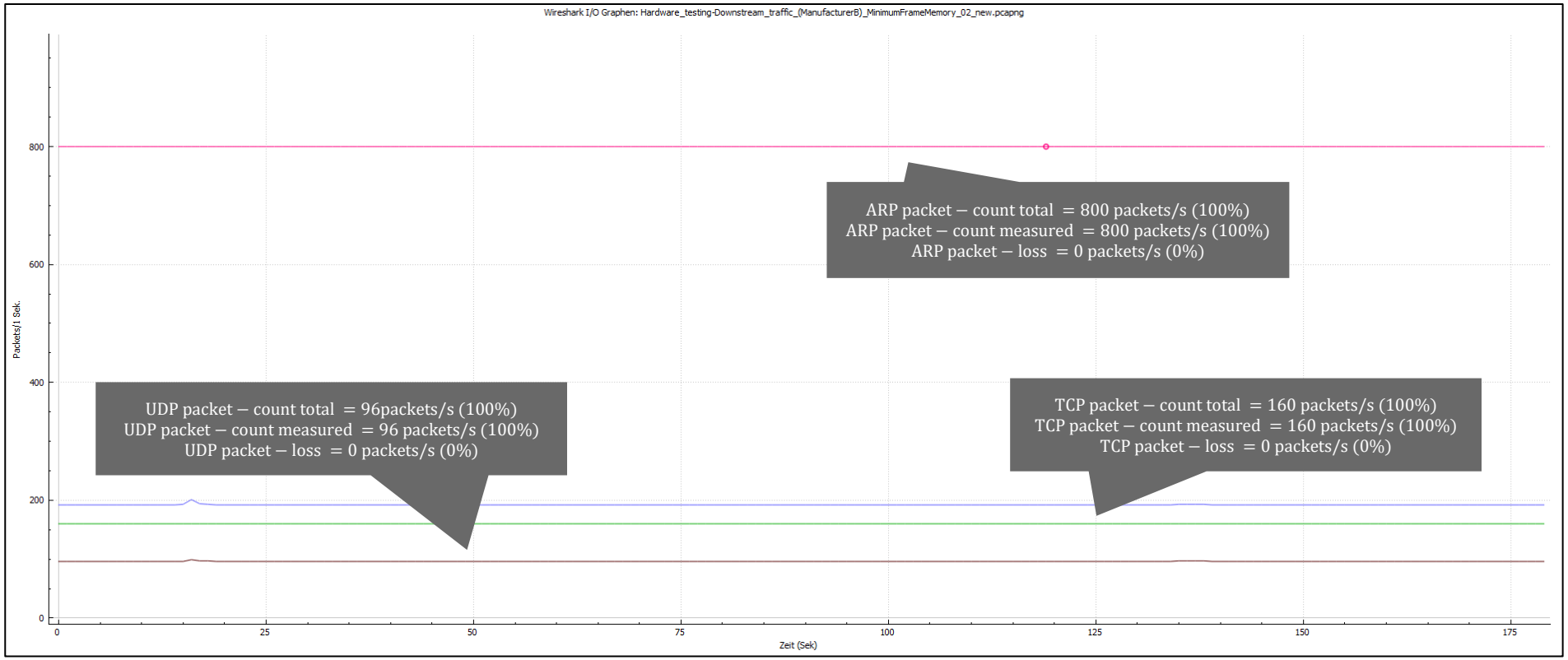

Figure 24: Downstream traffic analysis @ ***'MinimumFrameMemory'*** **condition (Alternative 1), Manufacturer B** - Measurement results (100 ARP packets / 125 ms)

Figure 24 shows that by doubling the PCS ('*Packet Count per Second*') for ARP-traffic from ~400 packet/s to ~800 packets/s, packet loss still does not occur for any traffic type. Hence, all packets of every traffic type have been forwarded successfully by the APL switch.

Summary (Figure 23 and Figure 24): The measurements show that stable packet-throughput of high-priority traffic, by gradual increase of ARP traffic up to ~800 packets/s, at a total packet load of 128 kByte, according to the '*MinimumFrameMemory*' condition, is always ensured.

Due to the total PCC ('*Packet Count per Cycle*') for UDP, TCP and APL traffic staying below the internal total '*bufferCount*' packet count limit (1024 packets/cycle) as well as the reserved packet count limit for each specific packet casting-type, packet loss does not occur (see chapter 5.2.1).
In conclusion, the APL switch of Manufacturer B shows the desired packet processing behavior and fulfills the packet-throughput requirements according to Table 10.

A.3.2 Packet-throughput analysis – absolute packet processing limit

The switch hardware intrinsically handles incoming packets according to packet prioritization based on the VLAN tag.

The IEEE 802.1Q standard specifies up to 8 different traffic types divided into classes ranging from 0 to 7. [IEEE_8021Q_05, p. 180 et seq.] The priority of each traffic type is not static and can be freely defined by the user.

However, common user priorities for PROFINET specific traffic are ‘6’ for RTC traffic and ‘0’ for IP (RPC via UDP) and DCP traffic. [PigRa_02, p. 64 et seq.] The user priorities ‘1...4’ are commonly not used for PROFINET specific traffic. Additionally, IP traffic via OPC UA / TCP also has no fixed user priority and can be freely defined by the user.

Note: The set VLAN-tags for testing purposes of each traffic type are:

- UDP (VLAN#: 6) priority over all other
- TCP (VLAN#: 4) prority over ARP
- ARP(VLAN#: 0) no priority

Each incoming packet is validated and handled according to its designated user priority. If the packet count of a specific traffic type exceeds the internal hardware limitations of the switch, packet prioritization still upholds the manageable packet count of higher prioritized packets before handling to the ones with lower priority. This ensures upholding of packet streams with high priority even in bursty traffic scenarios that would otherwise pose a potential risk of packet loss.

Hardware tests shall validate whether the switches manage to uphold stable packet-throughput of higher priority traffic. The absolute maximum packet processing limit is verified by gradually increasing the packet load generated for different traffic types. Figure 25 illustrates this approach and shows the gradual increase of low-priority ARP and TCP traffic, while maintaining high-priority UDP traffic at a constant packet load.

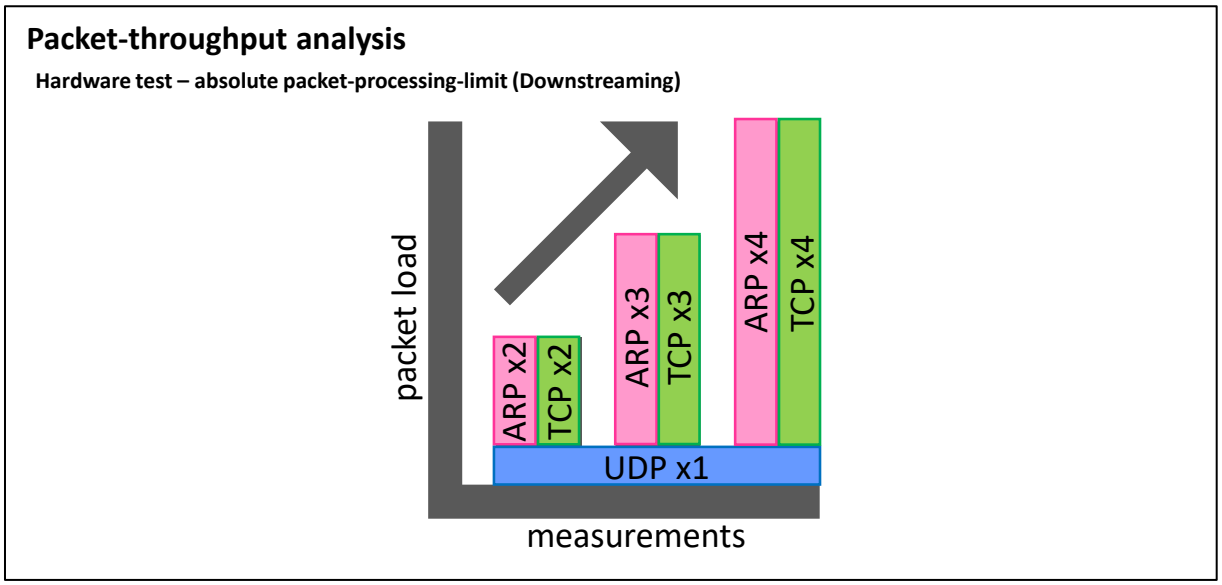

Figure 25: Gradual increase of low-priority traffic for investigation of absolute packet processing limit

A.3.2.1 Packet processing @ decreasing ARP cycle time

The following tests have been conducted by increasing ARP traffic through decreasing its cycle time. The traffic parameters used in these tests are stated in Table 11.

Table 11: Downstream traffic analysis (Alternative 1), Manufacturers A&B – Traffic parameters, increasing ARP traffic @ varying cycle time

	ARP request		
user priority	0		
total packet payload **(TPP)**	46 Byte (data) + 42 Byte (framing/transmission) = 88 Byte		
packet cycle time **(PCT)**	**62,5 / 31,25 / 15,625 ms**		
Packet Count per Cycle **(PCC)**	100 packets/cycle		
Packet data Payload per Cycle **(PPC)**	100 packets/cycle · 46 Byte = 4.600 Byte/cycle (~4,5 kByte/cycle)		
total frame payload per cycle **(FPC)**	100 packets/cycle · 88 Byte = 8.800 Byte/cycle (~8,6 kByte/cycle)		
Packet Count per Second **(PCS)**	**1.600 packets/s**	**3.200 packets/s**	**6.400 packets/s**
packet data payload per second **(PPS)**	1.600 packets/s · 46 Byte = 73.600 Byte/s (~**1,9 kByte/s**)	3.200 packets/s · 46 Byte = 147.200 Byte/s (~**143,8 kByte/s**)	6.400 packets/s · 46 Byte = 294.400 Byte/s (~**287,5 kByte/s**)
total frame payload per second **(FPS)**	1.600 packets/s · 88 Byte = 140.800 Byte/s (~**137,5 kByte/s**)	3.200 packets/s · 88 Byte = 281.600 Byte/s (~**275 kByte/s**)	6.400 packets/s · 88 Byte = 563.200 Byte/s (~**550 kByte/s**)

The following figures show the packet-throughput behavior of the switch when using the stated in Table 11. The parameters for all other traffic types (UDP, ARP) remained the same as stated in Table 10.

Figure 26 to Figure 31 show the packet-throughput behavior of Manufacturer A in downstream direction. Figure 32 to Figure 37 show the packet-throughput behavior of Manufacturer B in downstream direction.

The magenta line resembles ARP traffic at $\sim 1.600 \ldots 6.400$ packets/s. The green line represents TCP traffic ~ 160 packets/s. The blue line shows the entire captured UDP traffic send by the workstation out to all field device emulators at $\sim 192 \ldots 256$ packets/s. The brown line shows the UDP traffic from one field device emulator at 16 packets/s for Manufacturer A and 96 p/s for Manufacturer B.

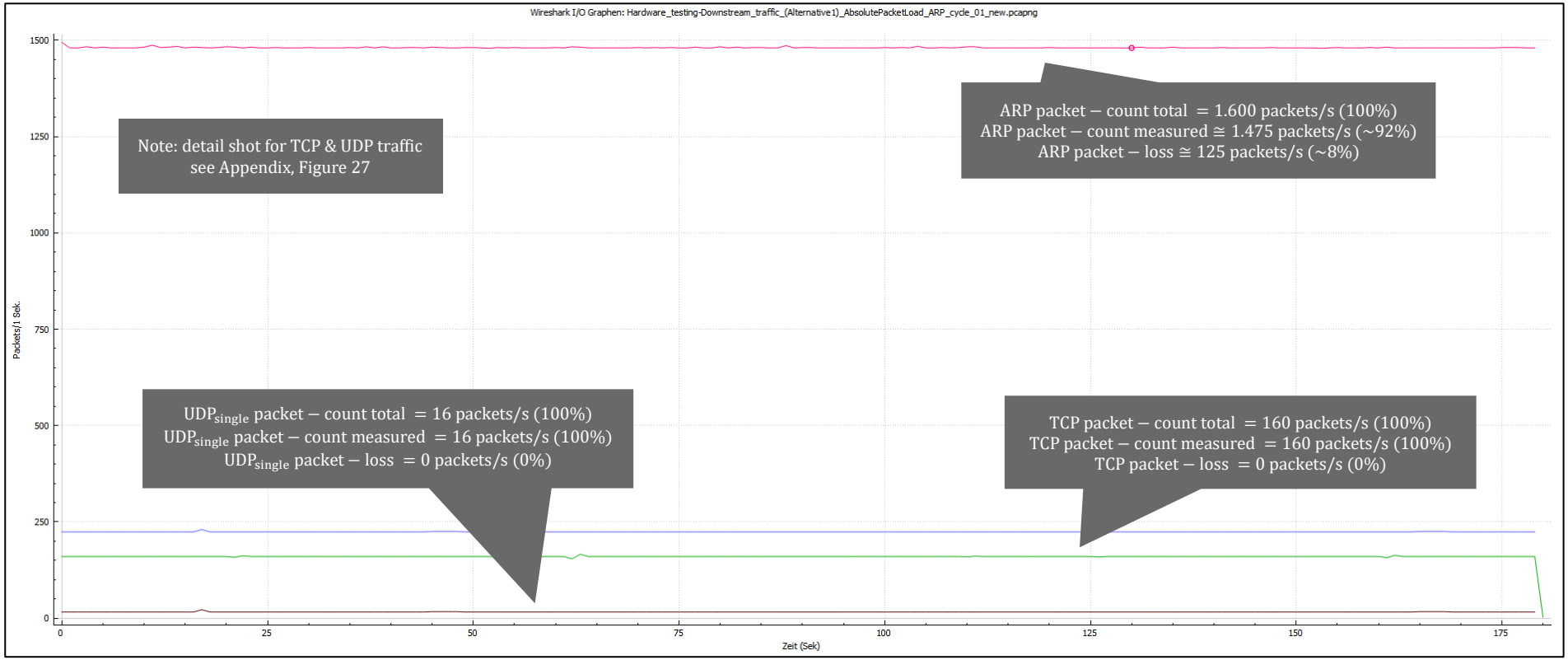

Figure 26: Downstream traffic analysis @ **decreasing ARP cycle time (Alternative 1), Manufacturer A** - Measurement results (100 ARP packets / 62,5 ms) – total packet count

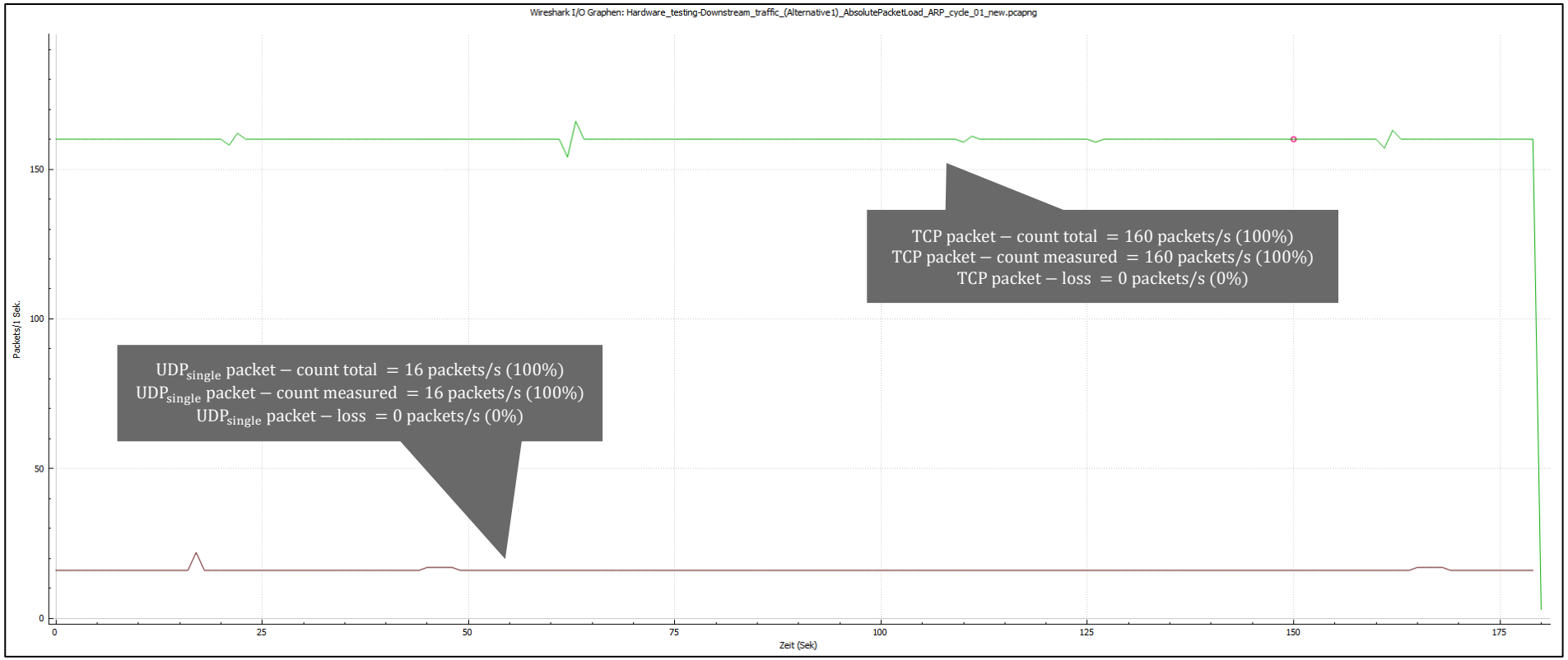

Figure 27: Downstream traffic analysis @ **decreasing ARP cycle time (Alternative 1), Manufacturer A** - Measurement results (100 ARP packets / 62,5 ms) – single source packet count

Figure 26 shows that ARP traffic has a packet loss of 8% at a PCS (‘*Packet Count per Second*’) of $1600\ \text{packets/s}$. However, UDP and TCP traffic are unaffected and show no packet loss. By dividing the ARP cycle time with the ARP PPT, according to formula (27) of the measurement summary in chapter A.3.1, the following total ARP packet count derives:

$$\rightarrow\ x_{\text{packets,ARP,1+8}} = \frac{T_{\text{ARP}} - PPT_{\text{UDP}} - PPT_{\text{TCP}}}{PPT_{\text{ARP}}} \tag{29}$$

$$\rightarrow x_{\text{packets,ARP,1+8}} = \frac{(62{,}5\ ms - 1{\cdot}2\ ms - 109\ ms)/\ cycle}{6\ \mu s + \frac{88\ Byte/packet}{10\ Mbit/s} + 4\ \mu s + 1{,}33\ \mu s} = \frac{\mathbf{0\ ms\ (1.+8.cycle)}}{6\ \mu s + 70{,}4\ \mu s + 4\ \mu s + 1{,}33\ \mu s}\frac{packets}{cycle} \cong 0\ \frac{packets}{cycle} \tag{30}$$

$$\rightarrow\ x_{\text{packets,ARP,2}} = \frac{T_{ARP} - PPT_{UDP} - [PPT_{TCP} - (T_{ARP} - PPT_{UDP})]}{PPT_{ARP}} \tag{31}$$

$$\rightarrow x_{\text{packets,ARP,2}} = \frac{[62{,}5\ ms - 1{\cdot}2\ ms - (109\ ms - 60{,}5ms)]/\ cycle}{6\ \mu s + \frac{88\ Byte/packet}{10\ Mbit/s} + 4\ \mu s + 1{,}33\ \mu s} = \frac{\mathbf{12\ ms}\ (\mathbf{2..cycle})}{6\ \mu s + 70{,}4\ \mu s + 4\ \mu s + 1{,}33\ \mu s}\frac{packets}{cycle} \cong 1.059\ \frac{packets}{cycle} \tag{32}$$

$$\rightarrow\ x_{\text{packets,ARP,3...8}} = \frac{T_{ARP} - T_{UDP}}{PPT_{ARP}} \tag{33}$$

$$\rightarrow x_{\text{packets,ARP,3...8}} = \frac{(62{,}5\ ms - 1{\cdot}2\ ms)/\ cycle}{6\ \mu s + \frac{88\ Byte/packet}{10\ Mbit/s} + 4\ \mu s + 1{,}33\ \mu s} = \frac{\mathbf{12\ ms}\ (\mathbf{3...7.cycle})}{6\ \mu s + 70{,}4\ \mu s + 4\ \mu s + 1{,}33\ \mu s}\frac{packets}{cycle} \cong 5.340\,\frac{packets}{cycle} \tag{34}$$

Note: While using an ARP cycle time of $62{,}5$ ms, UDP and TCP traffic with cycle times of $62{,}5$ ms and 500 ms respectively simultaneously arrives at the ingress port of the switch, where they are processed according to packet prioritization.

By dividing the TCP cycle time with the ARP or UDP cycle time, 8 different cycles subsequently repeating each other over and over again, can be observed based on the packet throughput behavior of the APL switch. Due to UDP and TCP traffic having a higher priority than ARP traffic, the processing time of ARP packets is prolonged into the subsequent cycle, which causes increased queueing delays for ARP packets:

- According to formula (30) the APL switch hardware could not be able to process any ARP packets each first and eight ARP cycle because they get delayed by the prioritized UDP- and TCP-traffic which gets processed first according to formula (23) and (25). Because the PPT of UDP and TCP traffic combined takes longer than the ARP-cycle time, said cycles are unable to process and forward any ARP packets which therefore stay queued inside the packet buffer of the switch.
- According to formula (32) the APL switch hardware could be able to process up to $1.059\ ARP\ packets$ each second ARP cycle in terms of PPT. This is possible due to the reduced packet count of already processed TCP packets of the previous cycle 1 or its repetition in cycle 8. Due to UDP traffic sharing the same cycle time of $62{,}5\ ms$ as ARP traffic, its packets delay the packet processing of ARP packets due to packet prioritization each cycle.
- According to formula (34) the APL switch hardware could be able to process up to $5.340\ ARP\ packets$ each third to seventh ARP cycle in terms of PPT. After reaching the third cycle the TCP packets repeating themselves every $500\ ms$ are completely processed by the previous cycles 1 and 2.

The calculation shows that the total packet count processable by the APL switch theoretically would grant a total ARP PPC of $1.059 \ldots 5.340\,\frac{\text{packets}}{\text{cycle}}$ depending on the cycle. This results in a PPS of $16.944 \ldots 85.440\ packets/s$. However, the actual measured ARP PCS shows a packet count of $\sim 1.475\,\frac{\text{packets}}{\text{s}}$. This indicates that the total PCS of $\sim 1.600\,\frac{packets}{s}$, according to Table 11, cannot be processed in time, thus resulting in a packet loss of ~ 125 packets/s.

Said packet loss of ARP packets happens due to the limited PPT available for processing the incoming ARP packets after forwarding the prioritized TCP and UDP packets. Furthermore, the total packet count of the incoming ARP packets per cycle cannot exceed the '*queueLength*' limit (128 packets/cycle) of the APL switch. All incoming packets which exceed the capacity of the already full packet queue get discarded by the APL switch (see chapter 5.1.2). Thus, the actual PPS calculates as follows:

$$packet_{\text{load,total}} = packet_{\text{count,ARP,1+8}} + packet_{\text{count,ARP,2}} + packet_{\text{count,ARP,3…7}} \quad (35)$$

$$\rightarrow PPS_{\text{UDP,total}} = 2 \cdot 0\,\frac{packets}{s} + 2 \cdot 128\,\frac{packets}{s} + 14 \cdot 100\,\frac{packets}{s} = 1.456\,\frac{packets}{s} \cong 1.480\,packets/s \quad (36)$$

The overall PPS is derived from three distinct PPS values from different cycles. Each cycle 100 ARP packets are generated according to Table 11.During the first and eighth cycle, no ARP packets are processed due to packet prioritization of TCP and UDP packets, according to formula (30).

This leads to packet overlapping of 100 ARP packets from the first and second cycle as well as the eighth and ninth cycle. Due to the '*queueLength*' limit, the packet queues cannot store more than 128 packets at the same time, thus causing packet discards of all excessive packets of the second and ninth cycle.

After successful processing and forwarding of all TCP packets, the PPT is again fast enough to process all 100 ARP packets for the 3rd…7th as well as the 10th…16th cycle. After 16 cycles, one second is reached which results in the total PPS of $1.456\,\frac{packets}{s}$.

In conclusion, the high-priority UDP and TCP traffic is still prioritized over the lower-priority ARP traffic. Hence, the packet prioritization of the APL switch works correctly.

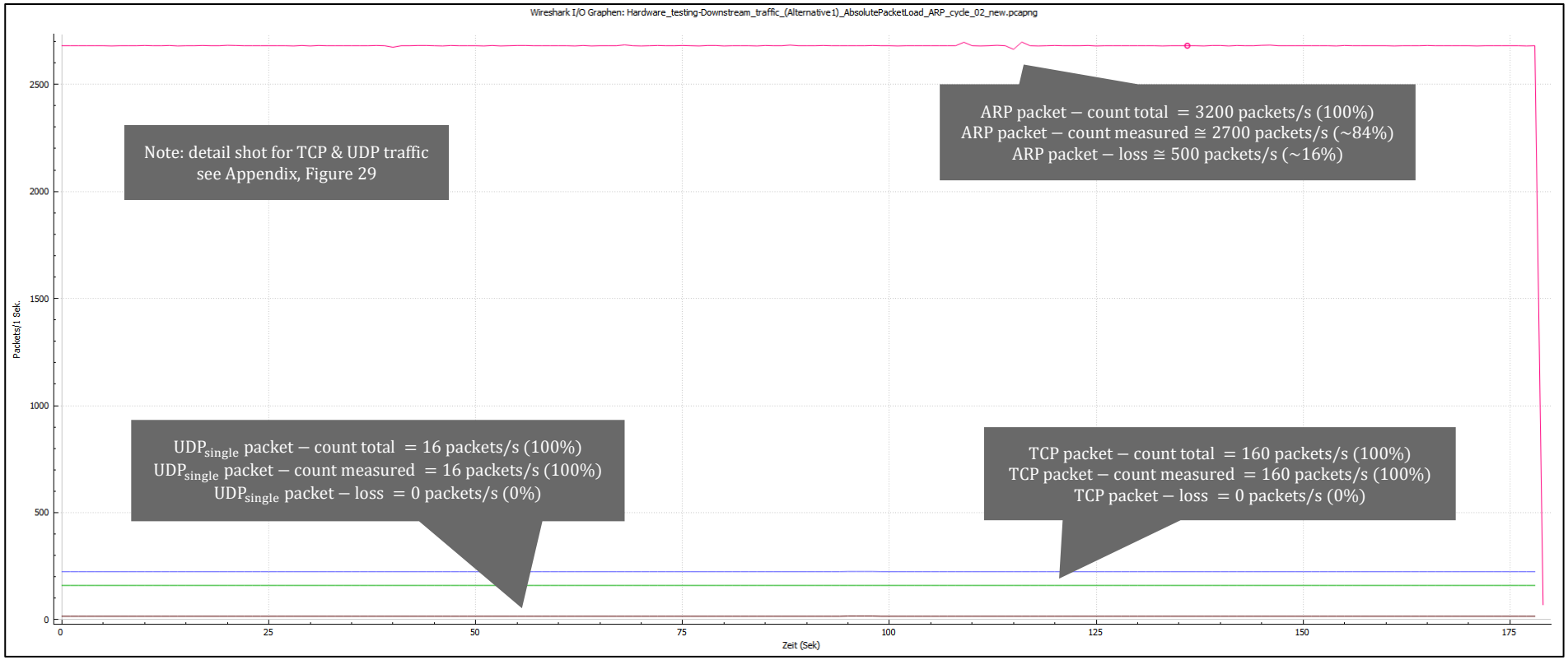

Figure 28: Downstream traffic analysis @ **decreasing ARP cycle time (Alternative 1), Manufacturer A** - Measurement results (100 ARP packets / 31,25 ms) – total packet count

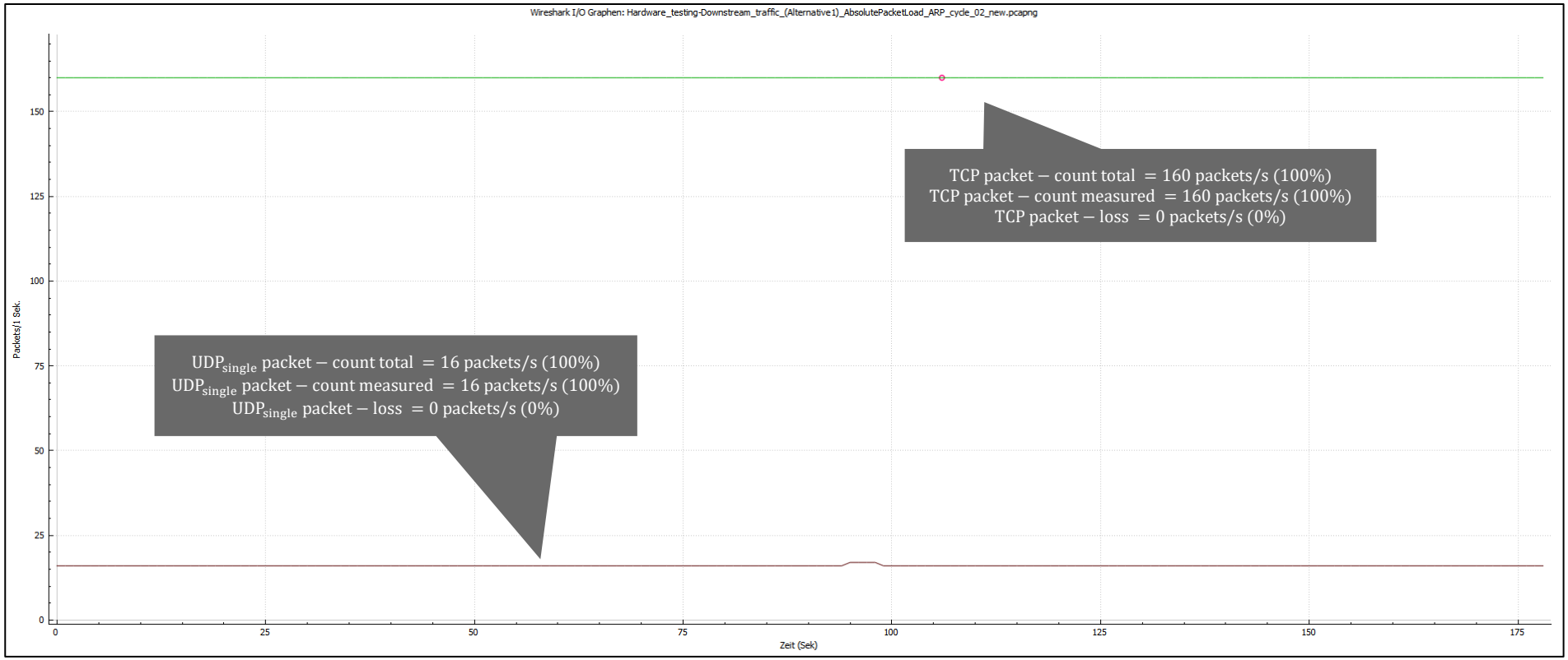

Figure 29: Downstream traffic analysis @ **decreasing ARP cycle time (Alternative 1), Manufacturer A** - Measurement results (100 ARP packets / 31,25 ms) – single source packet count

Figure 28 shows that by doubling the PCS (‘*Packet Count per Second*’) of ARP traffic from $\sim$1.600 packets/s to $\sim$3.200 packets/s, packet loss increases from 8% to 16%, while UDP and TCP traffic remain unaffected and show no packet loss. This shows that the same phenomenon which can be observed in the previous Figure 26 is also apparent in Figure 28, but in a more pronounced form. Hence, the packet prioritization of the APL switch works correctly.

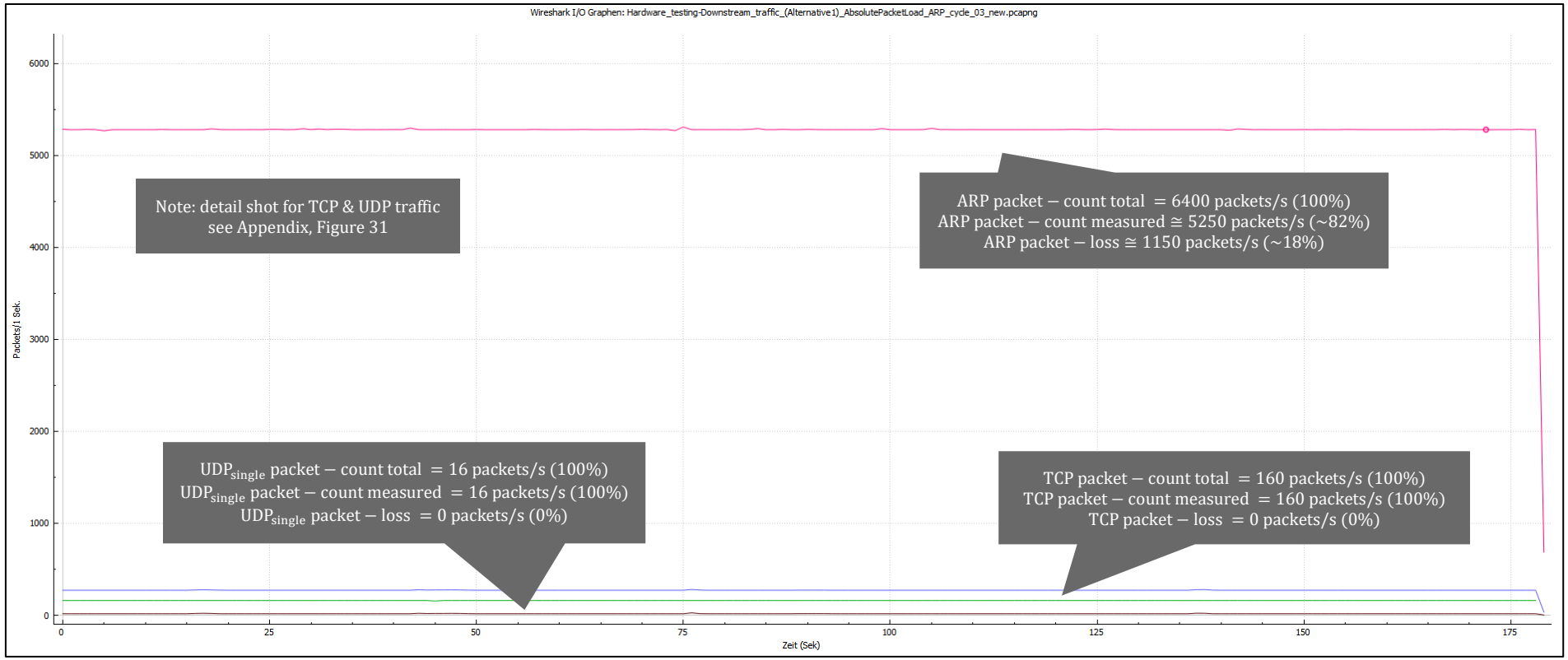

Figure 30: Downstream traffic analysis @ **decreasing ARP cycle time (Alternative 1), Manufacturer A** - Measurement results (100 ARP packets / 15,625 ms) – total packet count

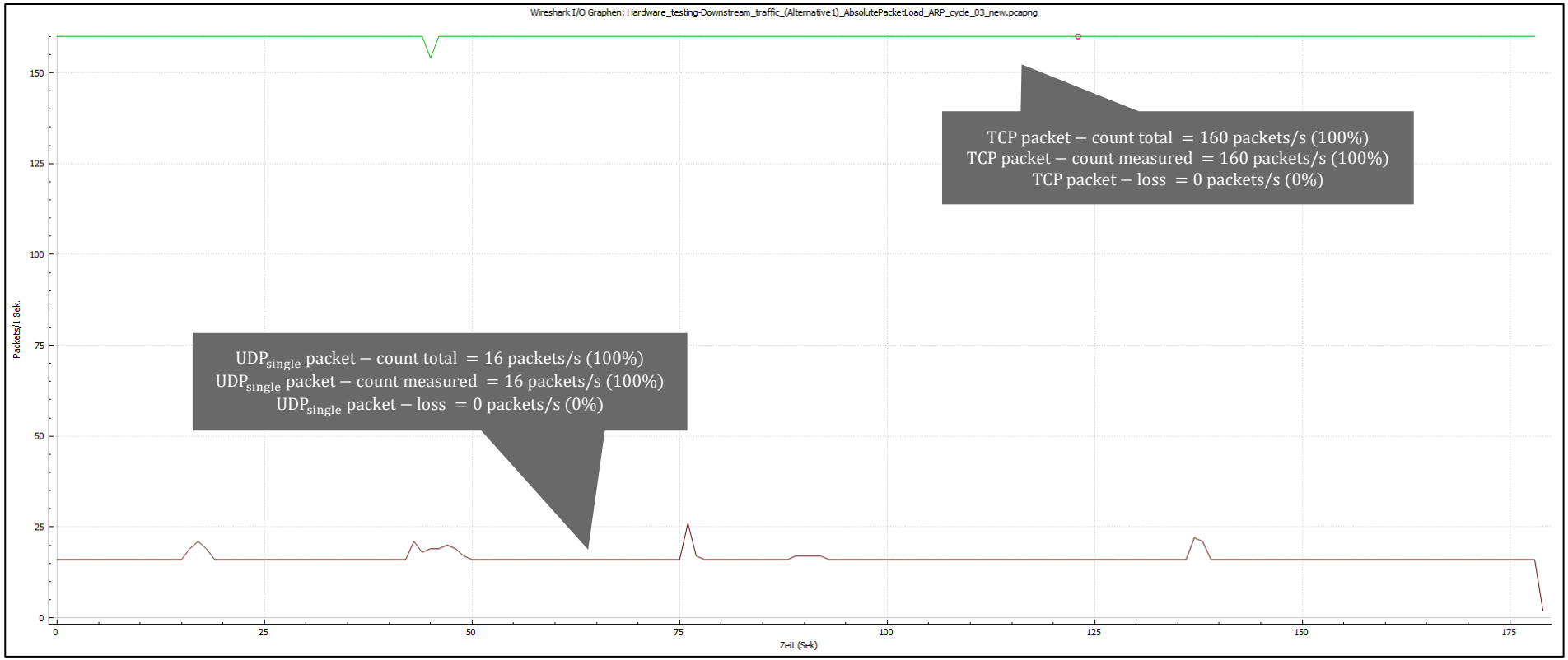

Figure 31: Downstream traffic analysis @ **decreasing ARP cycle time (Alternative 1), Manufacturer A** - Measurement results (100 ARP packets / 15,625 ms) – single source packet count

Figure 30 shows that by quadrupling the PCS (‘*Packet Count per Second*’) of ARP traffic from $\sim$1.600 packets/s to $\sim$6.400 packets/s, packet loss increases from 8% to 18%, while UDP and TCP traffic remain unaffected and show no packet loss. This shows that the same phenomenon which can be observed in the previous Figure 26 is also apparent in Figure 30, but in a more pronounced form. Hence, the packet prioritization of the APL switch works correctly.

Summary (Figure 26 to Figure 31): The measurements show that stable packet-throughput of high-priority traffic, by gradual increase of ARP traffic up to ~6.400 packets/s is always ensured.

The combined PPT ('*Packet Processing Time*') for subsequently forwarding all UDP, TCP and ARP packets of one cycle according to priority, exceeds the lowered cycle time of APL traffic, which leads to packet overlapping inside the ARP packet queue. Exceeding the '*queueLength*' limit of the switch, regarding packet count, leads to discarding of excessive packets arriving at the packet queue inside the switch (see chapter 5.1.2). However, the PCS ('*Packet Count per Second*') of UDP and TCP traffic shows that the APL switch works according to packet prioritization, managing to uphold high-priority UDP and TCP traffic without traffic loss, while sacrificing low priority ARP traffic instead.

In conclusion, the APL switch of Manufacturer A fulfills its packet-throughput requirements according to Table 11 regarding the desired packet processing behavior, regardless of working outside the constraints of its respective hardware limitations.

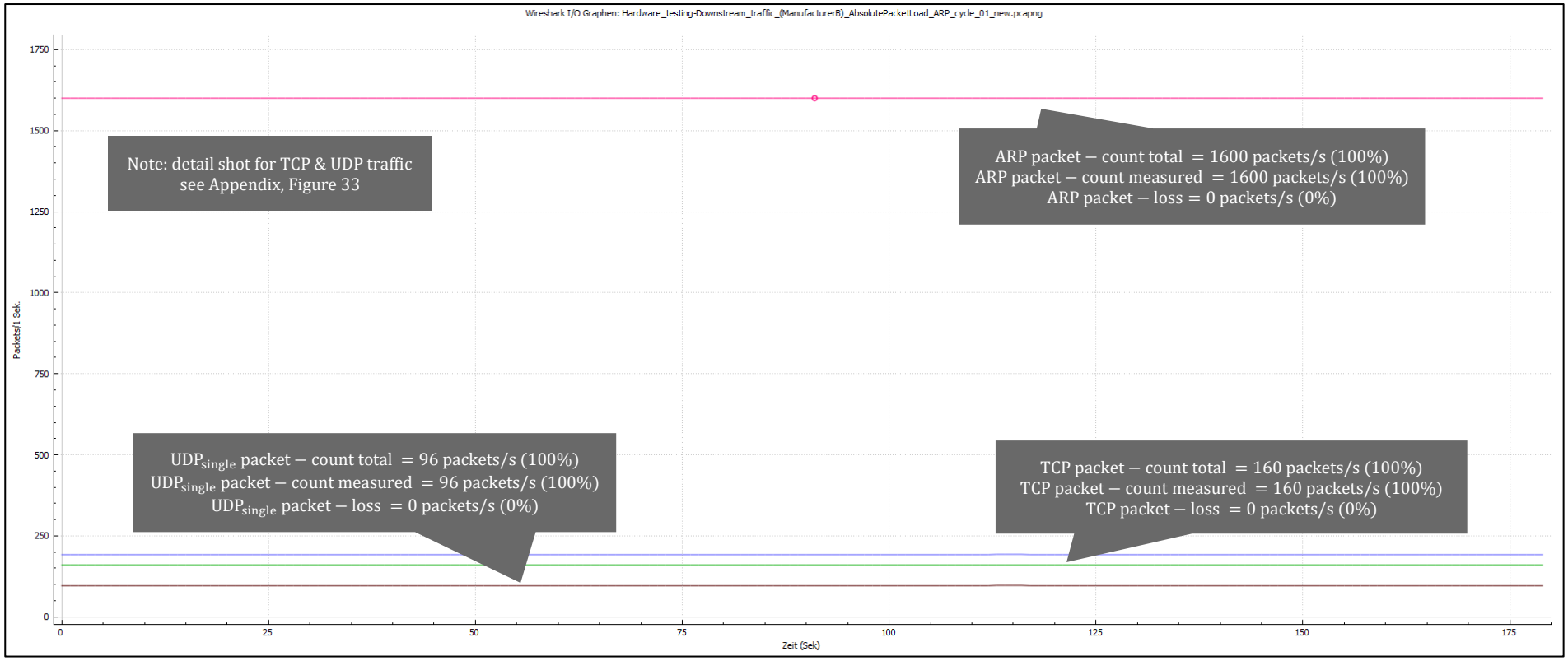

Figure 32: Downstream traffic analysis @ **decreasing ARP cycle time (Alternative 1), Manufacturer B** - Measurement results (100 ARP packets / 62,5 ms) – total packet count

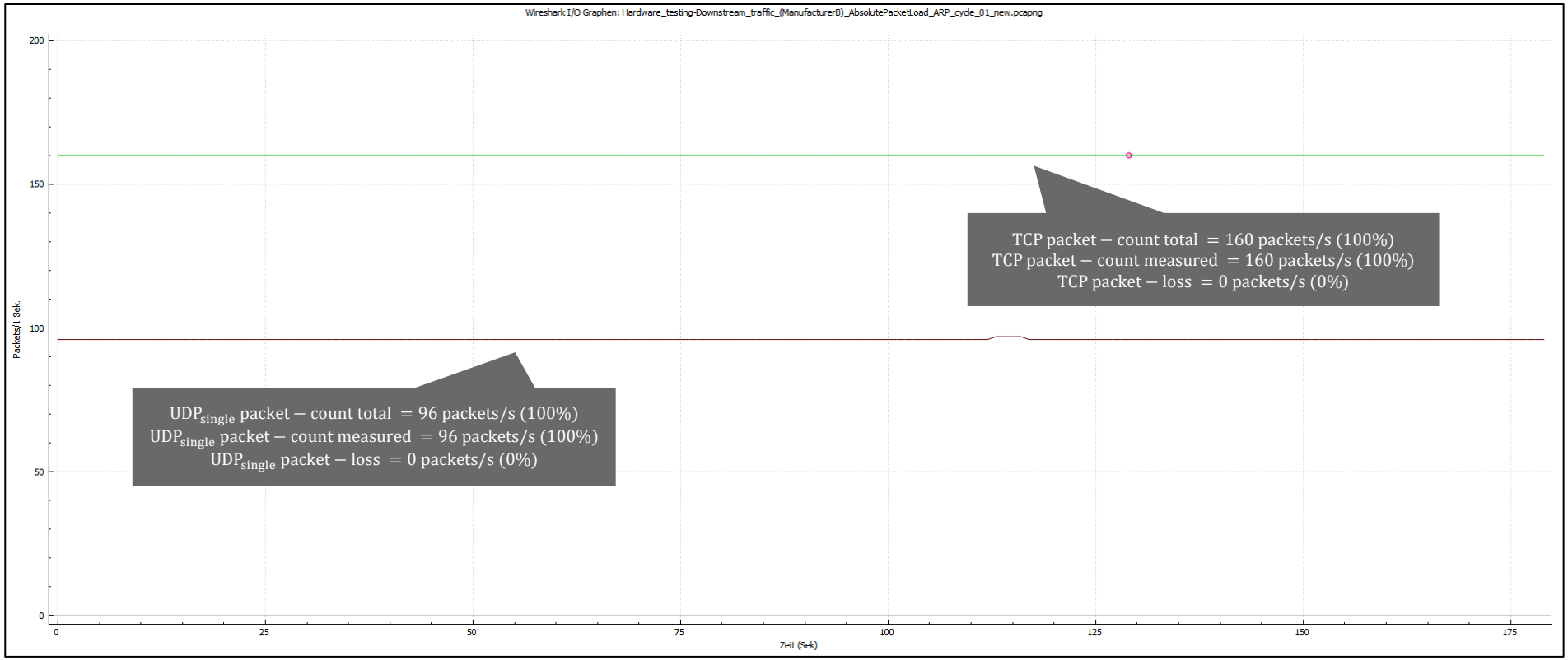

Figure 33: Downstream traffic analysis @ **decreasing ARP cycle time (Alternative 1), Manufacturer B** - Measurement results (100 ARP packets / 62,5 ms) – single source packet count

Figure 32 shows that no traffic type experiences packet loss at a APL traffic PCS ('*Packet Count per Second*') of 1.600 packets/s . Hence, all packets are forwarded successfully by the APL switch.

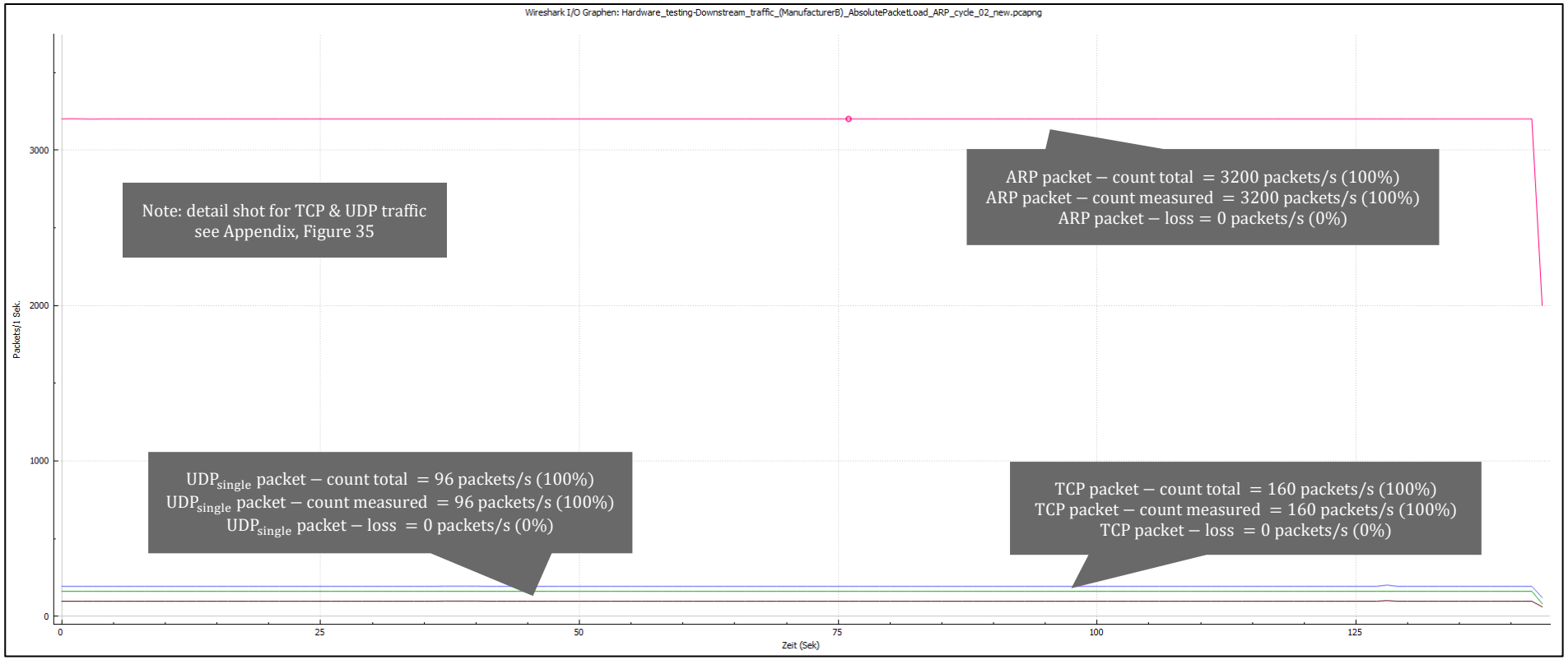

Figure 34: Downstream traffic analysis @ **decreasing ARP cycle time (Alternative 1), Manufacturer B** - Measurement results (100 ARP packets / 31,25 ms) – total packet count

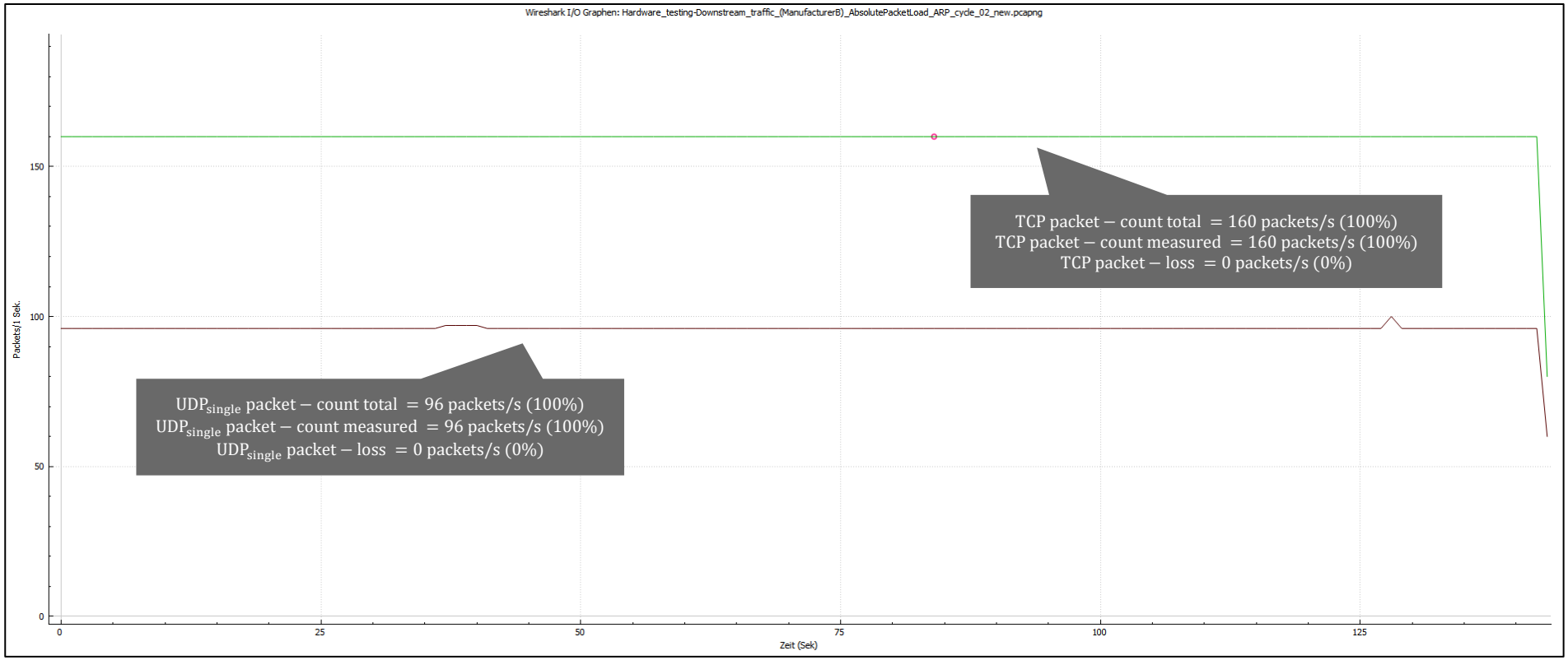

Figure 35: Downstream traffic analysis @ **decreasing ARP cycle time (Alternative 1), Manufacturer B** - Measurement results (100 ARP packets / 31,25 ms) – single source packet count

Figure 34 shows that by doubling the PCS ('*Packet Count per Second*') of ARP traffic from $\sim$1.600 packets/s to $\sim$3.200 packets/s, packet loss still does not occur. Hence, all packets are forwarded successfully by the APL switch.

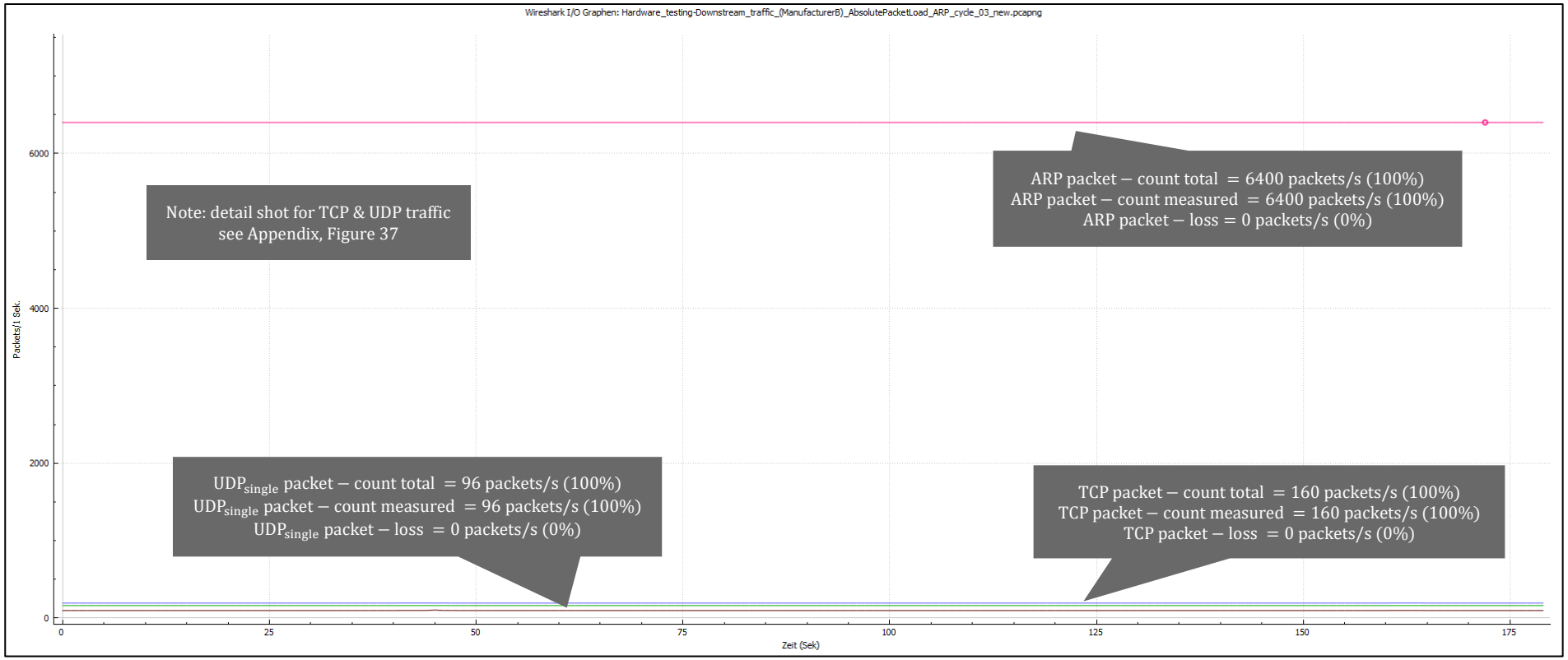

Figure 36: Downstream traffic analysis @ **decreasing ARP cycle time (Alternative 1), Manufacturer B** - Measurement results (100 ARP packets / 15,625 ms) – total packet count

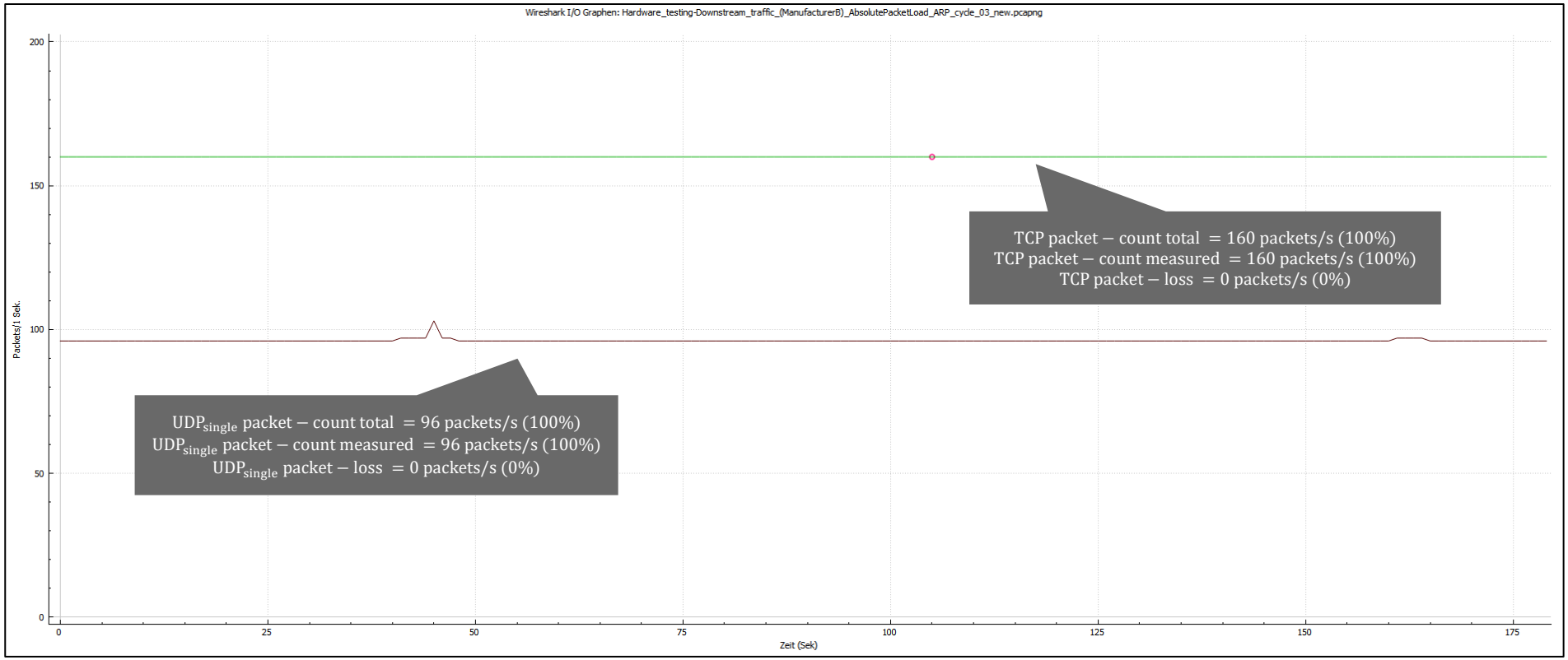

Figure 37: Downstream traffic analysis @ **decreasing ARP cycle time (Alternative 1), Manufacturer B** - Measurement results (100 ARP packets / 15,625 ms) – single source packet count

Figure 36 shows that by quadrupling the PCS ('*Packet Count per Second*') of ARP traffic from ~ 1.600 packets/s to ~ 6.400 packets/s, packet loss still does not occur. Hence, all packets are forwarded successfully by the APL switch.

Summary (Figure 32 to Figure 37): The measurements show, that stable packet-throughput of high-priority traffic is ensured, when gradually increasing ARP traffic up to $\sim 6.400\ packets/s$. Due to UDP, TCP and APL traffic's total PCS ('*Packet Count per Cycle*') staying below the internal total '*bufferCount*' packet count limit (1.024 packets/cycle) as well as the reserved packet count limit for each specific packet casting-type, packet loss does not occur (see chapter 5.2.1). In conclusion, the APL switch of Manufacturer B fulfills its packet-throughput requirements according to Table 11 regarding the desired packet processing behavior.

A.3.2.2 Packet processing @ increasing ARP packet count

The following tests have been conducted by increasing ARP traffic through increasing its packet count. The traffic parameters used in these tests are stated in Table 12.

Table 12: Downstream traffic analysis (Alternative 1), Manufacturers A&B – Traffic parameters, increasing ARP traffic @ varying *Packet Count per Cycle*

	ARP request		
user priority	0		
total packet payload **(TPP)**	46 Byte (data) + 42 Byte (framing/transmission) = 88 Byte		
packet cycle time **(PCT)**	125 ms		
Packet Count per Cycle **(PCC)**	**200 / 400 / 800 packets / cycle**		
Packet data Payload per Cycle **(PPC)**	~8,9 /~18 / ~35,9 kByte/cycle		
total frame payload per cycle **(FPC)**	~17,2 / ~34,4 / ~68,8 kByte/cycle		
Packet Count per Second **(PCS)**	**1.600 packets/s**	**3.200 packets/s**	**6.400 packets/s**
packet data payload per second **(PPS)**	1.600 packets/s · 46 Byte = 73.600 Byte/s (~**z1, 9 kByte/s**)	3.200 packets/s · 46 Byte = 147.200 Byte/s (~**143, 8 kByte/s**)	6.400 packets/s · 46 Byte = 294.400 Byte/s (~**287, 5 kByte/s**)
total frame payload per second **(FPS)**	1.600 packets/s · 88 Byte = 140.800 Byte/s (~**137, 5 kByte/s**)	3.200 packets/s · 88 Byte = 281.600 Byte/s (~**275 kByte/s**)	6.400 packets/s · 88 Byte = 563.200 Byte/s (~**550 kByte/s**)

The following figures show the packet-throughput behavior of the switch when using the stated in Table 12. The parameters for all other traffic types (UDP, ARP) remained the same as stated in Table 10.

Figure 38 to Figure 43 show the packet-throughput behavior of Manufacturer A in downstream direction. Figure 44 to Figure 49 show the packet-throughput behavior of Manufacturer B in downstream direction.

The magenta line resembles ARP traffic at $\sim 1.600 \ldots 6.400$ packets/s. The green line represents TCP traffic ~ 160 packets/s. The blue line shows the entire captured UDP traffic send by the workstation out to all field device emulators at $\sim 192 \ldots 256$ pakets/s. The brown line shows the UDP traffic from one field device emulator at 16 pakets/s for Manufacturer A and 96 packets/s for Manufacturer B.

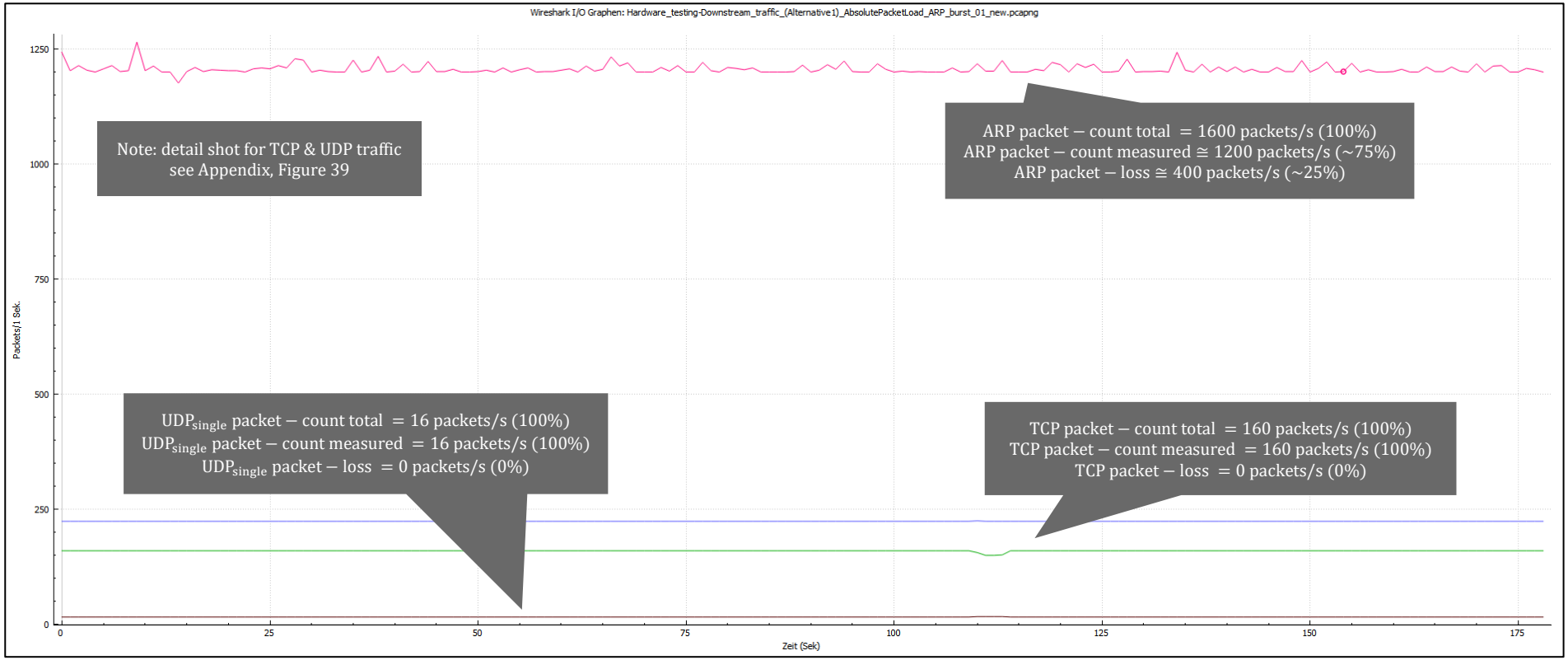

Figure 38: Downstream traffic analysis @ **increasing ARP packet count (Alternative 1), Manufacturer A** - Measurement results (200 ARP packets / 125 ms) – total packet count

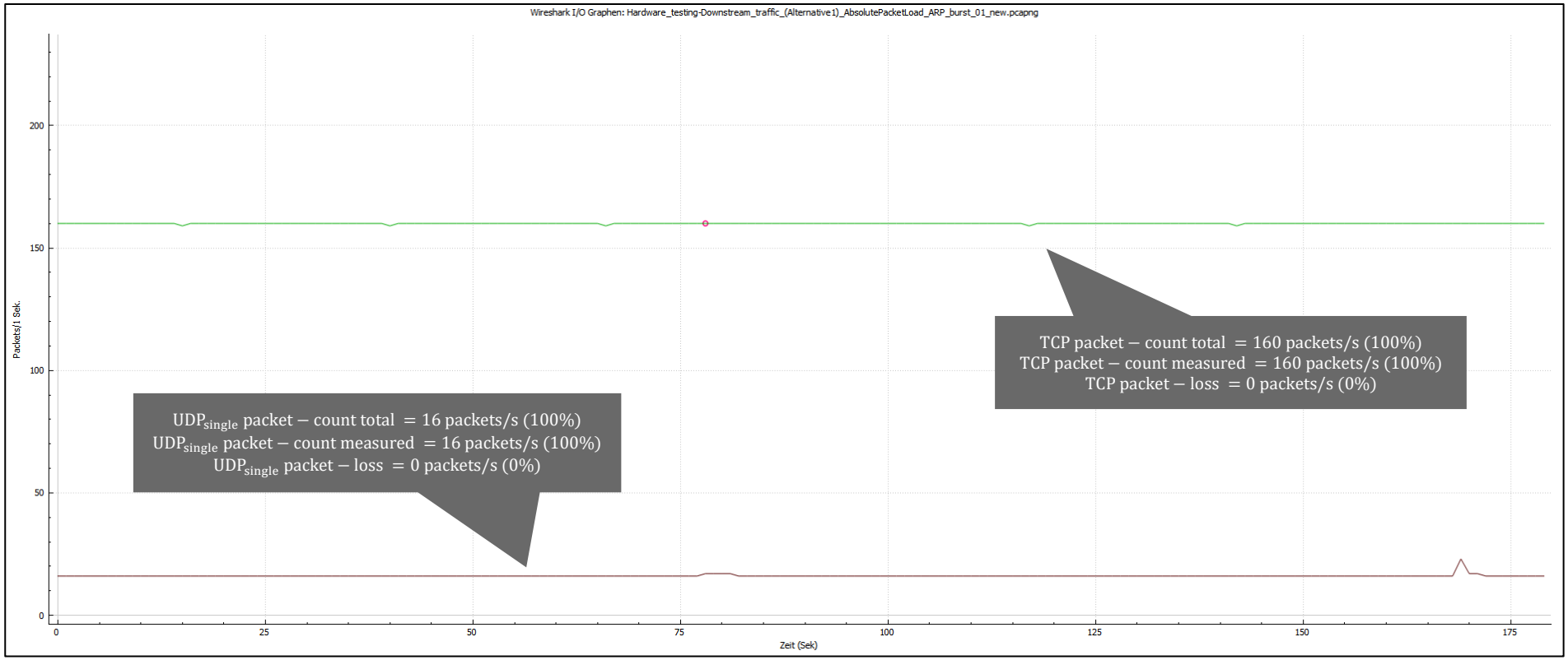

Figure 39: Downstream traffic analysis @ **increasing ARP packet count (Alternative 1), Manufacturer A** - Measurement results (200 ARP packets / 125 ms) – single source packet count

Figure 39 shows that ARP traffic has a packet loss of 25% at a PCS ('*Packet Count per Second*') of 1600 packets/s . However, UDP and TCP traffic are unaffected and show no packet loss. By dividing the ARP cycle time with the ARP PPT, according to formula (27) of the measurement summary in chapter A.3.1, the following total ARP packet count derives:

$$\rightarrow x_{\text{packets,ARP,1...8}} = \frac{T_{\text{ARP}} - PPT_{\text{TCP}} - 2 \cdot PPT_{\text{UDP}}}{PPT_{\text{ARP}}} \tag{37}$$

$$\rightarrow x_{\text{packets,ARP,1...8}} = \frac{(125\ ms - 109\ ms - 2 \cdot 2\ ms)/\ cycle}{6\ \mu s + \frac{88\ Byte/packet}{10\ Mbit/s} + 4\ \mu s + 1{,}33\ \mu s} = \frac{\mathbf{12\ ms\ (1...8.cycle)}}{6\ \mu s + 70{,}4\ \mu s + 4\ \mu s + 1{,}33\ \mu s} \frac{packets}{cycle} \cong 1.059\ \frac{packets}{cycle} \tag{38}$$

Note: While using an ARP cycle time of $125\ \mathrm{ms}$, UDP with a cycle time of $62{,}5\ \mathrm{ms}$ simultaneously arrives a total of two times at the ingress port of the switch, where they are processed according to packet prioritization.

By dividing the TCP cycle time with the UDP cycle time, 8 different cycles subsequently repeating each other over and over again, can be observed based on the packet throughput behavior of the APL switch. Due to UDP having a higher priority than ARP traffic, the processing time of ARP packets is delayed until all UDP packets are forwarded.

- According to formula (38) the APL switch hardware could be able to process up to $1.059\ ARP\ packets$ each ARP cycle in terms of PPT. This is possible because the PPT of all traffic types fits into the ARP cycle time of $125\ ms$.

The calculation shows that the total packet count processable by the APL switch theoretically would grant a total ARP PPC of $1.059\,\frac{\mathrm{packets}}{\mathrm{cycle}}$. This results in a PPS of $\sim 16.944\ \mathrm{packets/s}$. However, the actual measured ARP PCS shows a packet count of $\sim 1.200\,\frac{\mathrm{packets}}{\mathrm{s}}$. This indicates that the total PCS of $\sim 1.600\,\frac{packets}{s}$, according to Table 12, cannot be processed in time, thus resulting in a packet loss of $\sim 400\ \mathrm{packets/s}$.

Said packet loss of ARP packets happens due to the packet count of incoming ARP packets each cycle which exceed the '*queueLength*' limit ($128\ \mathrm{packets/cycle}$) of the APL switch. All incoming packets which exceed the capacity of the already full packet queue get discarded by the APL switch (see chapter 5.1.2).

However, the high-priority UDP and TCP traffic are still being prioritized over the lower-priority ARP traffic. Hence, the packet prioritization of the APL switch works correctly.

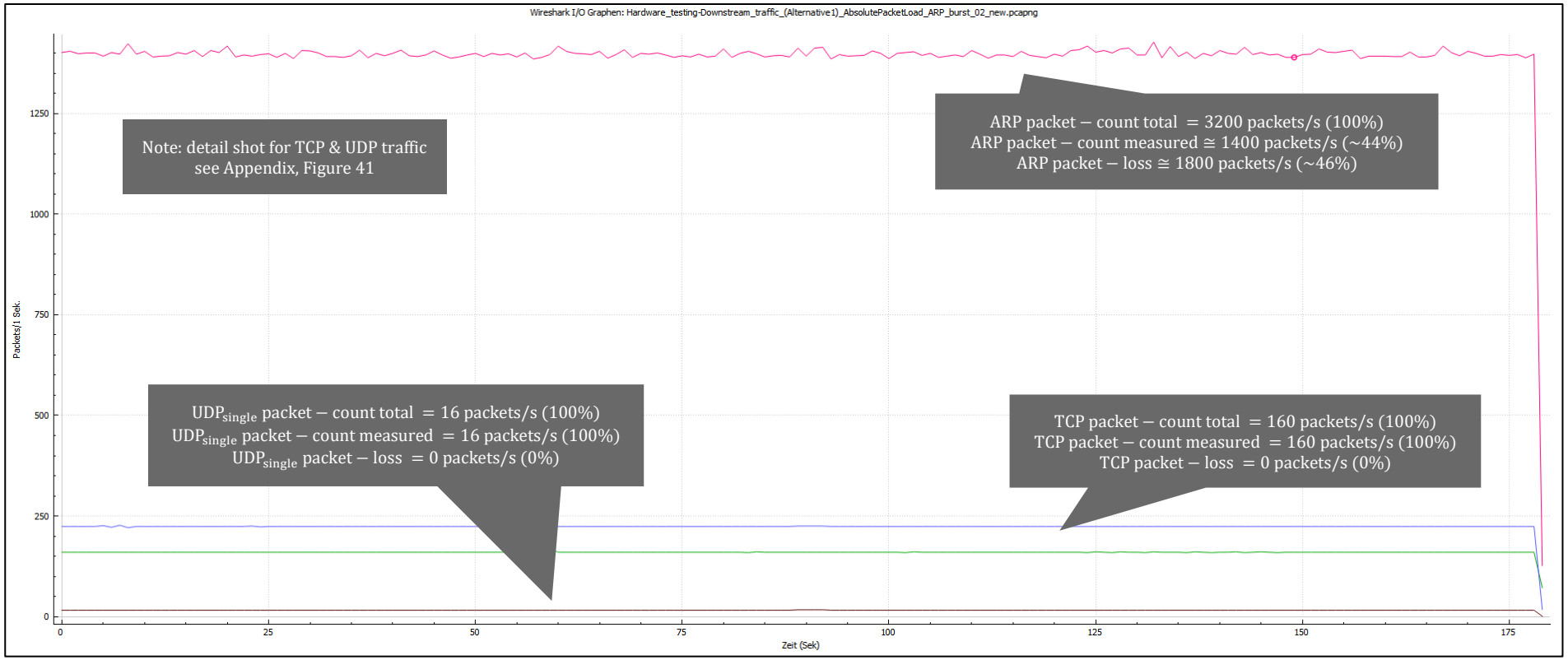

Figure 40: Downstream traffic analysis @ **increasing ARP packet count (Alternative 1), Manufacturer A** - Measurement results (400 ARP packets / 125 ms) – total packet count

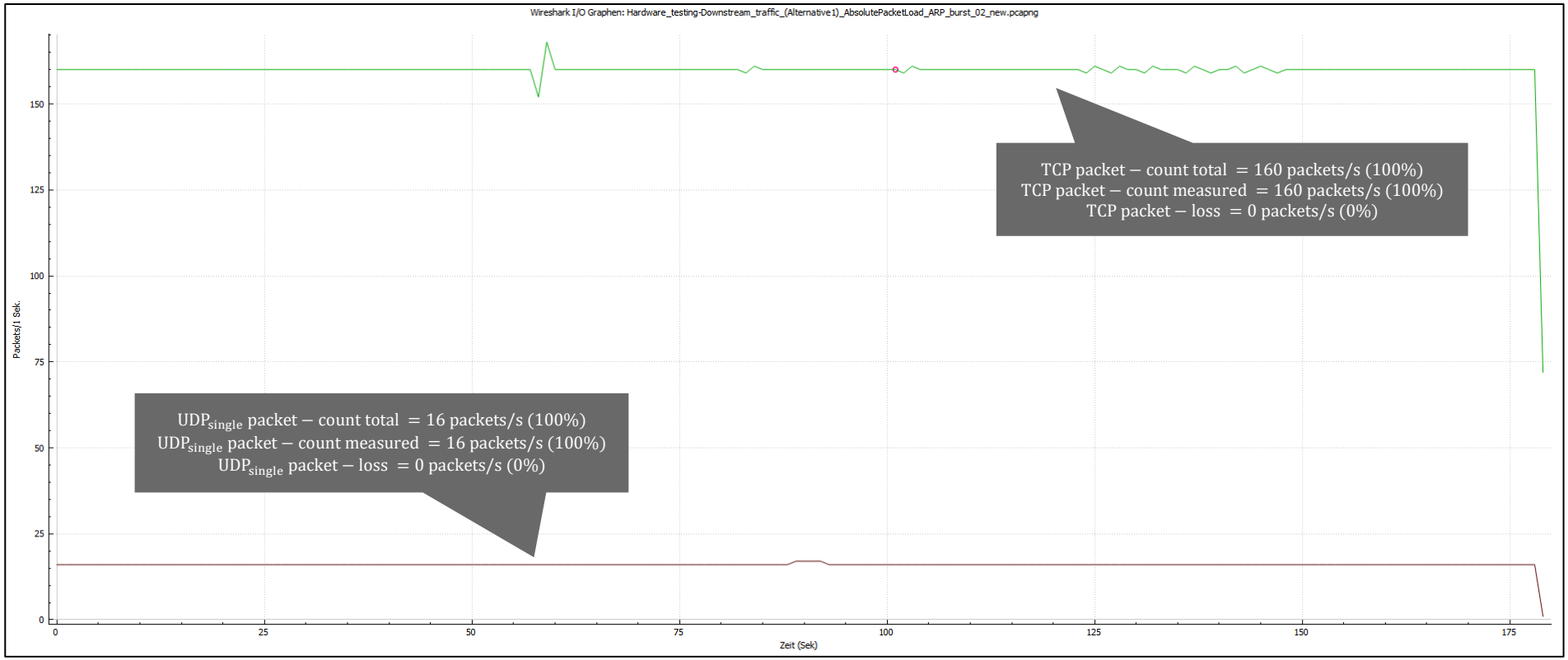

Figure 41: Downstream traffic analysis @ **increasing ARP packet count (Alternative 1), Manufacturer A** - Measurement results (400 ARP packets / 125 ms) – single source packet count

Figure 40 shows that by doubling the PCS ('*Packet Count per Second*') of ARP traffic from ~1.600 packets/s to ~3.200 packets/s, packet loss increases from 25% to 46%, while UDP and TCP traffic remain unaffected and show no packet loss. This shows that the same phenomenon which can be observed in the previous Figure 38 is also apparent in Figure 40, but in a more pronounced form. Hence, the packet prioritization of the APL switch works correctly.

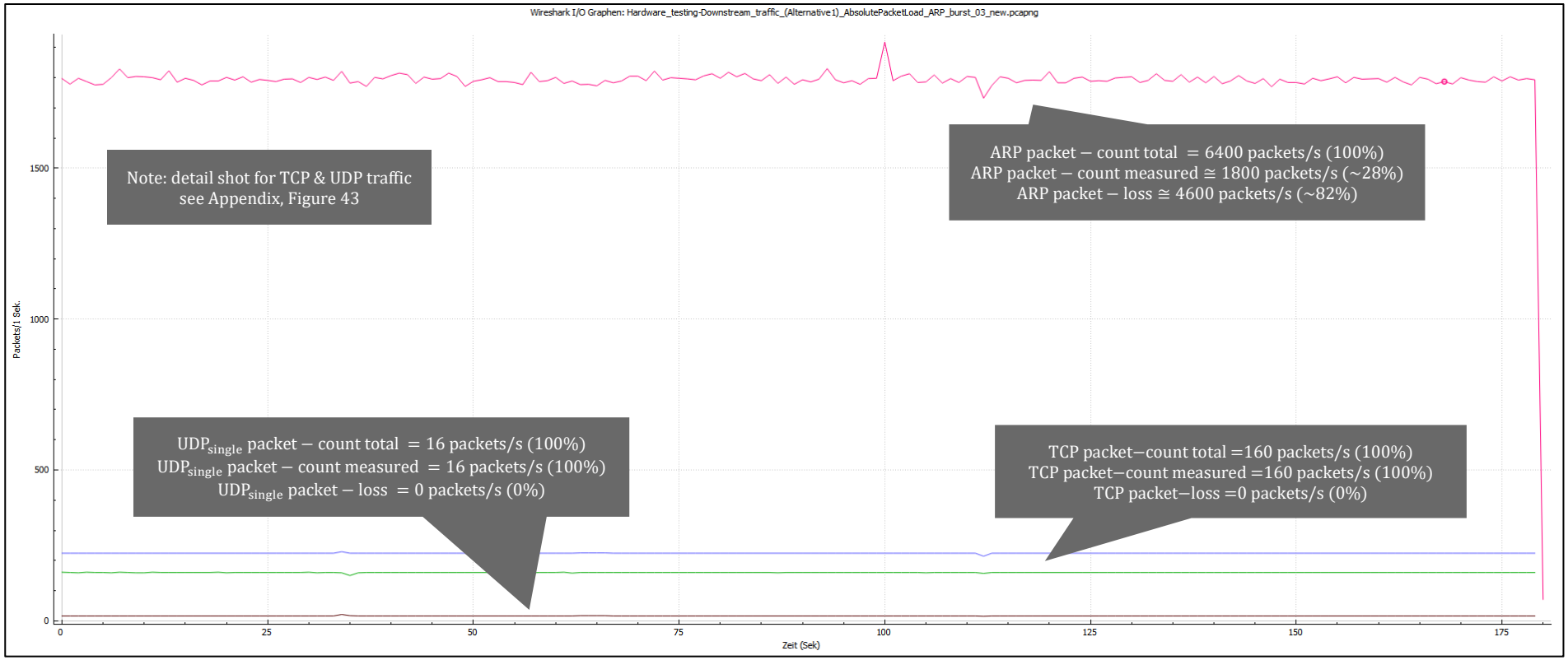

Figure 42: Downstream traffic analysis @ **increasing ARP packet count (Alternative 1), Manufacturer A** - Measurement results (800 ARP packets / 125 ms) – total packet count

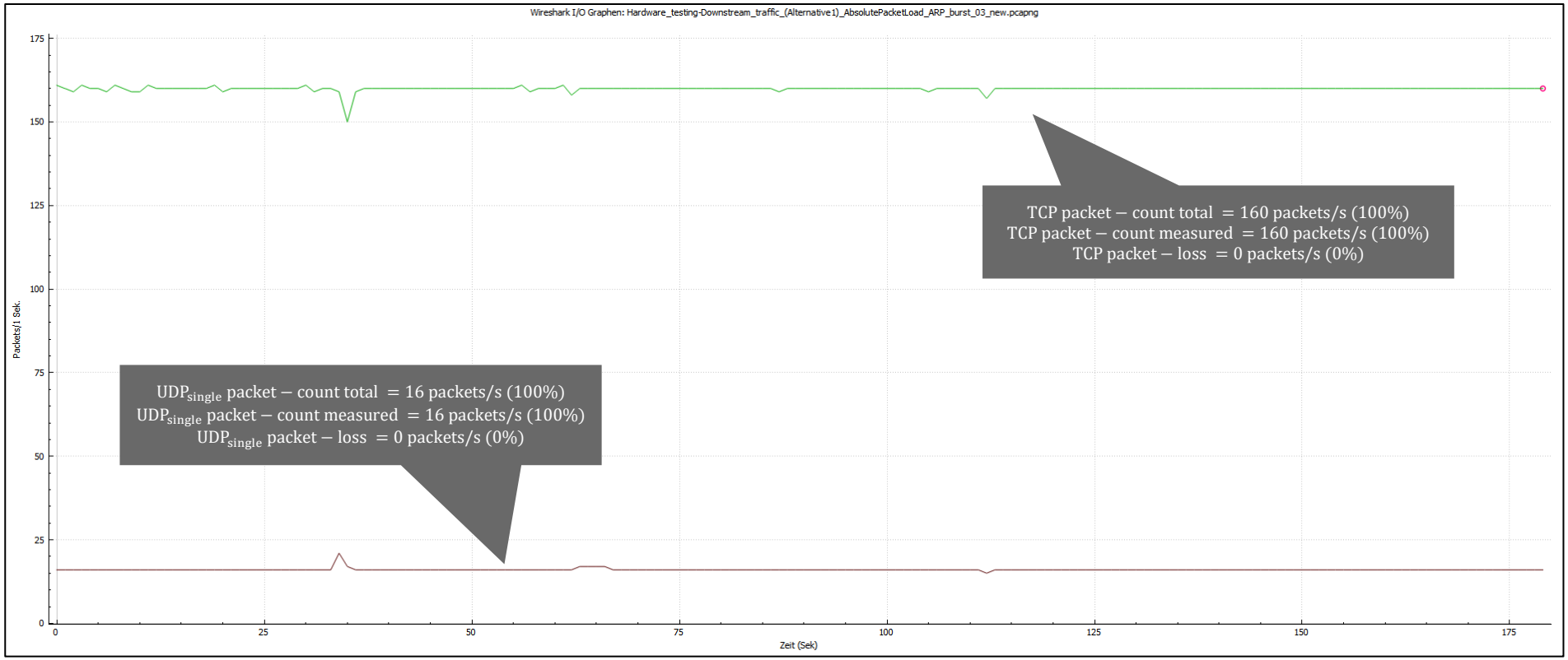

Figure 43: Downstream traffic analysis @ **increasing ARP packet count (Alternative 1), Manufacturer A** - Measurement results (800 ARP packets / 125 ms) – single source packet count

Figure 42 shows that by quadrupling the PCS (‘*Packet Count per Second*’) of ARP traffic from $\sim$1.600 packets/s to $\sim$6.400 packets/s, packet loss increases from 25% to 82%, while UDP and TCP traffic remain unaffected and show no packet loss. This shows that the same phenomenon which can be observed in the previous Figure 38 is also apparent in Figure 42, but in a more pronounced form. Hence, the packet prioritization of the APL switch works correctly.

Summary (Figure 38 to Figure 43): The measurements show that stable packet-throughput of high-priority traffic, by gradual increase of ARP traffic up to ~6.400 packets/s is always ensured.

Exceeding the '*queueLength*' limit of the switch, regarding packet count, leads to discarding of excessive packets arriving at the packet queue inside the switch (see chapter 5.1.2). However, the PCS ('*Packet Count per Second*') of UDP and TCP traffic shows that the APL switch works according to packet prioritization, managing to uphold high-priority UDP traffic without traffic loss and trying to forward as much TCP traffic as possible which is next in priority, while sacrificing low priority ARP traffic instead.

In conclusion, the APL switch of Manufacturer A fulfills its packet-throughput requirements according to Table 12 regarding the desired packet processing behavior, regardless of working outside the constraints of its respective hardware limitations.

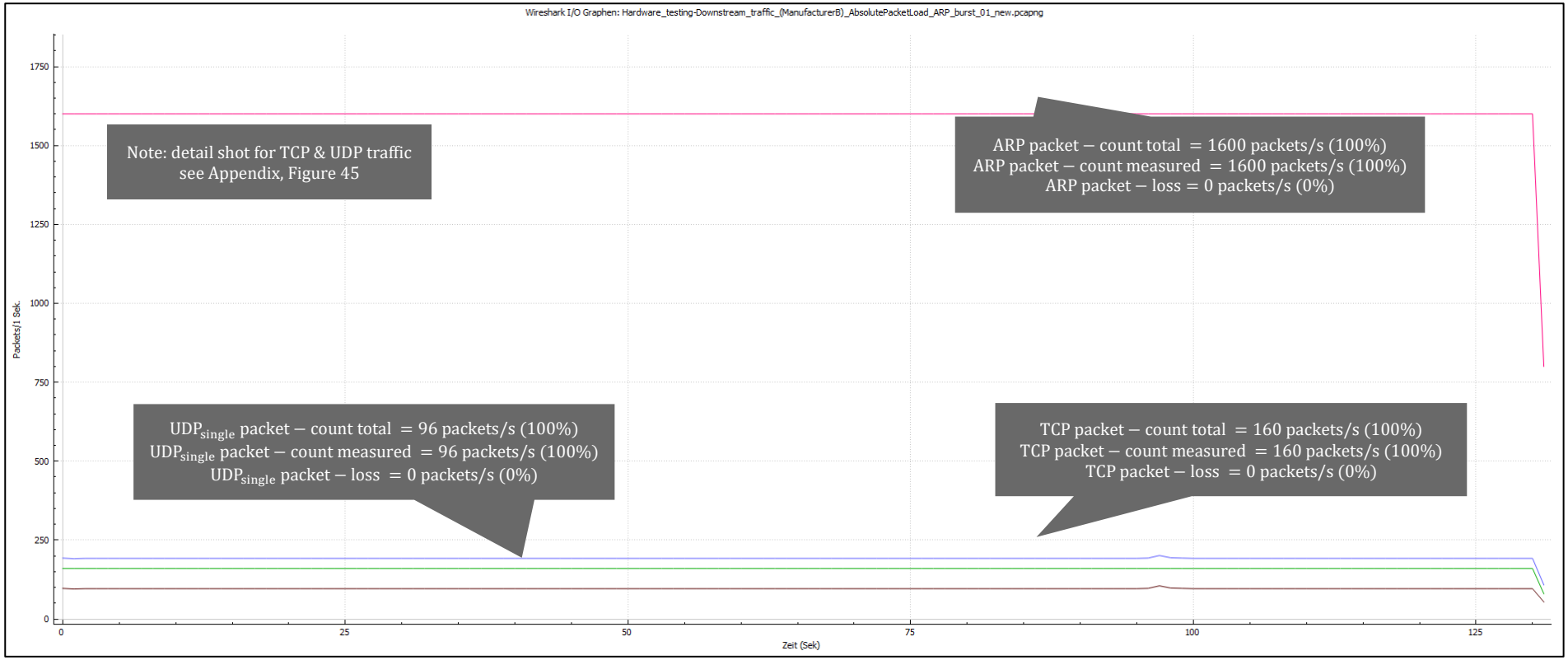

Figure 44: Downstream traffic analysis @ **increasing ARP packet count (Alternative 1), Manufacturer B** - Measurement results (200 ARP packets / 125 ms) – total packet count

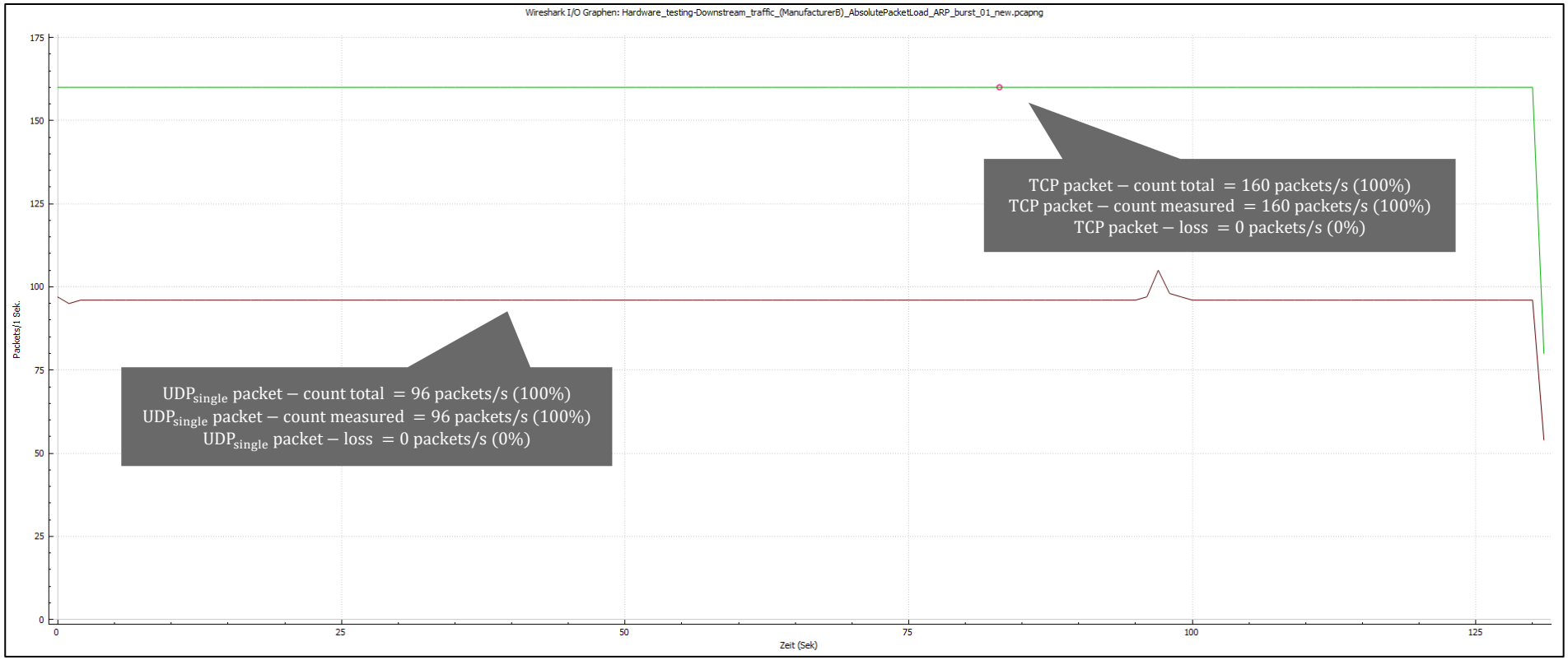

Figure 45: Downstream traffic analysis @ **increasing ARP packet count (Alternative 1), Manufacturer B** - Measurement results (200 ARP packets / 125 ms) – single source packet count

Figure 44 shows that no traffic type experiences packet loss at a APL traffic PCS ('*Packet Count per Second*') of $1600\ \text{packets/s}$. Hence, all packets are forwarded successfully by the APL switch.

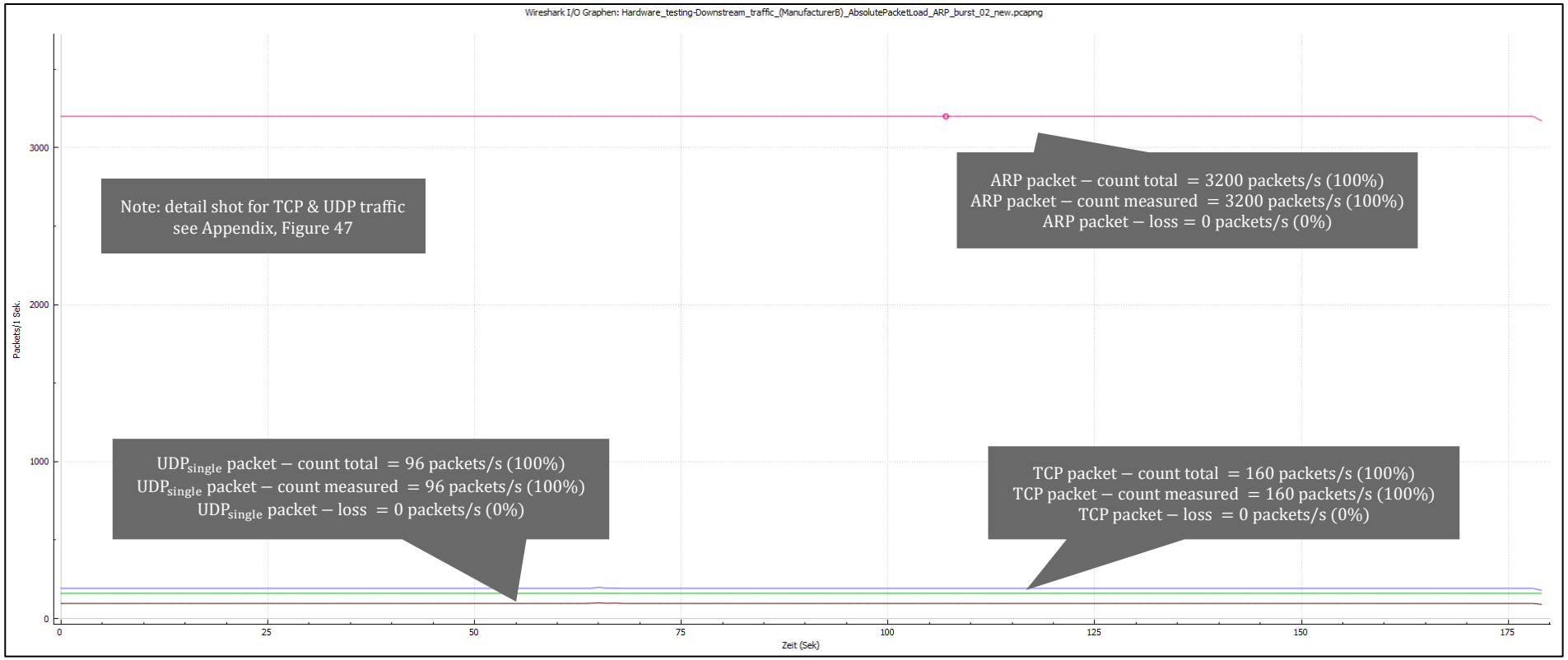

Figure 46: Downstream traffic analysis @ **increasing ARP packet count (Alternative 1), Manufacturer B** - Measurement results (400 ARP packets / 125 ms) – total packet count

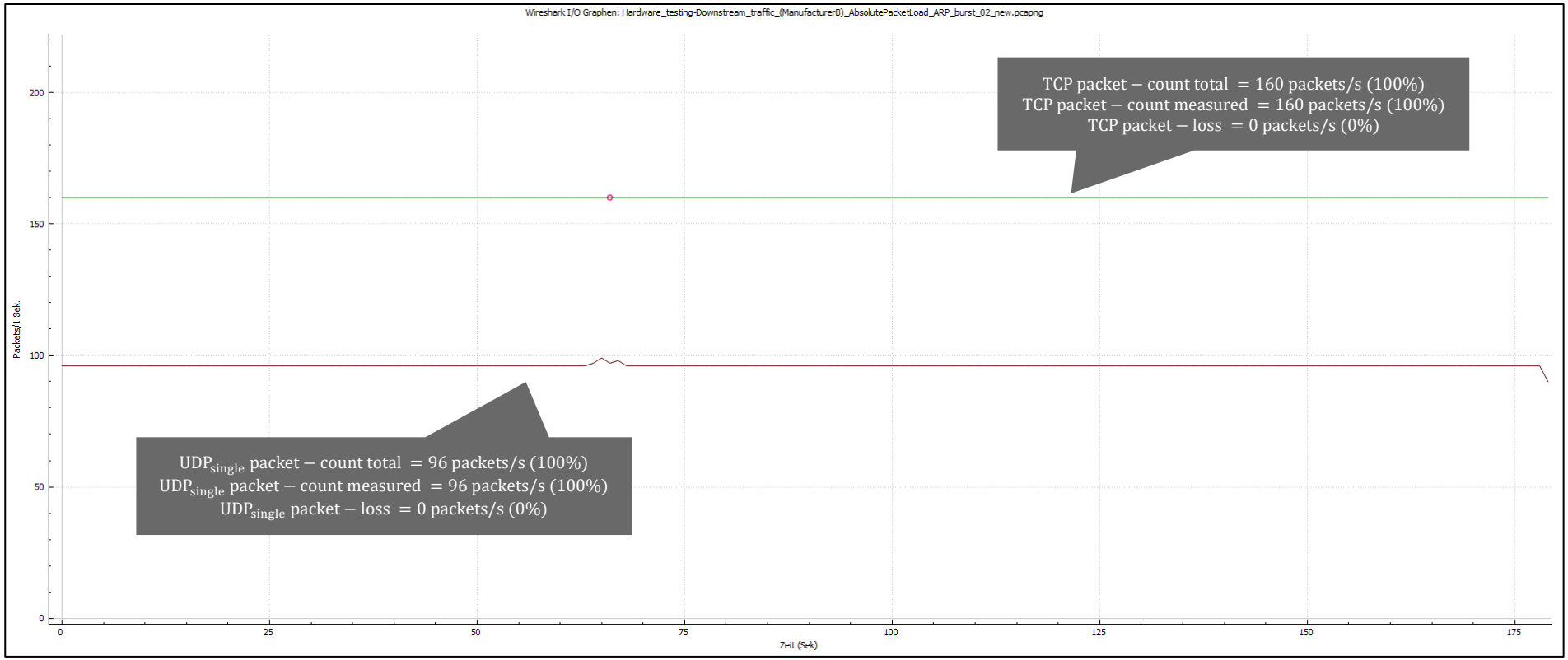

Figure 47: Downstream traffic analysis @ **increasing ARP packet count (Alternative 1), Manufacturer B** - Measurement results (400 ARP packets / 125 ms) – single source packet count

Figure 46 shows that by doubling the PCS (‘*Packet Count per Second*’) of ARP traffic from $\sim 1.600\ \mathrm{p/s}$ to $\sim 3.200\ \mathrm{p/s}$, packet loss still does not occur. Hence, all packets are forwarded successfully by the APL switch.

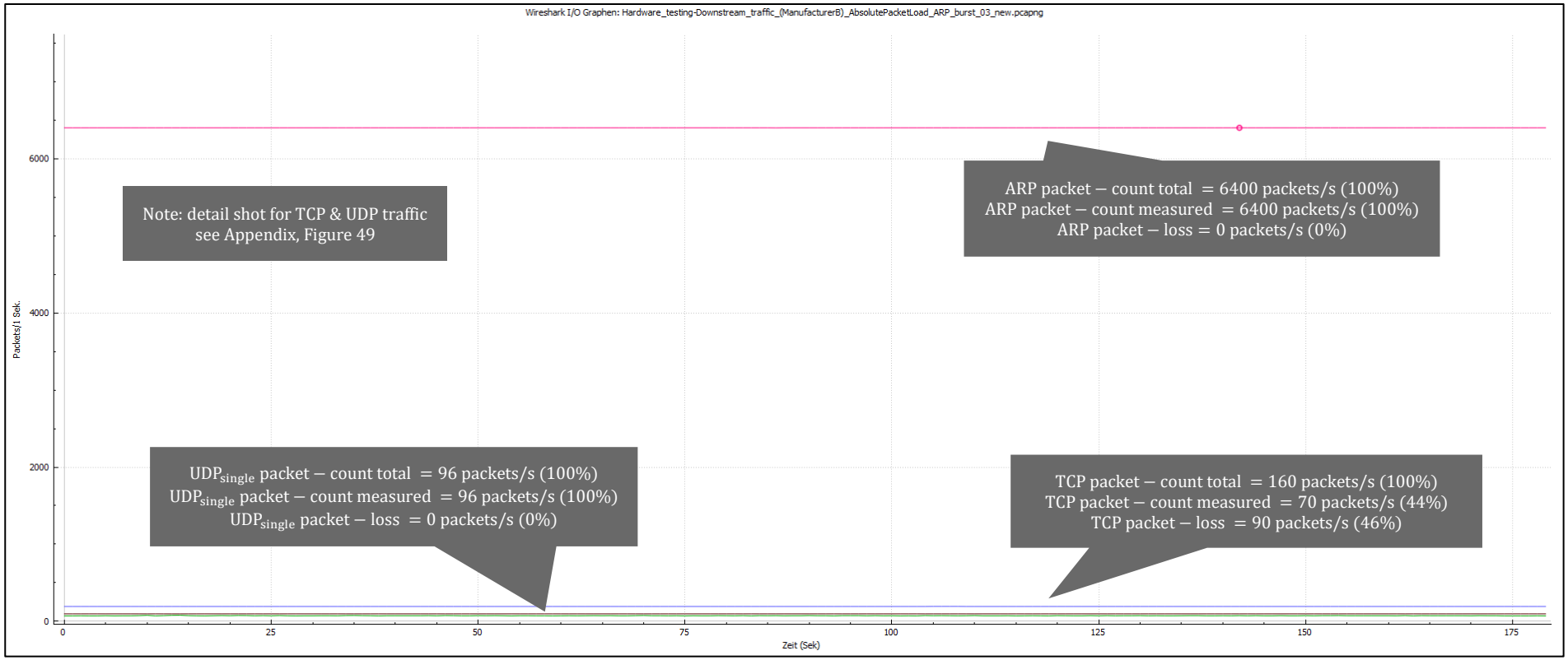

Figure 48: Downstream traffic analysis @ **increasing ARP packet count (Alternative 1), Manufacturer B** - Measurement results (800 ARP packets / 125 ms) – total packet count

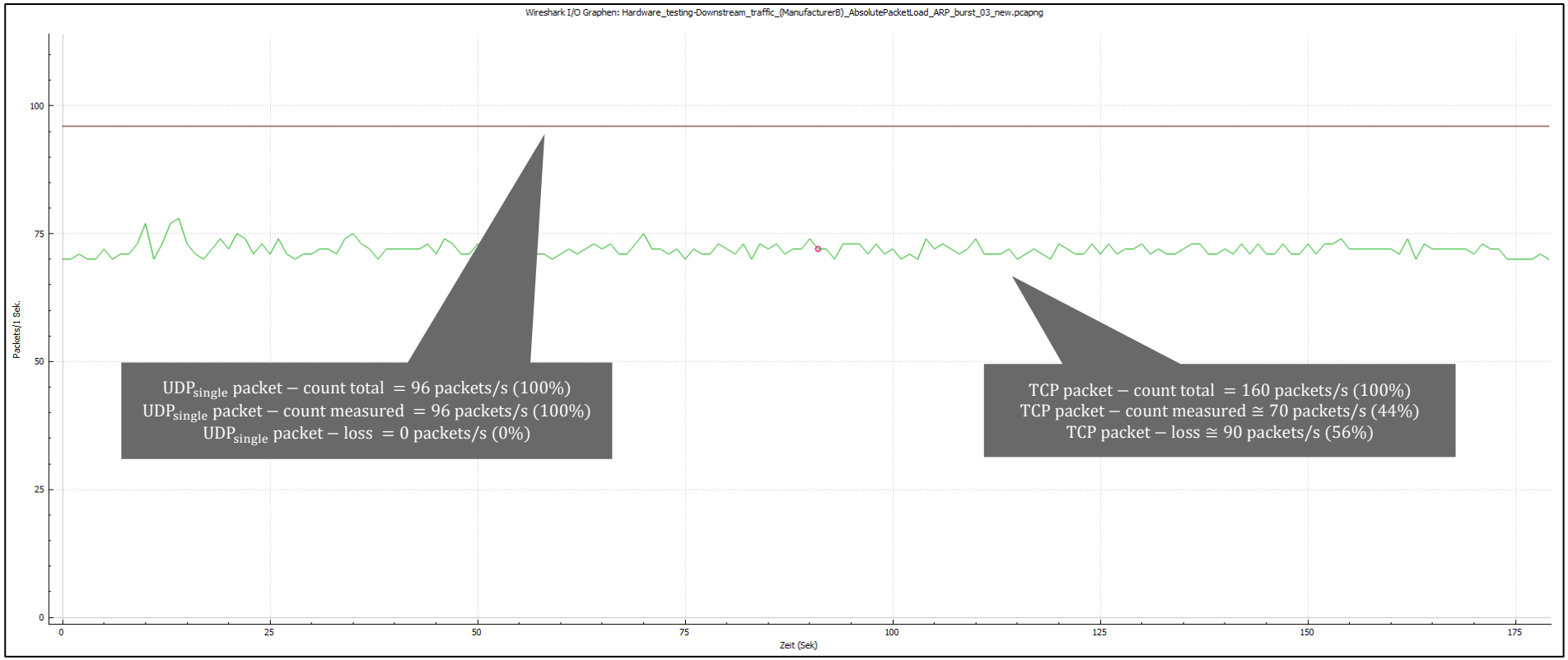

Figure 49: Downstream traffic analysis @ **increasing ARP packet count (Alternative 1), Manufacturer B** - Measurement results (800 ARP packets / 125 ms) – single source packet count

Figure 48 shows that by quadrupling the PCS ('*Packet Count per Second*') of ARP traffic from ~ 1.600 p/s to ~ 6.400 p/s, ARP and UDP traffic remain unaffected and show no packet loss. However, TCP traffic now shows packet loss going from 0% to 46%.

The total PCC ('*Packet Count per Cycle*') of UDP, TCP and APL traffic combined stating 904 packets/cycle, according to Table 10 and Table 12, stays below the total '*bufferCount*' limit of the APL switch. However, taking a look at the specific '*bufferCount*' limit based on the packet count reserved for each respective packet casting-type it becomes abundant, that ARP traffic extends the reserved packet count of 512 packets/cycle for multi- and broadcasting at a PCC of 800 packets/cycle according to Table 12 (see chapter 5.2.1).

Interestingly the result of exceeding the '*bufferCount*' limit via increased ARP traffic does not cause packet loss of ARP packets but higher prioritized TCP traffic instead. This is acting against the packet processing behavior according to packet prioritization and leads to the conclusion, that broad- and multicasting should not be performed above the respective '*bufferCount*' limit of said APL switch.

Summary (Figure 44 to Figure 49): The measurements show that stable packet-throughput of high-priority traffic, by gradual increase of ARP traffic up to $\sim 6.400\ packets/s$ is always ensured, while working inside the respective hardware limitations of the APL switch.

Exceeding the '*bufferCount*' limit of the switch, regarding packet count, leads to discarding of excessive packets arriving at the packet buffer inside the switch (see chapter 5.2.1). This leads to faulty behavior regarding packet processing of the APL switch and can lead to packet loss of mid-priority TCP traffic if not limited accordingly.

In conclusion, the APL switch of Manufacturer B fulfills its packet-throughput requirements according to Table 12 regarding the desired packet processing behavior, while working in the constraints of its respective hardware limitations.

A.3.2.3 Packet processing @ decreasing TCP cycle time

The following tests have been conducted by increasing TCP traffic through decreasing its cycle time. The traffic parameters used in these tests are stated in Table 13.

Table 13: Downstream traffic analysis (Alternative 1), Manufacturers A&B – Traffic parameters, increasing TCP traffic @ varying cycle time

	TCP IP field device update		
user priority	4		
total packet payload **(TPP)**	1.500 Byte (data) + 42 Byte (framing/transmission) = 1.542 Byte		
packet cycle time **(PCT)**	**125 ms / 62,5 ms / 31,25 ms**		
Packet Count per Cycle **(PCC)**	80 packets/cycle		
Packet data Payload per Cycle **(PPC)**	80 packets / cycle · 1.500 Byte = 120.000 Byte/*cycle* (~117,2 kByte/cycle)		
total frame payload per cycle **(FPC)**	80 packets / cycle · 1.542 Byte = 123.360 Byte/*cycle* (~120,5 kByte/cycle)		
Packet Count per Second **(PCS)**	**640 packets/s**	**1.280 packets/s**	**2.560 packets/s**
packet data payload per second **(PPS)**	640 packets / s · 1.500 Byte = 960.000 Byte/s (~**937,5 kByte/s**)	1.280 packets / s · 1.500 Byte = 1.920.000 Byte/s (~**1,83 MByte/s**)	2.560 packets / s · 1.500 Byte = 3.840.000 Byte/s (~**3,66 MByte/s**)
total frame payload per second **(FPS)**	640 packets / s · 1.542 Byte = 986.880 Byte/s (~**963,8 kByte/s**)	1.280 packets / s · 1.542 Byte = 1.973.760 Byte/s (~**1,88 MByte/s**)	2.560 packets / s · 1.542 Byte = 3.974.520 Byte./s (~**3,79 MByte/s**)

The following figures show the packet-throughput behavior of the switch when using the stated in Table 13. The parameters for all other traffic types (UDP, ARP) remained the same as stated in Table 10.

Figure 50 to Figure 55 show the packet-throughput behavior of Manufacturer A in downstream direction. Figure 56 to Figure 61 show the packet-throughput behavior of Manufacturer B in downstream direction.

The magenta line resembles ARP traffic at ~800 p/s. The green line represents TCP traffic ~640 … ~2.560 packets/s. The blue line shows the entire captured UDP traffic send by the workstation out to all field device emulators at ~192 … 256 packets/s. The brown line shows the UDP traffic from one field device emulator at 16 packets/s for Manufacturer A and 96 packets/s for Manufacturer B.

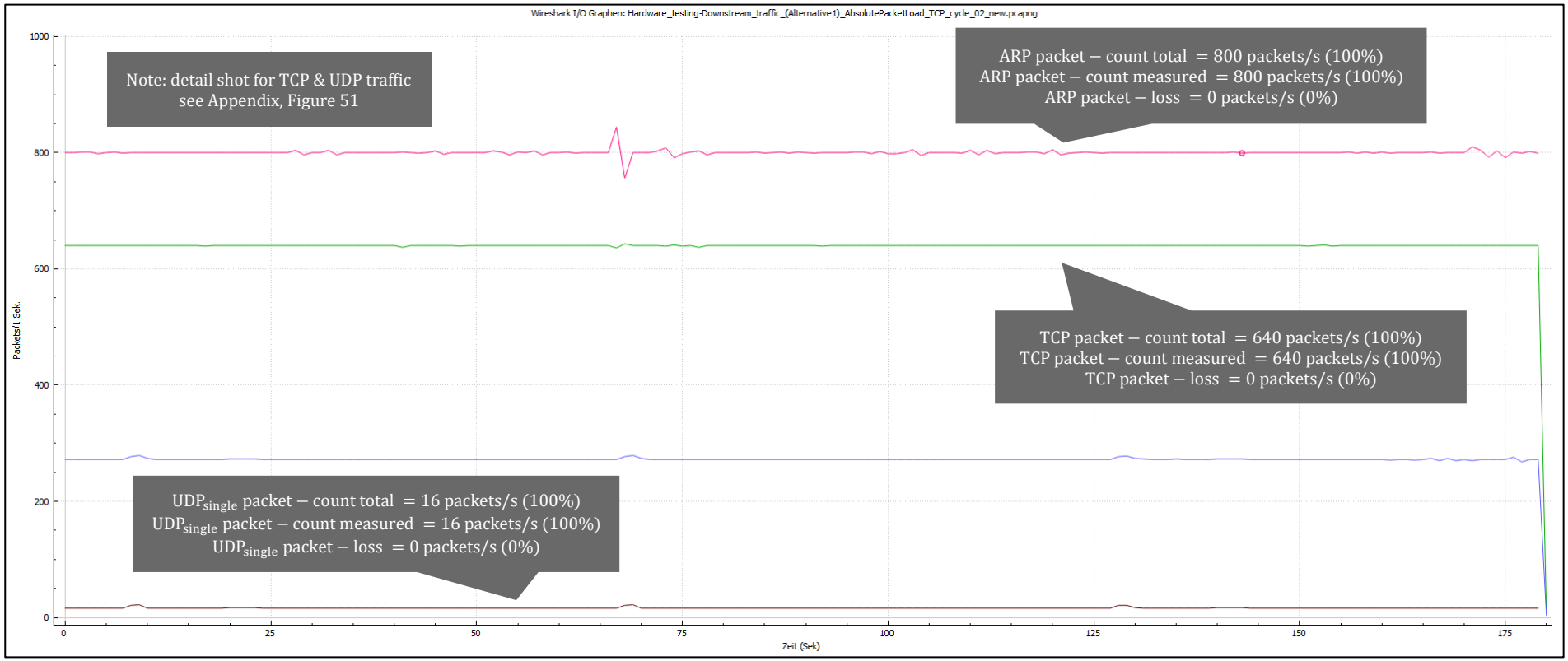

Figure 50: Downstream traffic analysis @ **decreasing TCP cycle time (Alternative 1), Manufacturer A** - Measurement results (80 TCP packets / 125 ms) – total packet count

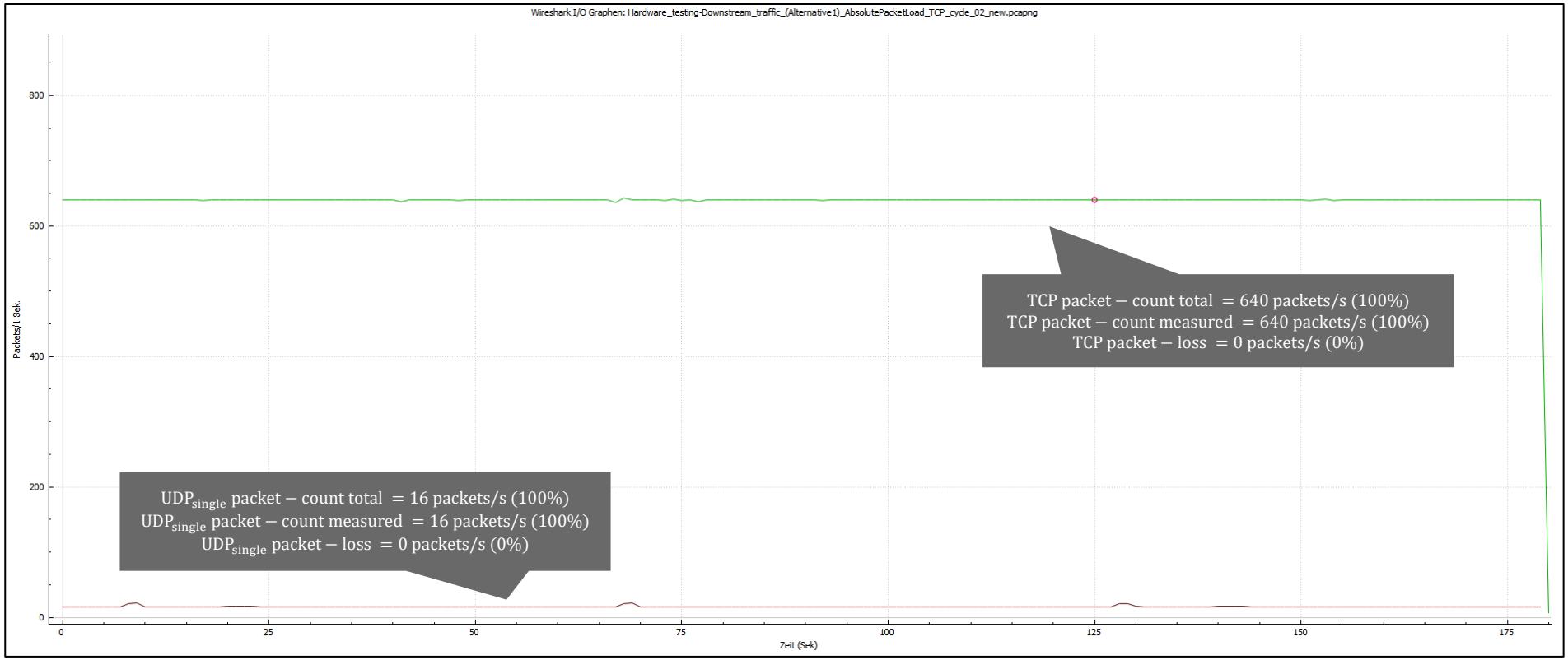

Figure 51: Downstream traffic analysis @ **decreasing TCP cycle time (Alternative 1), Manufacturer A** - Measurement results (80 TCP packets / 125 ms) – single source packet count

Figure 50 shows that no traffic type experiences packet loss at a TCP traffic PCS ('*Packet Count per Second*') of $640\ \text{packets/s}$. By dividing the TCP cycle time with the TCP PPT, according to formula (25) of the measurement summary in chapter A.3.1, the following total TCP packet count derives:

$$\rightarrow\ x_{\text{packets,TCP,1...8}} = \frac{T_{\text{TCP}} - 8 \cdot PPT_{\text{UDP}} - 4 \cdot PPT_{\text{ARP}}}{PPT_{\text{TCP}}} \tag{39}$$

$$\rightarrow x_{\text{packets,TCP,1...8}} = \frac{(500\ ms - 8 \cdot 2\ ms - 4 \cdot 8\ ms)/\ cycle}{125\ \mu s + \frac{1542\ Byte/packet}{10\ Mbit/s} + 4\ \mu s + 1{,}33\ \mu s} = \frac{\mathbf{452\ ms\ (1...8.cycle)}}{125\ \mu s + 1{,}23 ms + 4\ \mu s + 1{,}33\ \mu s} \frac{packets}{cycle} \cong 332 \frac{packets}{cycle} \tag{40}$$

Note: While using an TCP cycle time of $500\ \mathrm{ms}$, UDP with a cycle time of $62{,}5\ \mathrm{ms}$ arrives a total of eight times and ARP traffic with a cycle time of $125\ \mathrm{ms}$ arrives a total of 4 times simultaneously at the ingress port of the switch, where they are processed according to packet prioritization.

By dividing the TCP cycle time with the UDP cycle time, 8 different cycles subsequently repeating each other over and over again, can be observed based on the packet throughput behavior of the APL switch. Due to UDP having a higher priority than TCP and ARP traffic, the processing time of TCP and ARP packets is delayed until all UDP packets are forwarded.

- According to formula (40) the APL switch hardware could be able to process up to $332\ TCP\ packets$ each TCP cycle in terms of PPT. This is possible because the PPT of all traffic types fits into the TCP cycle time of $500\ ms$.

The calculation shows that the total packet count processable by the APL switch theoretically would grant a total TCP PPC of $332\,\frac{\mathrm{packets}}{\mathrm{cycle}}$. This results in a PPS of $\sim 664\ \mathrm{packets/s}$. This indicates that the total PCS of $\sim 640\,\frac{packets}{s}$, according to Table 13, can be processed in time, thus resulting in no packet loss. Hence, all packets are forwarded successfully by the APL switch.

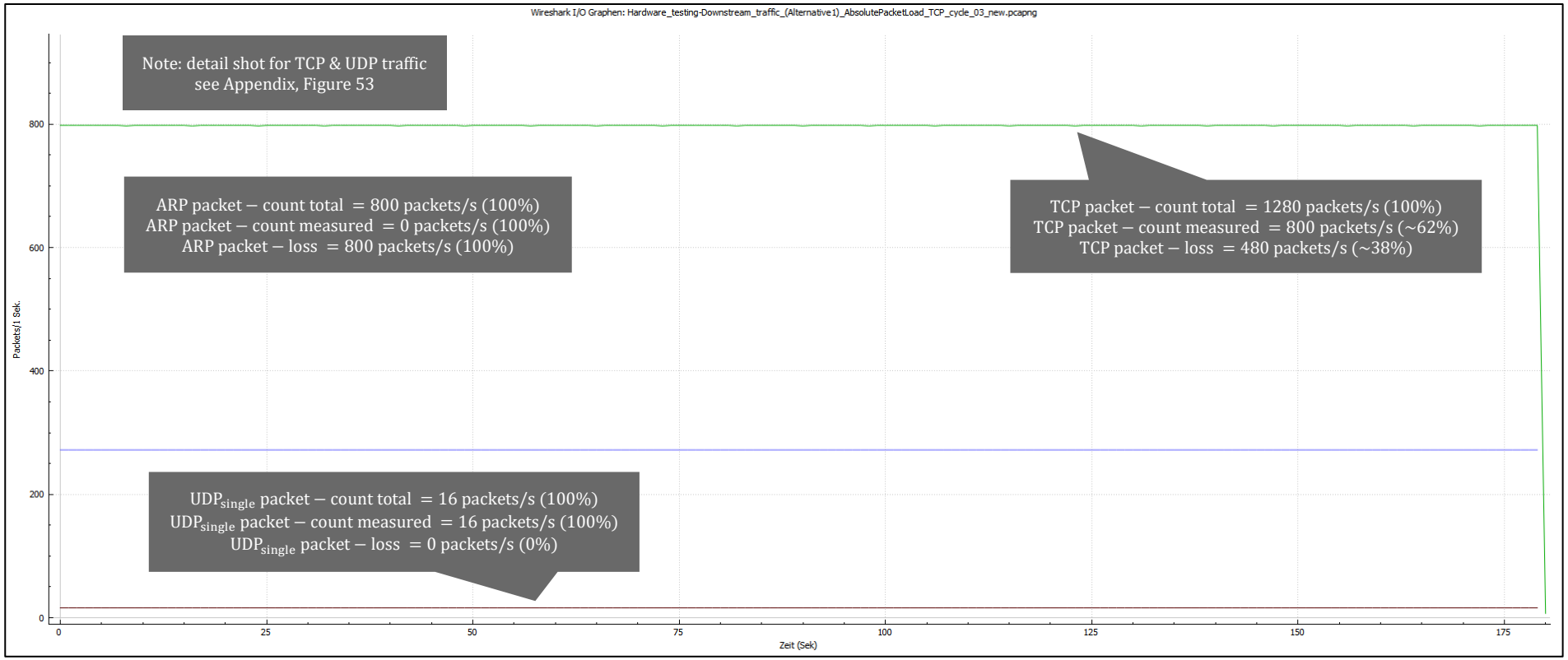

Figure 52: Downstream traffic analysis @ **decreasing TCP cycle time (Alternative 1), Manufacturer A** - Measurement results (80 TCP packets / 62,5 ms) – total packet count

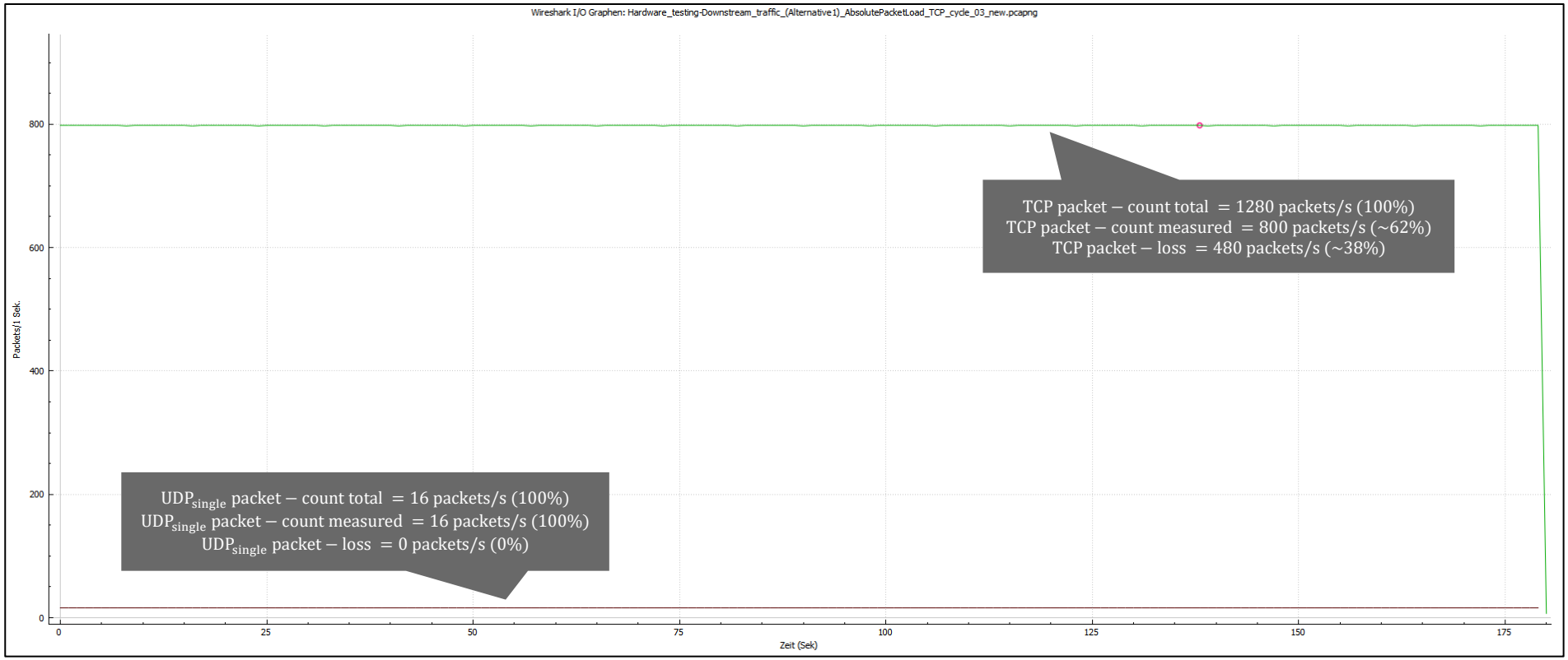

Figure 53: Downstream traffic analysis @ **decreasing TCP cycle time (Alternative 1), Manufacturer A** - Measurement results (80 TCP packets / 62,5 ms) – single source packet count

Figure 52 shows that by doubling the PCS ('*Packet Count per Second*') of TCP traffic from ~ 640 packets/s to ~ 1.280 packets/s, packet loss increases from 0% to 38%, while ARP traffic gets dropped completely. However, the UDP traffic is still unaffected and shows no packet loss. The ARP packet loss happens due to packet prioritization of the higher prioritized TCP traffic. By decreasing the TCP cycle time below the actual PPT of the internal switch hardware, according to formula (25) of the measurement summary in chapter A.3.1, the incoming TCP packets according to Table 13 cannot be processed fast enough so that TCP packets start to overlap inside the packet queue of the APL switch:

$$PPT_{\mathrm{TCP}} = 80x\left(125\ \mu s + \frac{1542\ Byte}{10\ Mbit/s} + 4\ \mu s + 1{,}33\ \mu s\right) \cong 109\ ms > T_{TCP} = 62{,}5\ ms \qquad (41)$$

By dividing the TCP cycle time with the TCP PPT, according to formula (25) of the measurement summary in chapter A.3.1, the following total TCP packet count derives:

$$\rightarrow\ x_{\text{packets,TCP,1}} = \frac{T_{\text{TCP}} - 1x\ PPT_{\text{UDP}}}{PPT_{\text{TCP}}} \tag{42}$$

$$\rightarrow x_{\text{packets,TCP,1}} = \frac{(62{,}5\ ms - 1{\cdot}2\ ms)/cycle}{125\ \mu s + \frac{1542\ Byte/packet}{10\ Mbit/s} + 4\ \mu s + 1{,}33\ \mu s} = \frac{\mathbf{60{,}5\ ms\ (1 \ldots 8.cycle)}}{125\ \mu s + 1{,}23 ms + 4\ \mu s + 1{,}33\ \mu s}\frac{packets}{cycle} \cong 44\frac{packets}{cycle} \tag{43}$$

Note: While using an TCP cycle time of $62{,}5$ ms, UDP with a cycle time of $62{,}5$ ms simultaneously arrives at the ingress port of the switch, where they are processed according to packet prioritization.

By dividing the TCP cycle time with the UDP cycle time, 1 cycle subsequently repeating itself over and over again, can be observed based on the packet throughput behavior of the APL switch. Due to UDP having a higher priority than ARP traffic, the processing time of ARP packets is delayed until all UDP packets are forwarded.

- According to formula (43) the APL switch hardware could be able to process up to $44\ TCP\ packets$ each TCP cycle in terms of PPT. This is possible because the PPT of all traffic types fits into the TCP cycle time of $500\ ms$.

One may notice that the ARP PPT used for calculation (43) is not apparent although ARP traffic with a cycle time of 125 ms arrives every second cycle. Due to packet prioritization ARP packets are treated as inferior to UDP and TCP packets due to packet priority and get queued indefinitely because UDP and TCP packets take up the entire cycle time in terms of PPT.

The calculation shows that the total packet count processable by the APL switch theoretically would grant a TCP PPC of $44\frac{\text{packets}}{\text{cycle}}$. This results in a PPS of ~ 704 packets/s. This indicates that the total PCS of $\sim 1.280 = \frac{packets}{s}$, according to Table 13, cannot be processed in time, thus resulting in a packet loss of ~ 480 packets/s. Said packet loss happens due to exceeding the '*dataRate*' limit of the APL 10 Mbit/s spur line (~ 1.19 Mbyte/s), connecting the APL switch with its respective field devices, due to the combined packet load of UDP and TCP traffic reaching said limitation:

$$dataRate_{\text{A\&B,max}} = 10\frac{Mbit}{s} = 1.250.000\ Byte \cong 1{,}19\ MByte/s \tag{44}$$

$$\frac{packet_{\text{load}}}{s} = PPS_{\text{type}} \cdot packet_{\text{size,type}} \tag{45}$$

$$\left.\frac{packet_{\text{load}}}{s}\right|_{\text{total}} = PPS_{\text{TCP}} \cdot packet_{\text{size,TCP}} + PPS_{\text{UDP}} \cdot packet_{\text{size,UDP}} \cdot x_{\text{devices}} \qquad (46)$$

$$= 800\frac{packets}{s} \cdot 1542\ Byte + 16\frac{packets}{s} \cdot 24 \cdot 88\ Byte = 1.267.392\frac{Byte}{s} \cong 1.21\frac{MByte}{s} > 1{,}19\ MByte/s$$

However, the APL switch still works according to packet prioritization and tries to forward as many high-priority UDP and TCP packets as possible while dropping lower prioritized ARP packets instead. Hence, the packet prioritization of the APL switch works correctly.

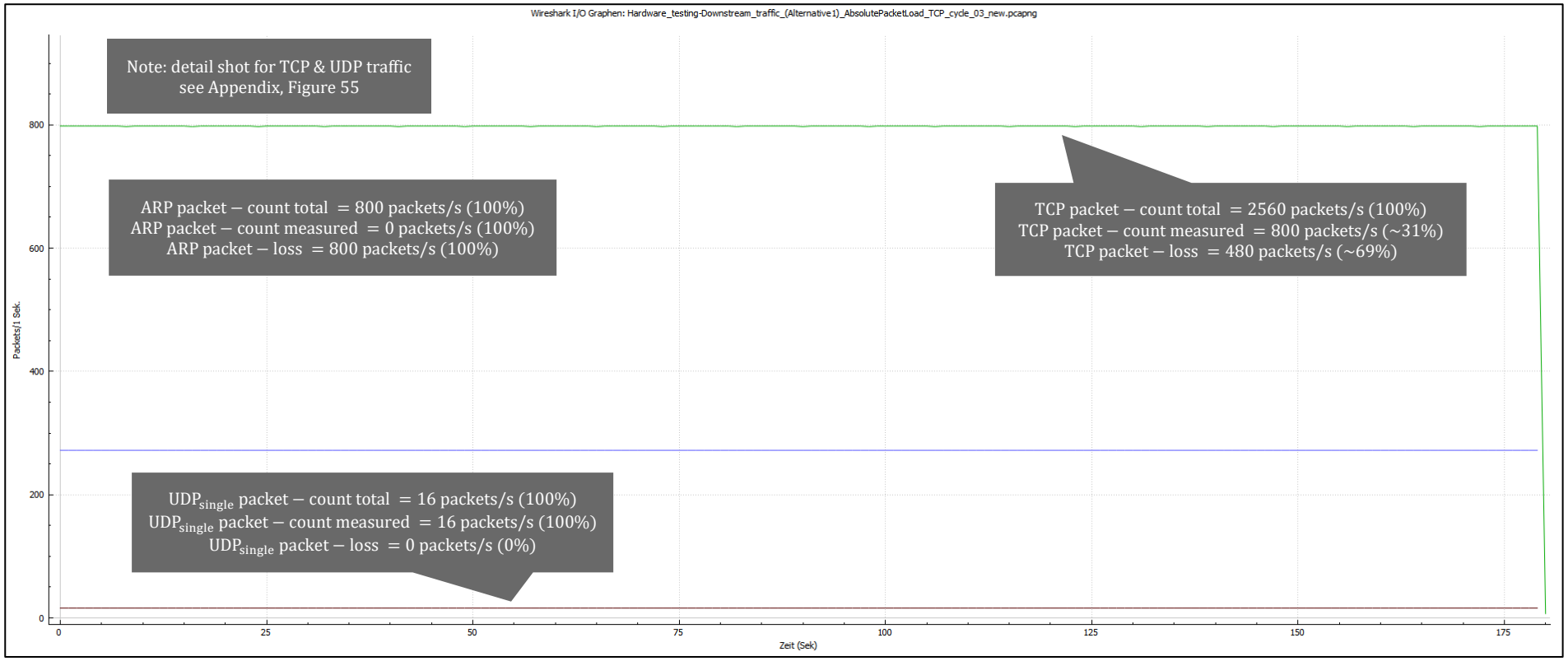

Figure 54: Downstream traffic analysis @ **decreasing TCP cycle time (Alternative 1), Manufacturer A** - Measurement results (80 TCP packets / 31,25 ms) – total packet count

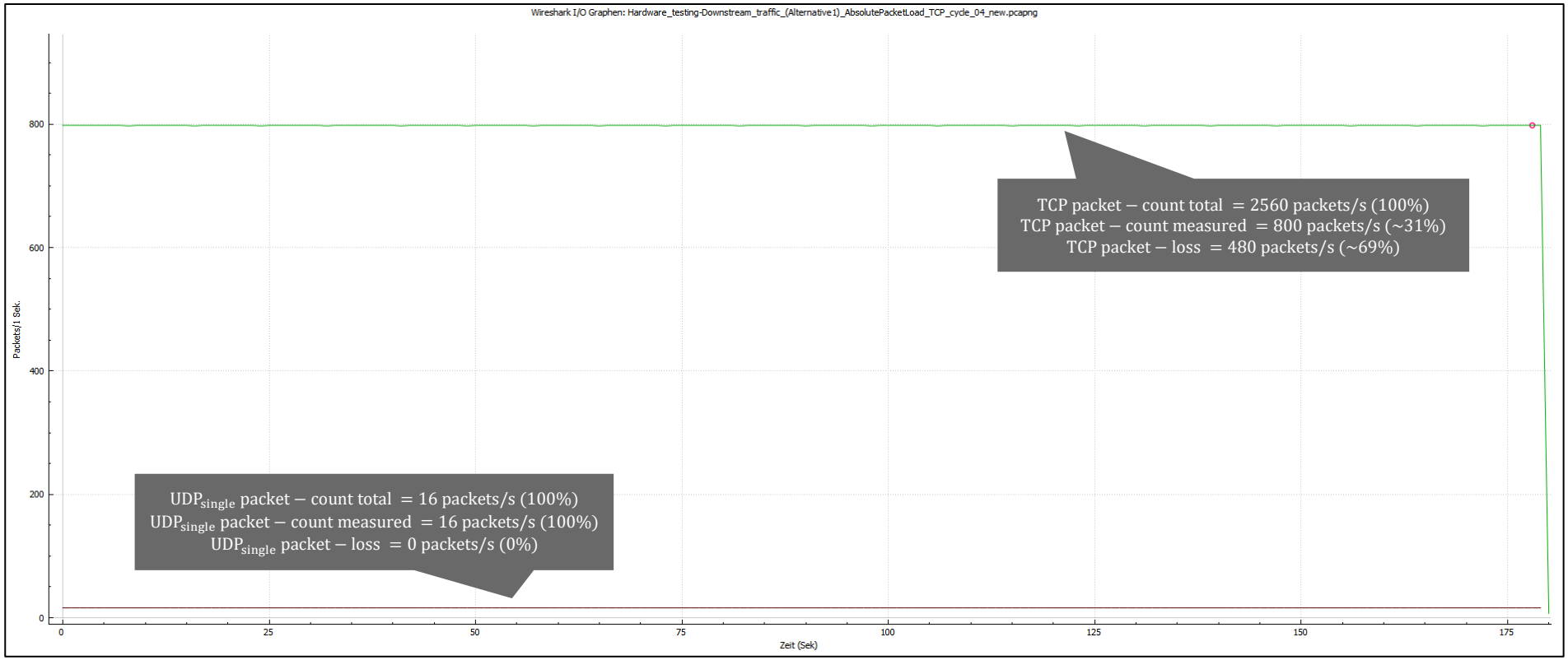

Figure 55: Downstream traffic analysis @ **decreasing TCP cycle time (Alternative 1), Manufacturer A** - Measurement results (80 TCP packets / 31,25 ms) – single source packet count

Figure 54 shows that by quadrupling the PCS ('*Packet Count per Second*') of TCP traffic from ~ 640 packets/s to ~ 2.560 packets/s, packet loss increases from 0% to 69%, while ARP traffic gets dropped completely. However, the UDP traffic is still unaffected and show no packet loss. This shows that the same phenomenon which can be observed in the previous Figure 50 is also apparent in Figure 54, but in a more pronounced form. Hence, the packet prioritization of the APL switch works correctly.

Summary (Figure 50 to Figure 55): The measurements show that stable packet-throughput of high-priority traffic, by gradual increase of TCP traffic up to ~2.560 packets/s is always ensured.

Exceeding the '*dataRate*' limit of the APL spur line, regarding packet load, leads to discarding of excessive packets forwarded by the egress port outside of the switch (see chapter 5.3.1). However, the PCS ('*Packet Count per Second*') of UDP and TCP traffic shows that the APL switch works according to packet prioritization, managing to uphold high-priority UDP traffic without traffic loss and trying to forward as much TCP traffic as possible which is next in priority, while sacrificing low priority ARP traffic instead.

In conclusion, the APL switch of Manufacturer A fulfills its packet-throughput requirements according to Table 13 regarding the desired packet processing behavior, regardless of working outside the constraints of its respective hardware limitations.

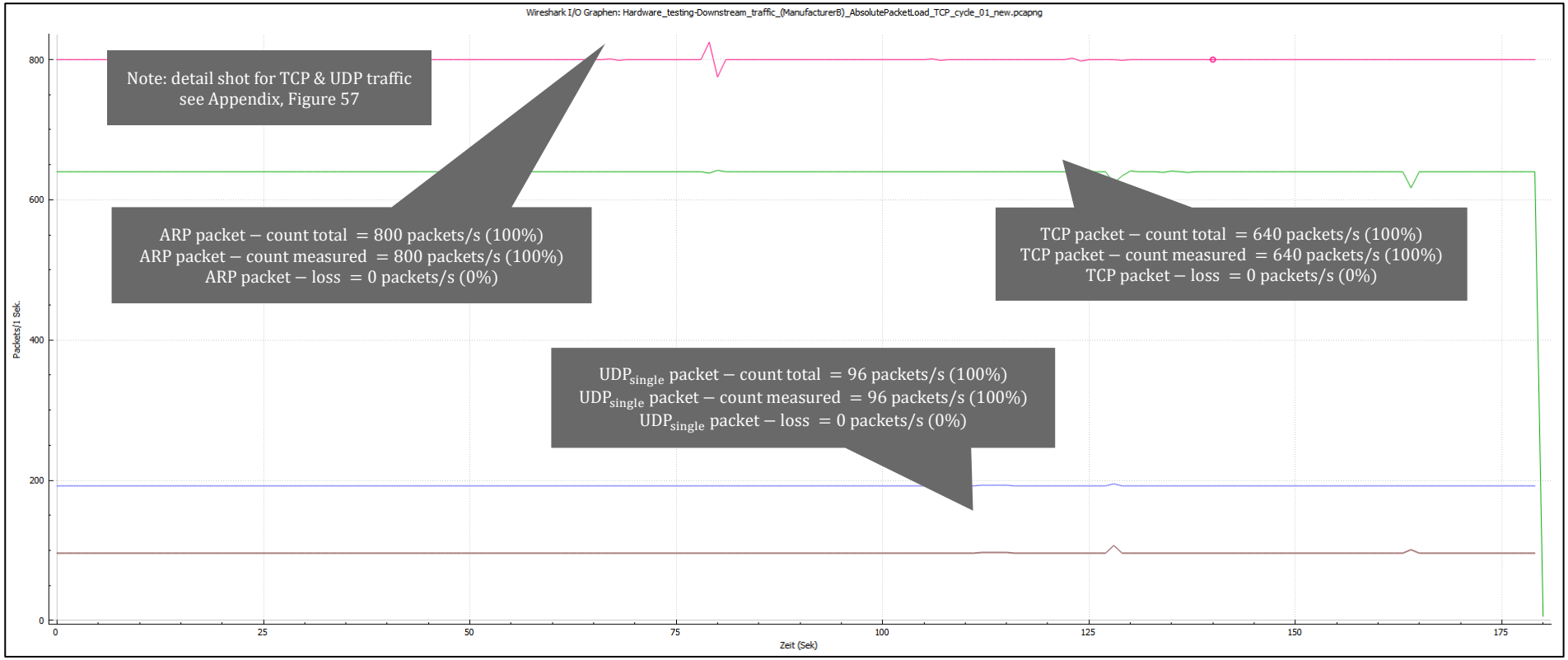

Figure 56: Downstream traffic analysis @ **decreasing TCP cycle time (Alternative 1), Manufacturer B** - Measurement results (80 TCP packets / 125 ms) – total packet count

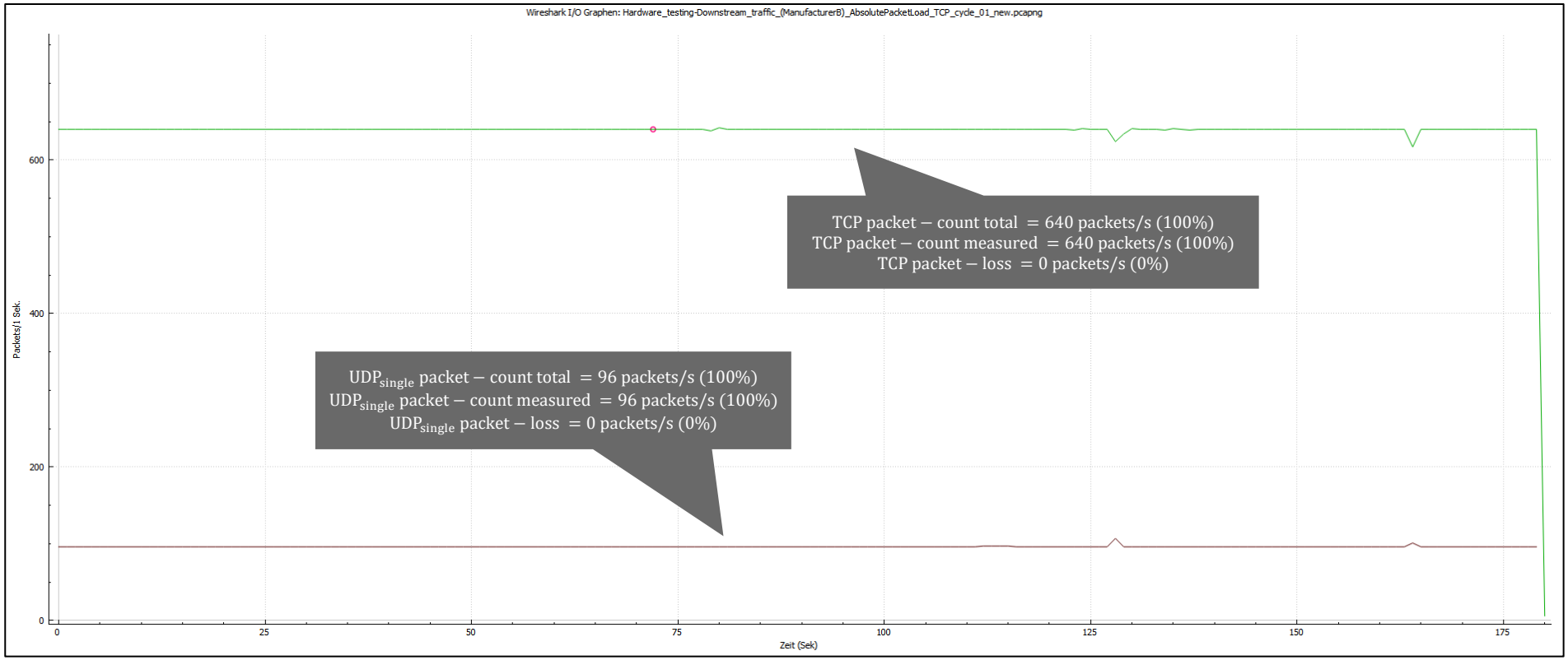

Figure 57: Downstream traffic analysis @ **decreasing TCP cycle time (Alternative 1), Manufacturer B** - Measurement results (80 TCP packets / 125 ms) – single source packet count

Figure 56 shows that no traffic type experiences packet loss at a TCP traffic PCS ('*Packet Count per Second*') of $640\ \text{packets/s}$. Hence, all packets are forwarded successfully by the APL switch.

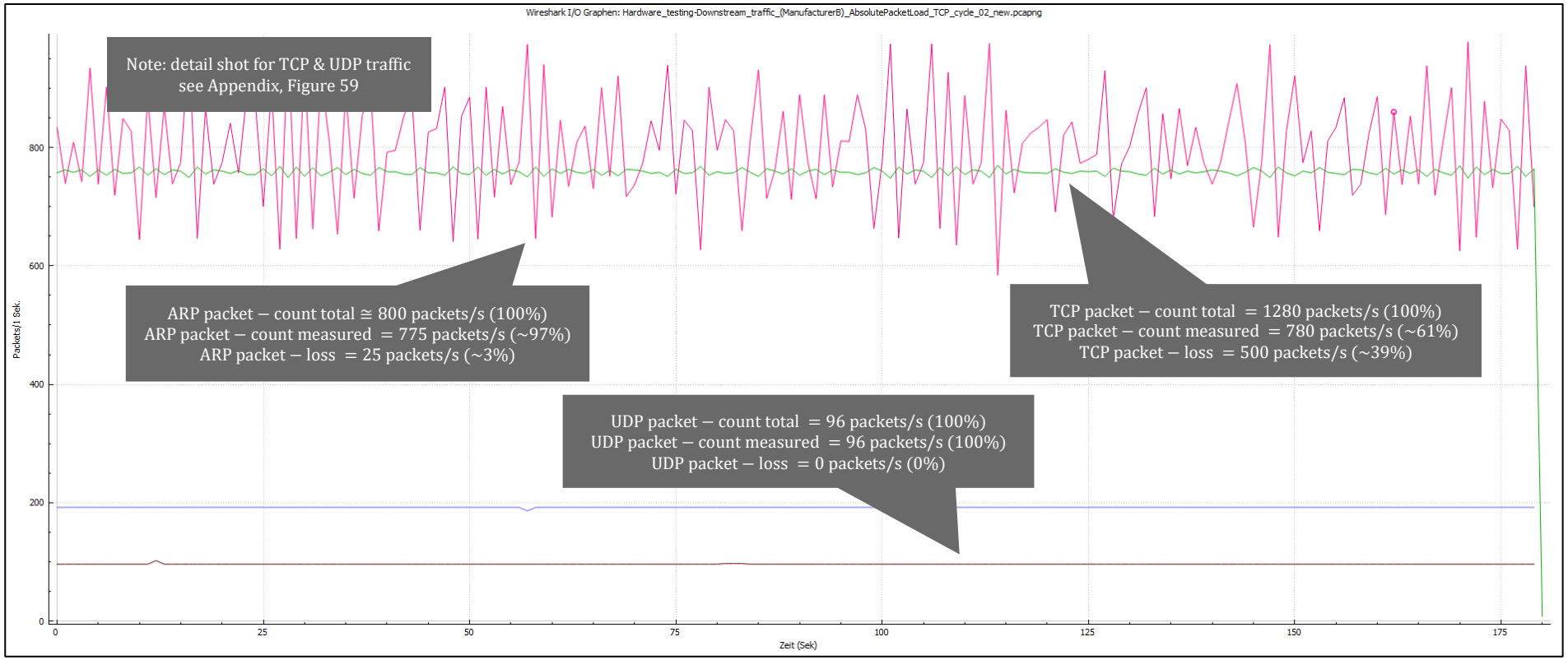

Figure 58: Downstream traffic analysis @ **decreasing TCP cycle time (Alternative 1), Manufacturer B** - Measurement results (80 TCP packets / 62,5 ms) – total packet count

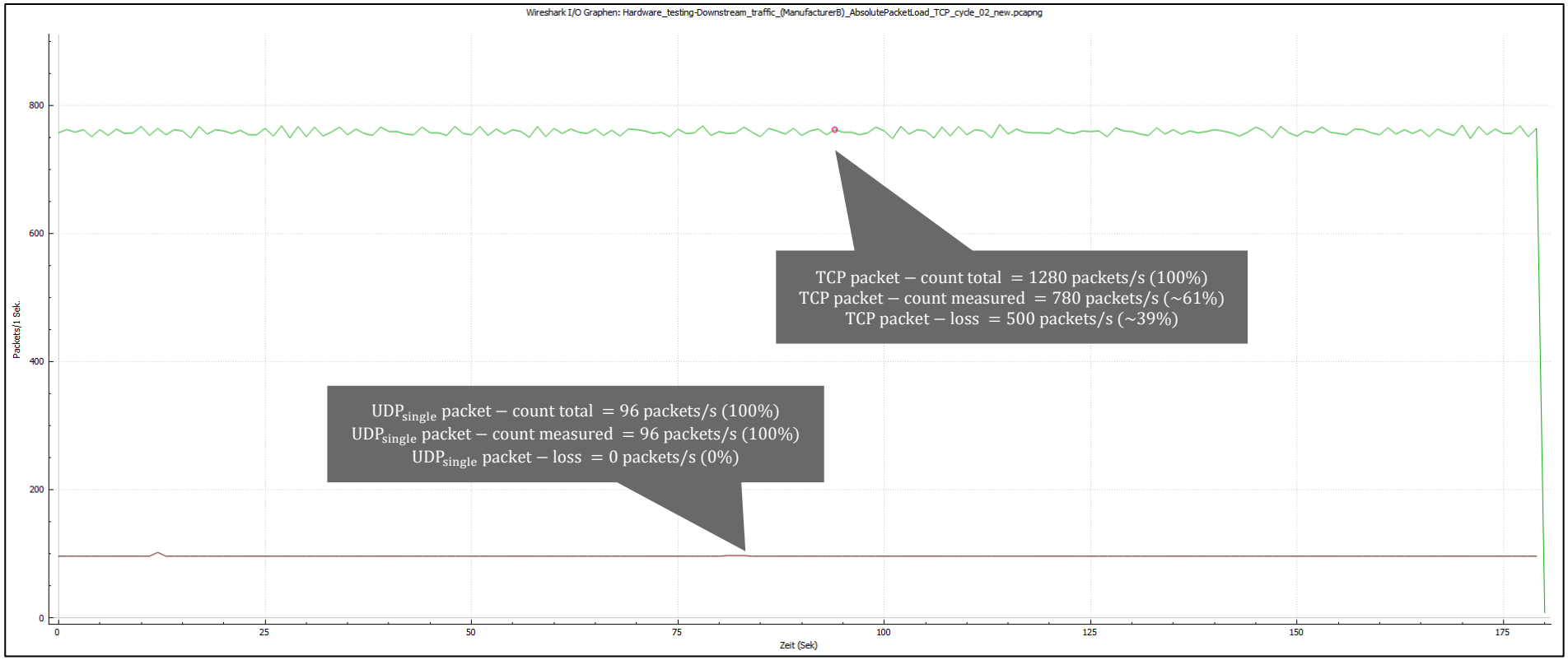

Figure 59: Downstream traffic analysis @ **decreasing TCP cycle time (Alternative 1), Manufacturer B** - Measurement results (80 TCP packets / 62,5 ms) – single source packet count

Figure 58 shows that by doubling the PCS ('*Packet Count per Second*') of TCP traffic from ~ 640 packets/s to ~ 1.280 packets/s, packet loss increases from 0% to 39%, while ARP traffic also drops from 0% to 3%. However, the UDP traffic is still unaffected and show no packet loss. The same phenomenon of packet-throughput limitation by reaching the '*dataRate*' limit of the APL line, which was described in the previous Figure 53 (chapter A.3.2.3), occurs again.

But although packet processing while working at the '*dataRate*' limit should lead to the total packet drop of APL traffic in favor of higher-priority UDP and TCP traffic, the APL switch of Manufacturer B partially forwards TCP as well as ARP traffic similar to its packet processing behavior in Figure 48 (chapter A.3.2.2). This time however said behavior was not triggered by exceeding the respective '*bufferCount*' limit of a specific casting-type but rather by exceeding the total '*bufferCount*' packet limit of 1024 packets/cycle.

By decreasing the TCP cycle time below the PPTs of the internal switch hardware, packets of the previous and current TCP cycle start to overlap inside the packet buffer. In a similar manner to the packet overlapping inside the packet queues of Manufacturer A mentioned in Figure 26 (chapter A.3.2.1), this leads to an overflow of the packet buffer resulting in packet loss.

However, the APL switch still manages to prioritize the high-priority non-connection based UDP traffic which is essential for the correct functionality of the connected field devices.

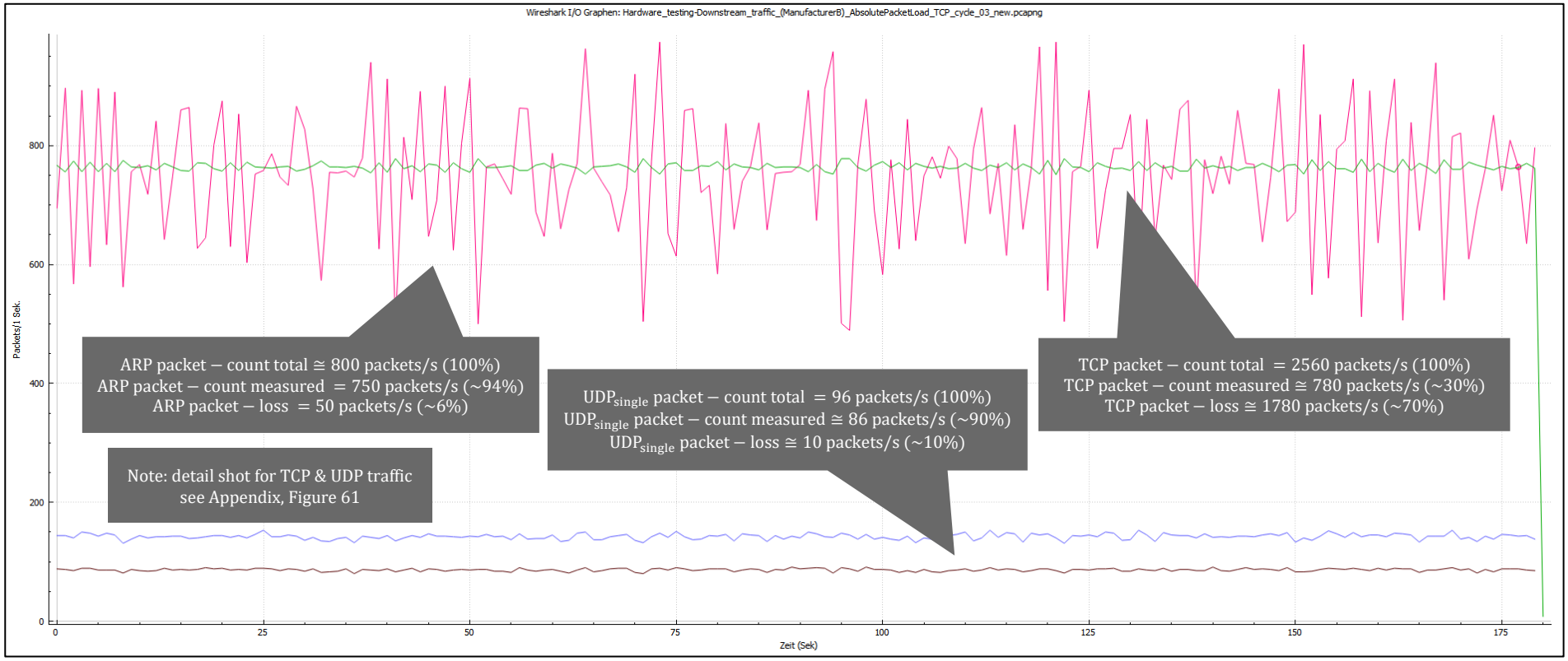

Figure 60: Downstream traffic analysis @ **decreasing TCP cycle time (Alternative 1), Manufacturer B** - Measurement results (80 TCP packets / 31,25 ms) – total packet count

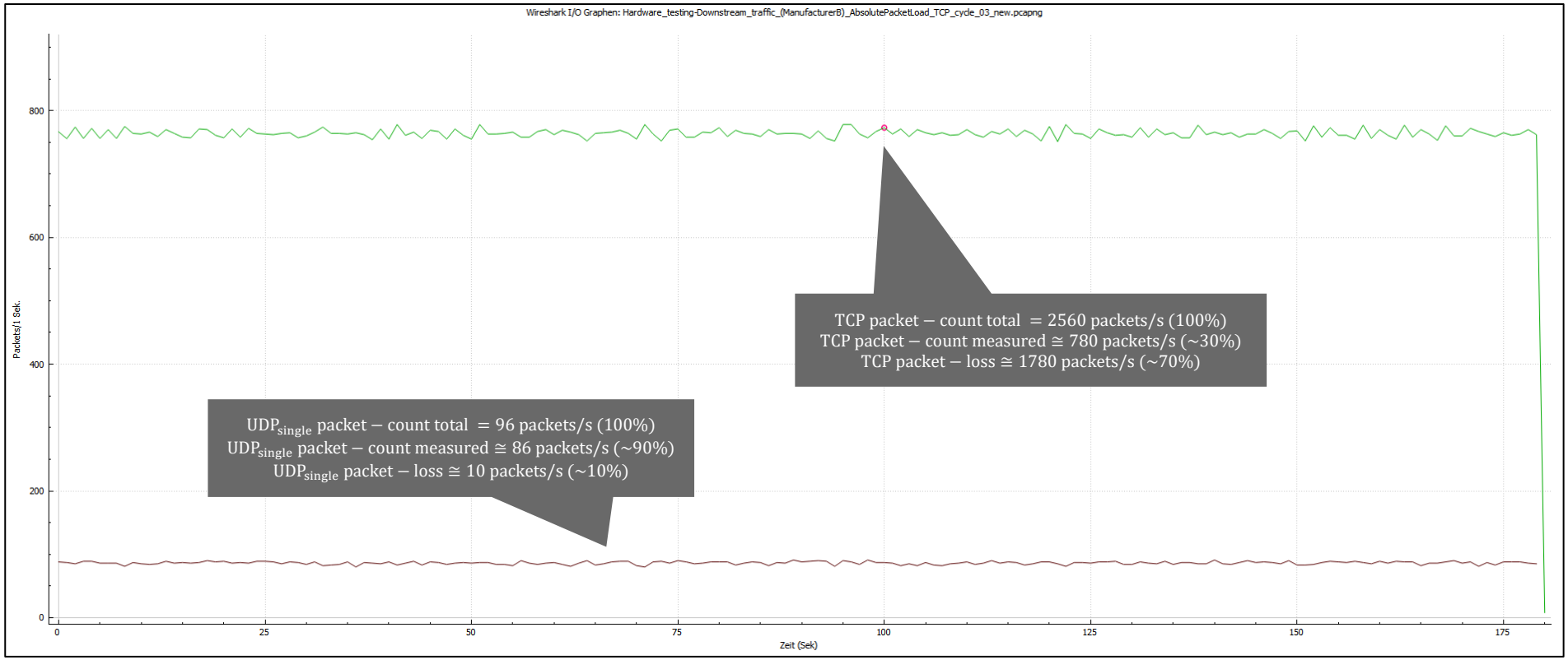

Figure 61: Downstream traffic analysis @ **decreasing TCP cycle time (Alternative 1), Manufacturer B** - Measurement results (80 TCP packets / 31,25 ms) – single source packet count

Figure 60 shows that by quadrupling the PCS ('*Packet Count per Second*') of TCP traffic from ~640 packets/s to ~2.560 packets/s, packet loss increases from 0% to 70%, while ARP traffic also drops from 0% to 6%. Besides the lower-priority traffic now the high-priority UDP traffic also shows packet loss at 10%. This shows that the same phenomenon which can be observed in the previous Figure 56 is also apparent in Figure 60, but in a more pronounced form.

This time however, the gradual increase of TCP traffic not only leads to increased packet loss of TCP traffic, but also puts the APL switch's packet processing performance in regard to forwarding high-priority UDP traffic at its limit. This is acting against the packet processing behavior according to packet prioritization and leads to the conclusion, that TCP unicasting should not be performed above the '*queueLength*' limit of said APL switch, by increasing the packet count above said limit.

Summary (Figure 56 to Figure 61): The measurements show that stable packet-throughput of high-priority traffic, by gradual increase of TCP traffic up to $\sim 2.560\ \text{packets/s}$ is always ensured, while wor king inside the respective hardware limitations of the APL switch.

Exceeding the '*bufferCount*' limit of the switch, regarding packet count, leads to discarding of excessive packets arriving at the packet buffer inside the switch (see chapter 5.2.1). Furthermore, exceeding the '*dataRate*' limit of the APL spur line, regarding packet load, leads to discarding of excessive packets forwarded by the egress port outside of the switch (see chapter 5.3.1). This leads to faulty behavior regarding packet processing of the APL switch and can lead to packet loss of high-priority UDP traffic if not limited accordingly.

In conclusion, the APL switch of Manufacturer B fulfills its packet-throughput requirements according to Table 13 regarding the desired packet processing behavior, while working in the constraints of its respective hardware limitations.

A.3.2.4 Packet processing @ increasing TCP packet count

The following tests have been conducted by increasing TCP traffic through increasing its packet count. The traffic parameters used in these tests are stated in Table 14.

Table 14: Downstream traffic analysis (Alternative 1), Manufacturers A&B – Traffic parameters, increasing TCP traffic @ varying *Packet Count per Cycle*

	TCP IP field device update		
user priority	4		
total packet payload **(TPP)**	1.500 Byte (data) + 42 Byte (framing/transmission) = 1.542 Byte		
packet cycle time **(PCT)**	500 ms		
Packet Count per Cycle **(PCC)**	**320 / 640 / 1.280 packets/cycle**		
Packet data Payload per Cycle **(PPC)**	~468,8 / ~937,5 / ~1.875 kByte/cycle		
total frame payload per cycle **(FPC)**	~481,9 / ~963,8 / ~1.972,5 kByte/cycle		
Packet Count per Second **(PCS)**	**640 packets/s**	**1.280 packets/s**	**2.560 packets/s**
packet data payload per second **(PPS)**	640 packets / s · 1.500 Byte = 960.000 Byte/s (**~937,5 kByte/s**)	1.280 packets / s · 1.500 Byte = 1.920.000 Byte/s (**~1,83 MByte/s**)	2.560 packets / s · 1.500 Byte = 3.840.000 Byte/s (**~3,66 MByte/s**)
total frame payload per second **(FPS)**	640 packets / s · 1.542 Byte = 986.880 Byte/s (**~963,8 kByte/s**)	1.280 packets / s · 1.542 Byte = 1.973.760 Byte/s (**~1,88 MByte/s**)	2.560 packets / s · 1.542 Byte = 3.974.520 Byte./s (**~3,79 MByte/s**)

The following figures show the packet-throughput behavior of the switch when using the stated in Table 14. The parameters for all other traffic types (UDP, ARP) remained the same as stated in Table 10.

Figure 62 to Figure 67 show the packet-throughput behavior of Manufacturer A in downstream direction. Figure 68 to Figure 73 show the packet-throughput behavior of Manufacturer B in downstream direction.

The magenta line resembles ARP traffic at ~800 packets/s. The green line represents TCP traffic ~640 … ~2.560 packets/s. The blue line shows the entire captured UDP traffic send by the workstation out to all field device emulators at ~192 … 256 packets/s. The brown line shows the UDP traffic from one field device emulator at 16 packets/s for Manufacturer A and 96 packets/s for Manufacturer B.

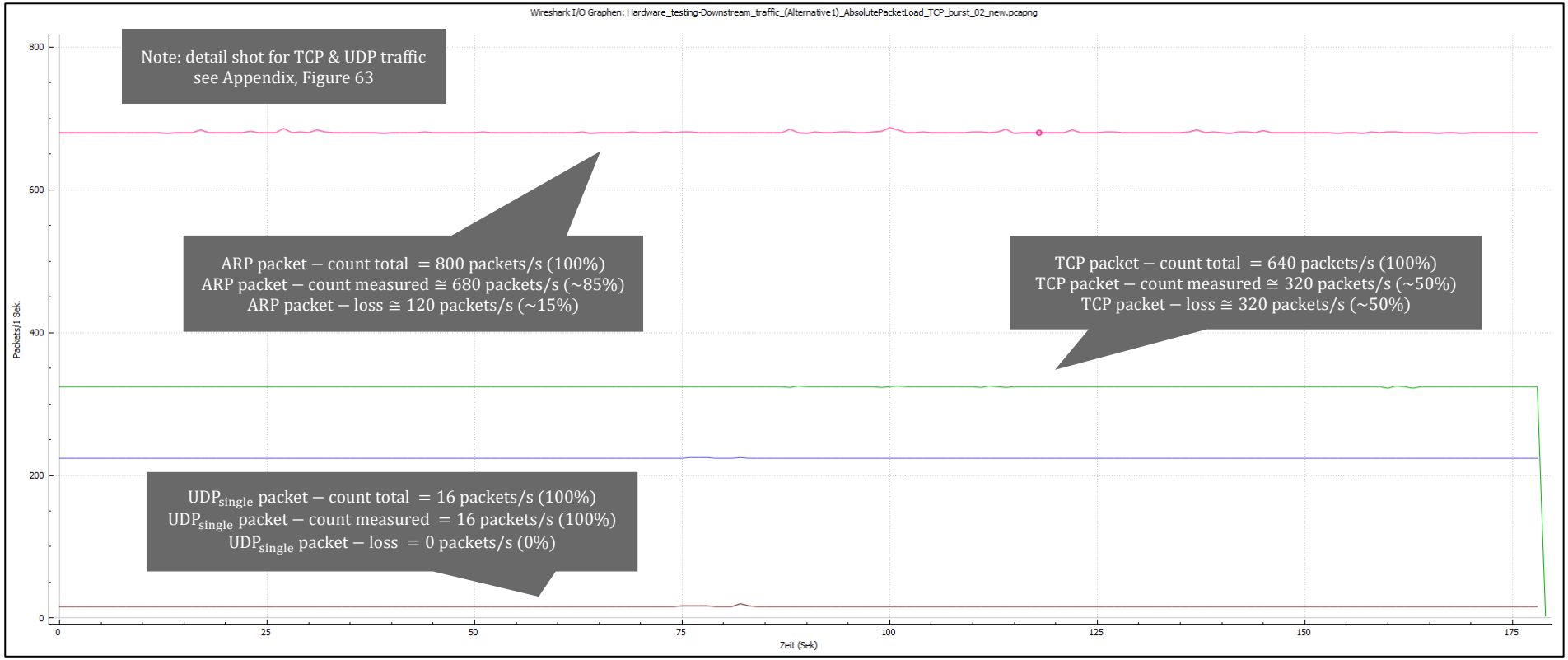

Figure 62: Downstream traffic analysis @ **increasing TCP packet count (Alternative 1), Manufacturer A** - Measurement results (320 TCP packets / 500 ms) – total packet count

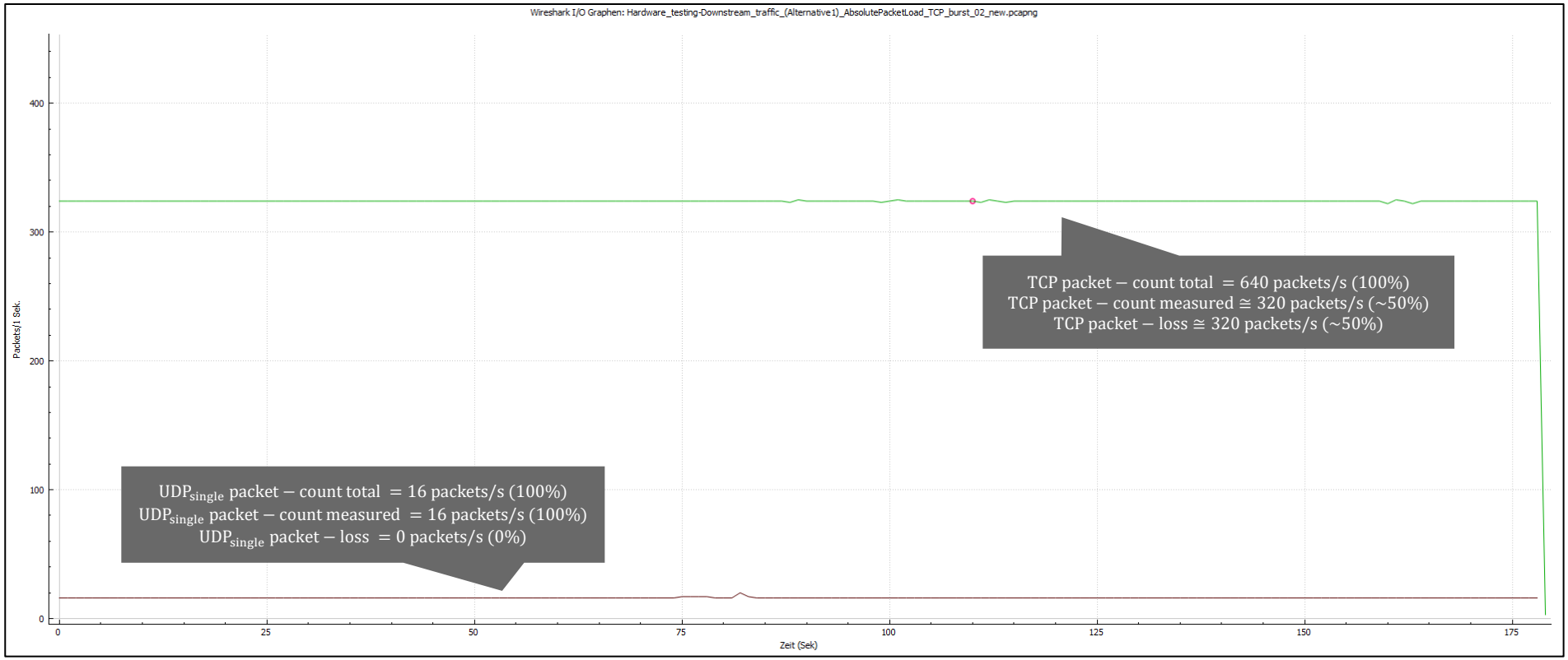

Figure 63: Downstream traffic analysis @ **increasing TCP packet count (Alternative 1), Manufacturer A** - Measurement results (320 TCP packets / 500 ms) – single source packet count

Figure 62 shows that TCP traffic has a packet loss of 50% at a total PCS ('*Packet Count per Second*') of 640 packets/s, while ARP traffic also drops by 15% at a total PCS of 800 packets/s. However, UDP traffic is unaffected and shows no packet loss at a total PCS of 16 packet/s at the recorded field device.

The ARP packet loss happens due to packet prioritization of the higher prioritized TCP traffic. By dividing the TCP cycle time with the TCP PPT, according to formula (25) of the measurement summary in chapter A.3.1, the following total TCP packet count derives:

$$\rightarrow \; x_{\text{packets,TCP,1…8}} = \frac{T_{\text{TCP}} - (8 \cdot PPT_{\text{UDP}} + 4 \cdot PPT_{\text{ARP}})}{PPT_{\text{TCP}}} \qquad (47)$$

$$\rightarrow x_{\text{packets,TCP,1…8}} = \frac{(500\ ms - 8{\cdot}2\ ms - 4{\cdot}\ 6{,}95\ ms)/\ cycle}{125\ \mu s + \frac{1542\ Byte/packet}{10\ Mbit/s} + 4\ \mu s + 1{,}33\ \mu s} = \frac{\mathbf{456{,}2\ ms\ (1…8.cycle)}}{125\ \mu s + 1{,}23 ms + 4\ \mu s + 1{,}33\ \mu s}\frac{packets}{cycle} \cong 334\frac{packets}{cycle} \qquad (48)$$

Note: During the TCP cycle time stating $500\ \text{ms}$, ARP traffic with a cycle time of $125\ \text{ms}$ simultaneously arrives a total of 4 times and UDP traffic with a cycle time of $62{,}5\ \text{ms}$ simultaneously arrives a total of 8 times at the ingress port of the switch, getting processed accordingly.

By dividing the TCP cycle time with the UDP cycle time, 8 different cycles subsequently repeating each other over and over again, can be observed based on the packet throughput behavior of the APL switch. Due to UDP having a higher priority than TCP and ARP traffic, the processing time of TCP and ARP packets is delayed until all UDP packets are forwarded.

- According to formula (48) the APL switch hardware could be able to process up to $334\ TCP\ packets$ each TCP cycle in terms of PPT. This is possible because the PPT of all traffic types fits into the TCP cycle time of $500\ ms$.

One may notice that the ARP PPT used for calculation (48) alters from the calculated value in formula (25) of the measurement summary in chapter A.3.1. This is due to the altered APR PCS ('*Packet Count per Second*') of $\sim 680\ \text{packets/s}$ shown in Figure 63, which is lowered due to packet prioritization.

The calculation shows that the total packet count processable by the APL switch theoretically would grant a TCP PPC of $334\frac{\text{packets}}{\text{cycle}}$. This results in a PPS of $640\ \text{packets/s}$. However, the actual TCP PCS only states a packet count of $\sim 320\ \text{packets/s}$ This indicates that the total PCS of $\sim 640\frac{packets}{s}$, according to Table 14, cannot be processed in time, thus resulting in a packet loss of $\sim 320\ \text{packets/s}$.

Said packet loss of TCP packets happens due to exceeding the '*queueLength*' limit of the APL switch ($128\ \text{packets/cycle}$), reaching the packet count limit of the APL switch. All incoming packets which exceed the capacity of the already full packet queue get discarded by the APL switch, similar to the measurements conducted in Figure 38 to Figure 43 (chapter A.3.2.2).

However, the high-priority UDP and TCP traffic are still being prioritized over the lower-priority ARP traffic. Hence, the packet prioritization of the APL switch works correctly.

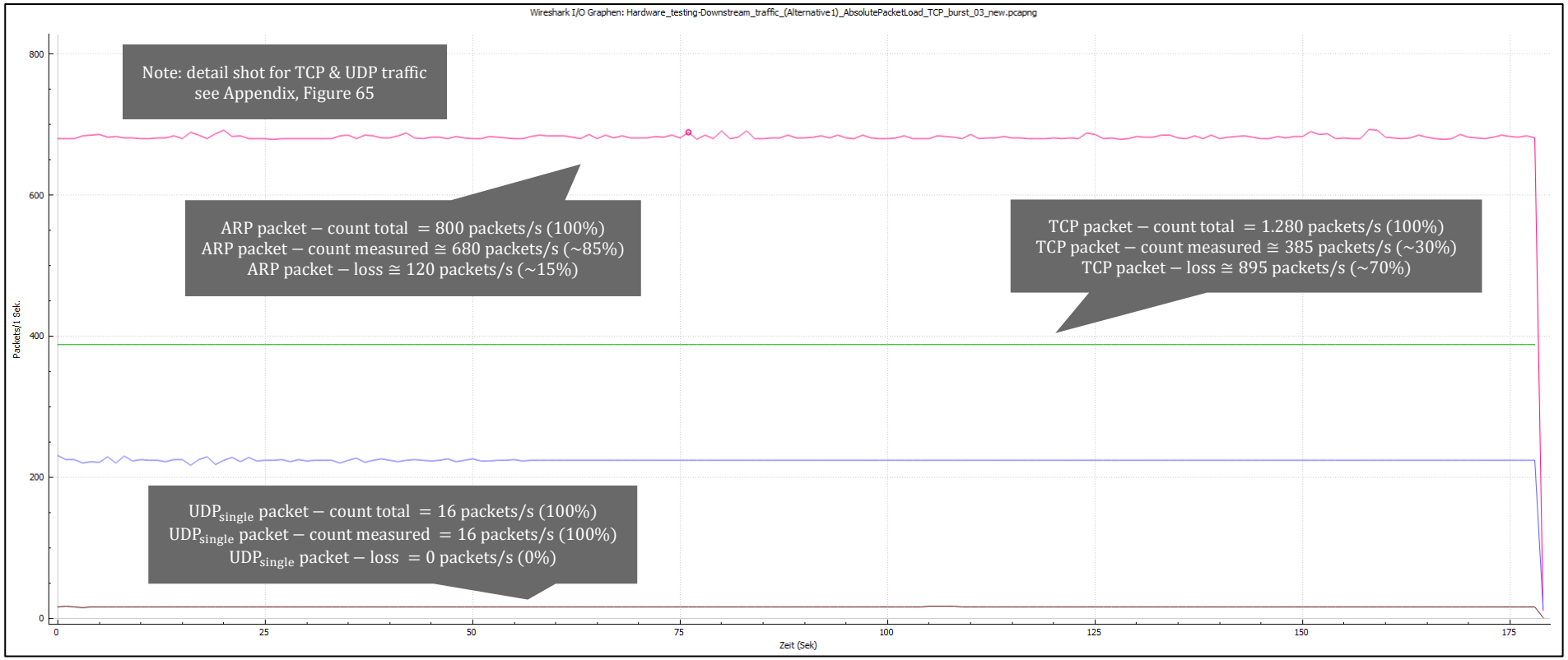

Figure 64: Downstream traffic analysis @ **increasing TCP packet count (Alternative 1), Manufacturer A** - Measurement results (640 TCP packets / 500 ms) – total packet count

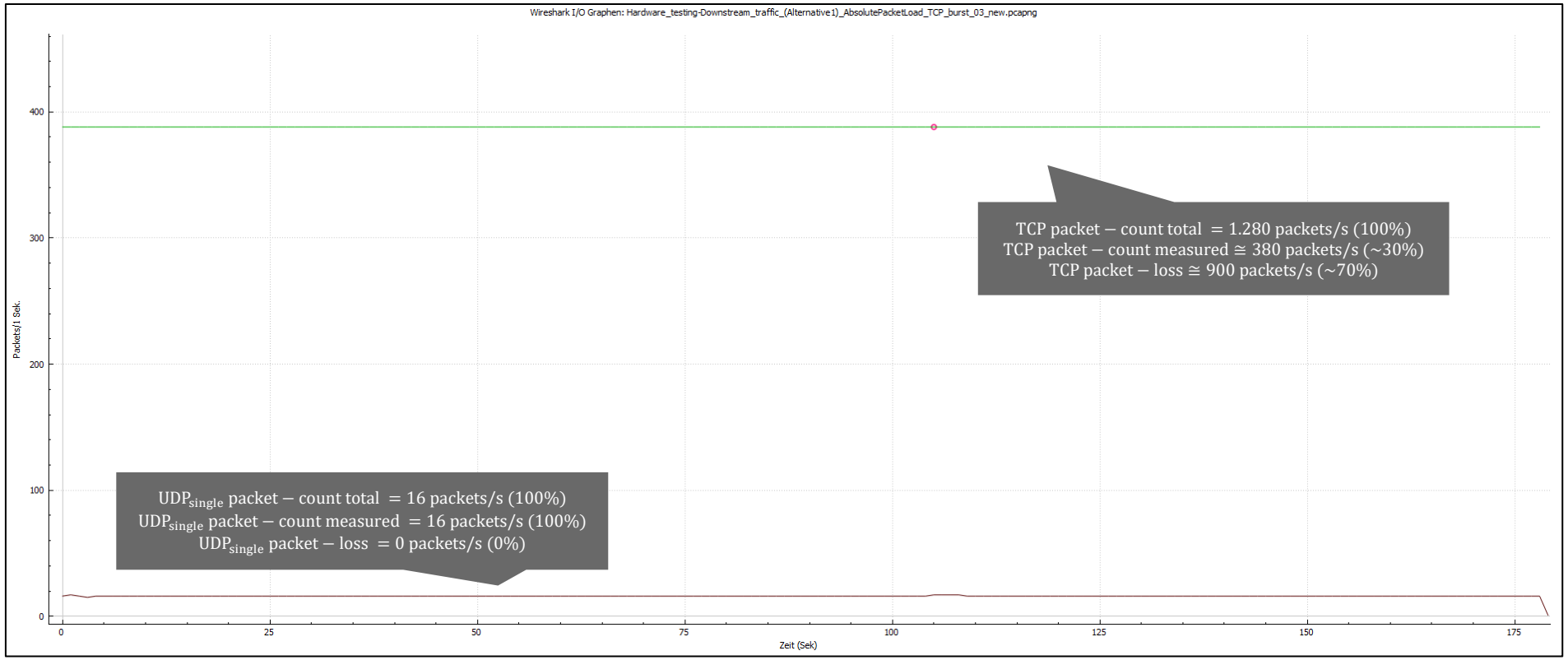

Figure 65: Downstream traffic analysis @ **increasing TCP packet count (Alternative 1), Manufacturer A** - Measurement results (640 TCP packets / 500 ms) – single source packet count

Figure 64 shows that by doubling the total PCS (‘*Packet Count per Second*’) of TCP traffic from ~ 640 packets/s to ~ 1.280 packets/s, packet loss increases from 50% to 70%, while ARP and UDP traffic packet loss remains the same. This shows that the same phenomenon which can be observed in the previous Figure 62 is also apparent in Figure 64, but in a more pronounced form. Hence, the packet prioritization of the APL switch works correctly.

However, the increasing instabilities regarding the packet-throughput behavior of ARP and UDP traffic hint at potential packet loss of said traffic types in favor of TCP traffic also explaining the sudden rise of TC PPS from ~ 320 packets/s to ~ 385 packets/s.

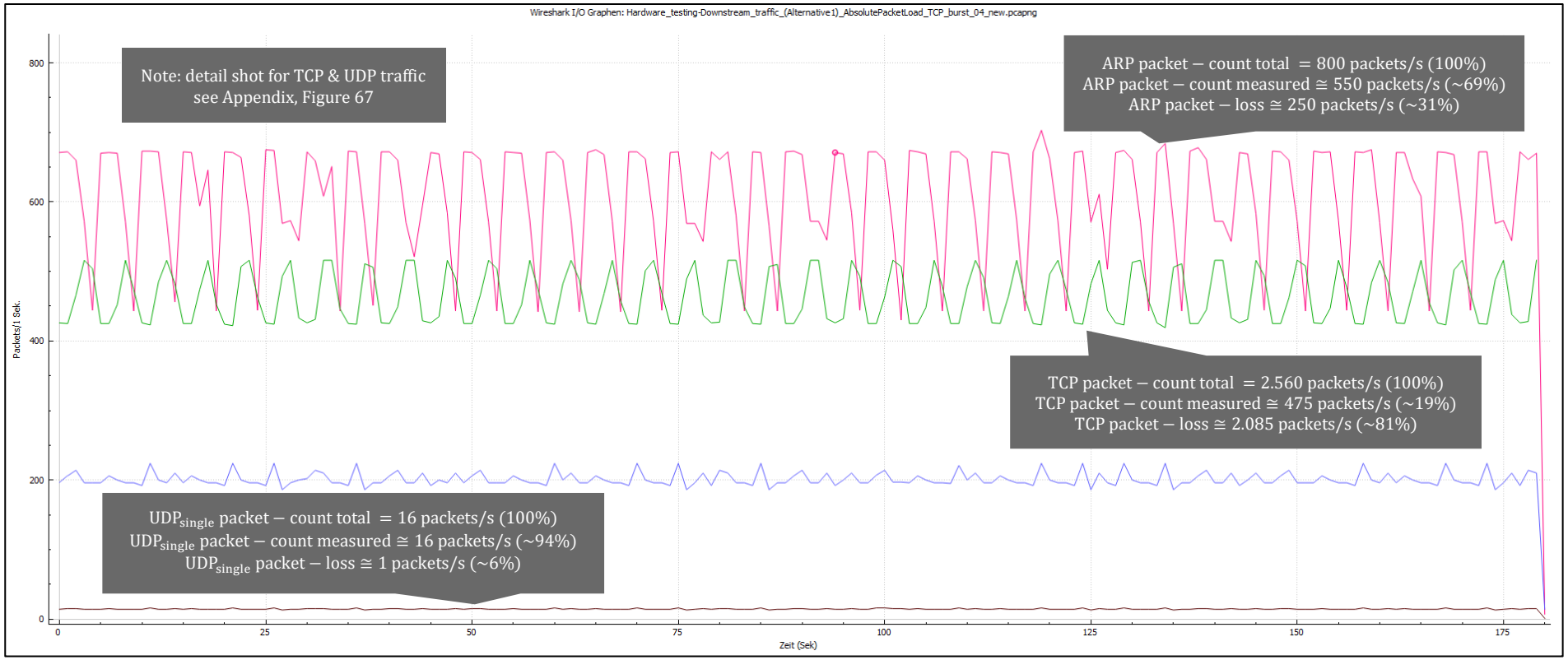

Figure 66: Downstream traffic analysis @ **increasing TCP packet count (Alternative 1), Manufacturer A** - Measurement results (1280 TCP packets / 500 ms) – total packet count

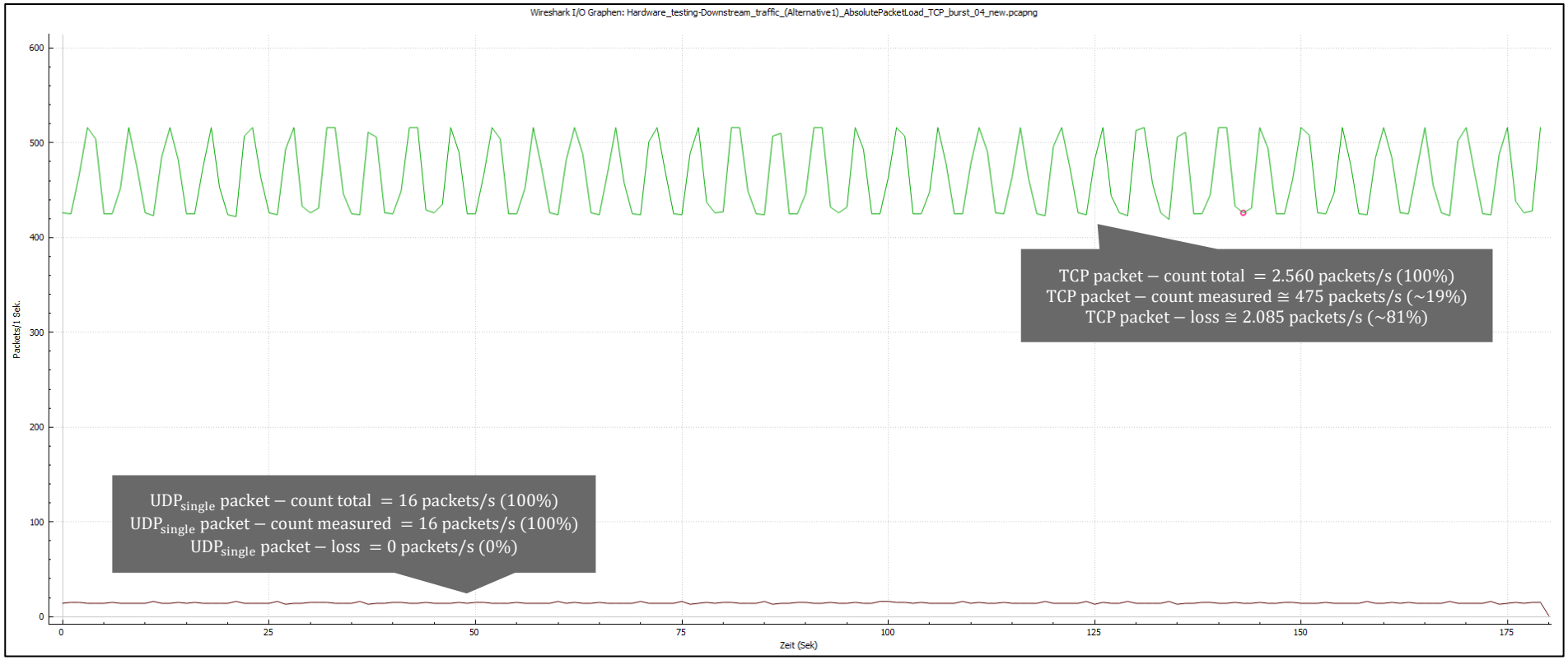

Figure 67: Downstream traffic analysis @ **increasing TCP packet count (Alternative 1), Manufacturer A -** Measurement results (1280 TCP packets / 500 ms) – single source packet count

Figure 66 shows that by quadrupling the total PCS ('*Packet Count per Second*') of TCP traffic from ~ 640 packets/s to ~ 2.560 packets/s, packet loss increases from 50% to 81%, while ARP traffic also drops from 15% to 31%. Additionally, UDP traffic shows high instabilities regarding its packet-throughput behavior also indicating packet loss at 6%. This shows that the same phenomenon which can be observed in the previous Figure 62 is also apparent in Figure 66, but in a more pronounced form.

By even further increasing TCP traffic, not only stable packet-throughput of TCP packets themselves, but also ARP and UDP packets is no longer ensured and leads to packet loss, regardless of packet priority. This is acting against the packet processing behavior according to packet prioritization and leads to the conclusion, that TCP unicasting should not be performed above the '*queueLength*' limit of said APL switch, by increasing the packet count above said limit.

Summary (Figure 62 to Figure 67): The measurements show that stable packet-throughput of high-priority traffic, by gradual increase of TCP traffic up to ~ 2.560 packets/s is always ensured, while working inside the respective hardware limitations of the APL switch.

Exceeding the '*queueLength*' limit of the switch, regarding packet count, leads to discarding of excessive packets arriving at the packet buffer inside the switch (see chapter 5.1.2). This leads to faulty behavior regarding packet processing of the APL switch and can lead to packet loss of high-priority UDP traffic if not limited accordingly.

In conclusion, the APL switch of Manufacturer A fulfills its packet-throughput requirements according to Table 14 regarding the desired packet processing behavior, while working in the constraints of its respective hardware limitations.

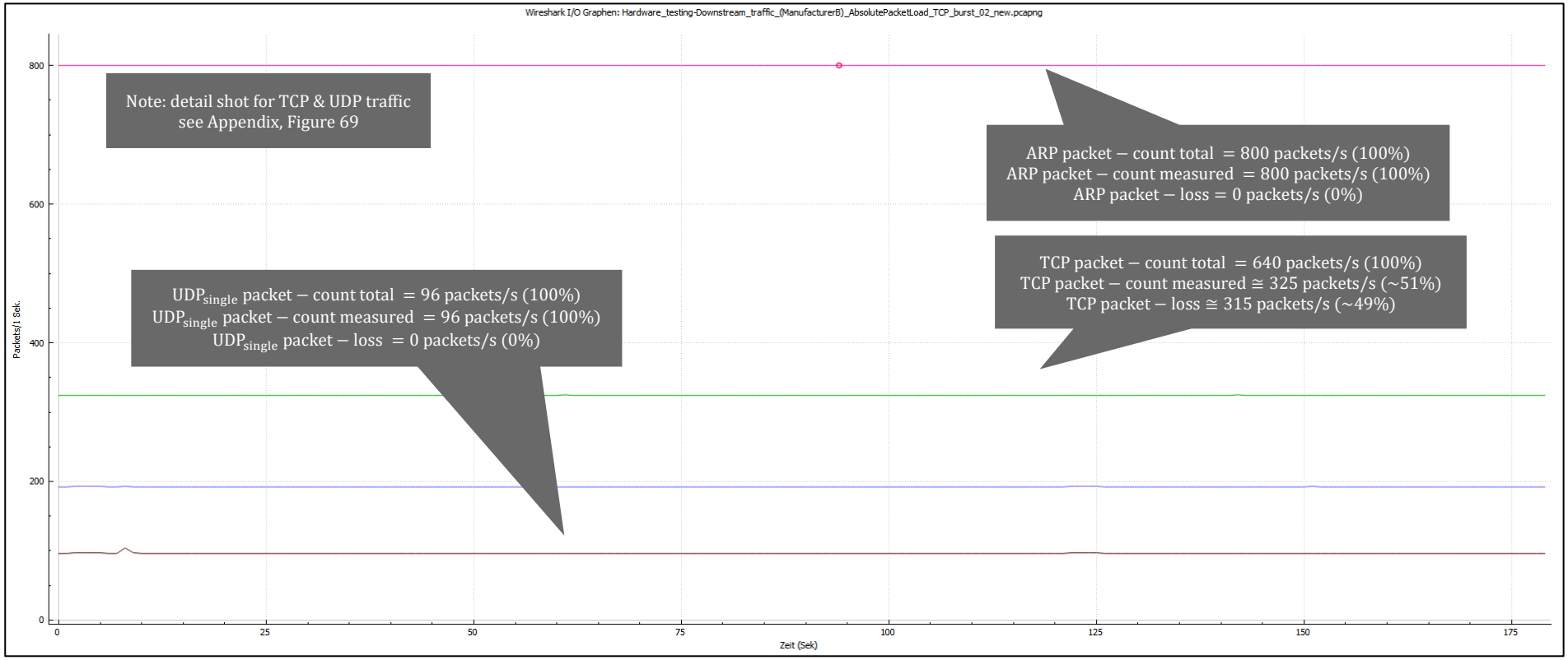

Figure 68: Downstream traffic analysis @ **increasing TCP packet count (Alternative 1), Manufacturer B** - Measurement results (320 TCP packets / 500 ms) – total packet count

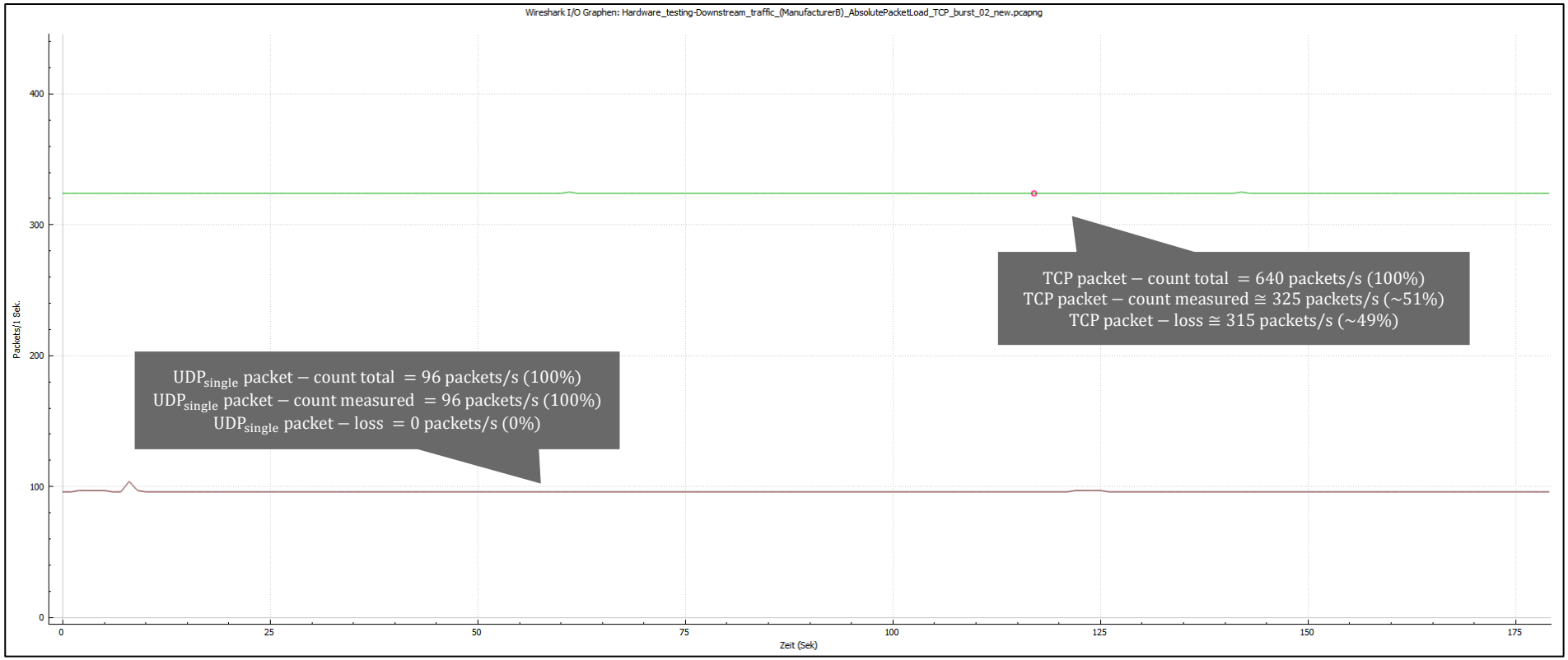

Figure 69: Downstream traffic analysis @ **increasing TCP packet count (Alternative 1), Manufacturer B** - Measurement results (320 TCP packets / 500 ms) – single source packet count

Figure 68 shows that TCP traffic has a packet loss of 55% at a total PCS (‘*Packet Count per Second*’) of 640 packets/s. However, ARP and UDP traffic are unaffected and show no packet loss.

The total PCC (‘*Packet Count per Cycle*’) of UDP, TCP and APL traffic combined stating 444 packets/cycle according to Table 10 and Table 14 stays below the total ‘*bufferCount*’ limit of the APL switch, regarding packet count. The PCCs for each respective packet casting-type (uni-, multi-, broadcast) also stay below the ‘*bufferCount*’ limitations stated in chapter 5.2.1. However, the actual TCP PCS only states a packet count of ~ 290 packets/s. Said packet loss of TCP packets happens due to exceeding the ‘*bufferCount*’ limit of the APL switch, regarding packet load.

Although the total packet count of all packets summed up stays below the '*bufferCount*' limit of 1.024 packets/cycle the total packet load of all packets combined exceeds the total buffer-load memory of the switch according to formula (19) (see chapter 5.2.1.):

$$bufferCount_{\mathrm{B,max}} = bufferCount_{\mathrm{B}}\ x\ packet_{\mathrm{size,max}} \tag{49}$$

$$bufferCount_{\mathrm{B,max}} = 1024 \cdot 256\ Byte\ = 262.144\ Byte \cong 256\ kByte \tag{50}$$

$$packet_{\mathrm{load,measured}} = PPS_{\mathrm{packet,type}} \cdot T_{\mathrm{packet,type}}\ \cdot packet_{\mathrm{size,ARP}} \tag{51}$$

$$packet_{\mathrm{load,total}} = packet_{\mathrm{load,ARP}} + packet_{\mathrm{load,TCP}} + packet_{\mathrm{load,UDP}} \tag{52}$$

$$= 800\frac{packets}{s} \cdot 125\ ms \cdot 88\ Byte\ + 325\frac{packets}{s} \cdot 500\ ms\ \cdot 1.542\ Byte$$

$$+ 4 \cdot\ 96\frac{packets}{s} \cdot 62{,}5\ ms \cdot 88\ Byte\ = 261.487\ Byte \cong 255{,}4\ kByte$$

All incoming packets which exceed the capacity of the already full packet buffer get discarded by the APL switch. However, the APL switch still manages to prioritize the high-priority non-connection based UDP traffic which is essential for the correct functionality of the connected field devices.

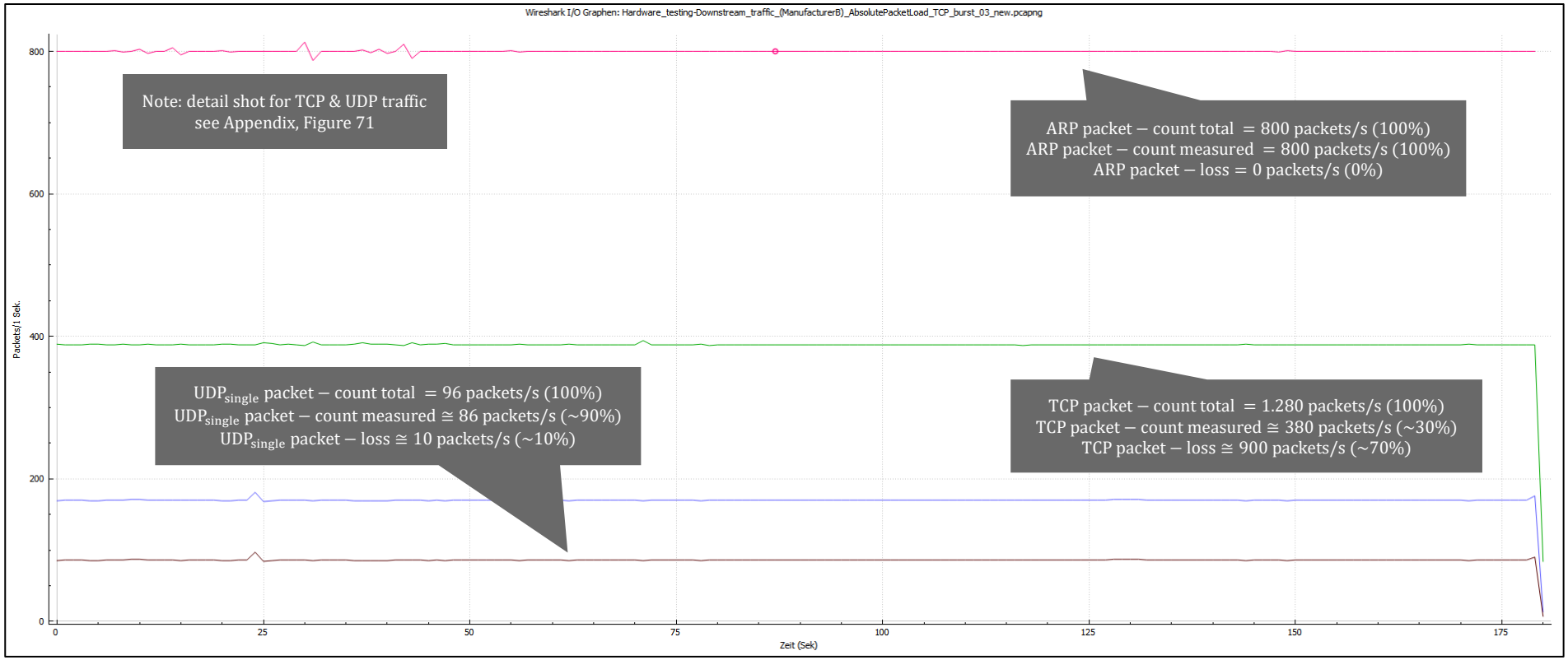

Figure 70: Downstream traffic analysis @ **increasing TCP packet count (Alternative 1), Manufacturer B** - Measurement results (640 TCP packets / 500 ms) – total packet count

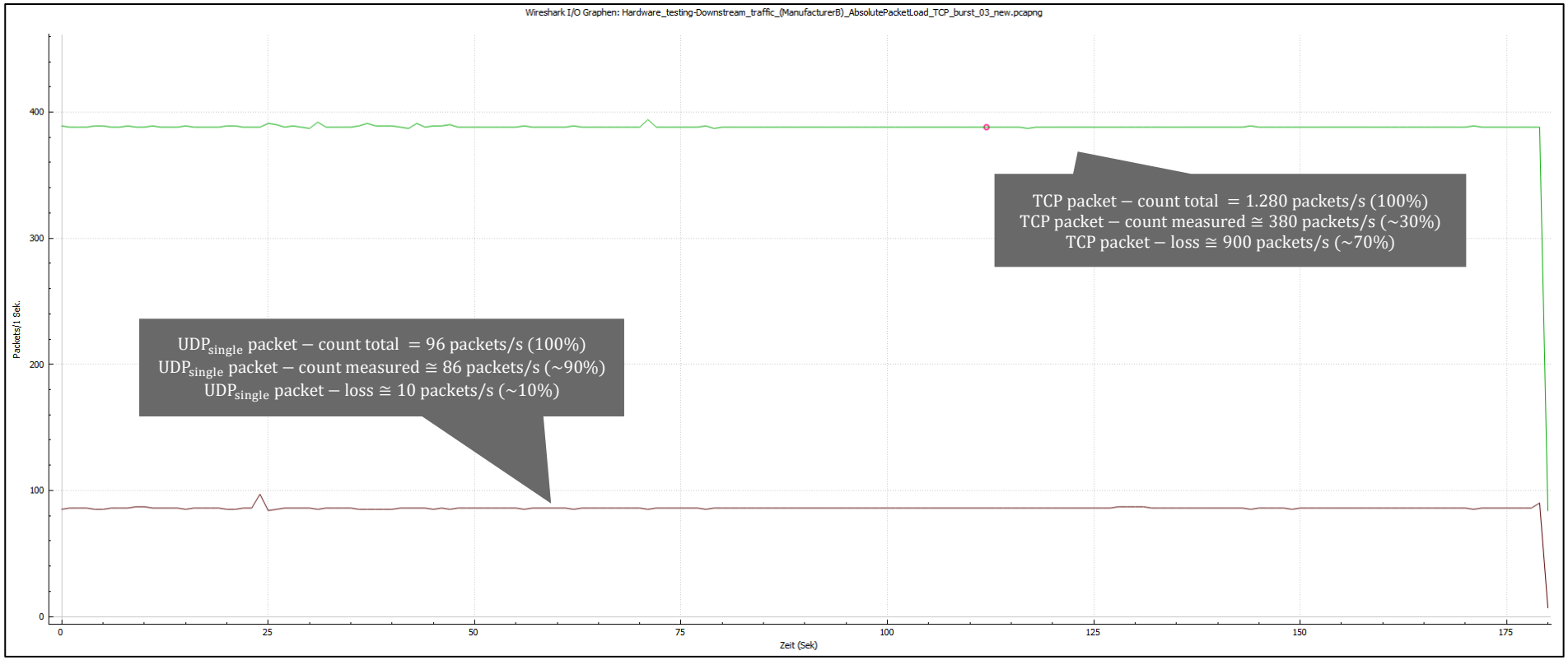

Figure 71: Downstream traffic analysis @ **increasing TCP packet count (Alternative 1), Manufacturer B** - Measurement results (640 TCP packets / 500 ms) – single source packet count

Figure 70 shows that, by doubling the total PCS ('*Packet Count per Second*') of TCP traffic from ~640 packets/s to ~1.280 packets/s, packet loss increases from 49% to 70%, while ARP traffic packet loss remains the same. Additionally, UDP traffic also shows packet loss at 10%.

This shows that the same phenomenon which can be observed in the previous Figure 68 is also apparent in Figure 70, but in a more pronounced form. While working at the '*bufferCount*' limit of the APL switch its packet forwarding behavior struggles with accurate packet prioritization, which leads to packet discarding of high-priority UDP traffic.

This is acting against the packet processing behavior according to packet prioritization and leads to the conclusion, that TCP unicasting should not be performed above the '*bufferCount*' limit of said APL switch, by increasing the packet count above said limit.

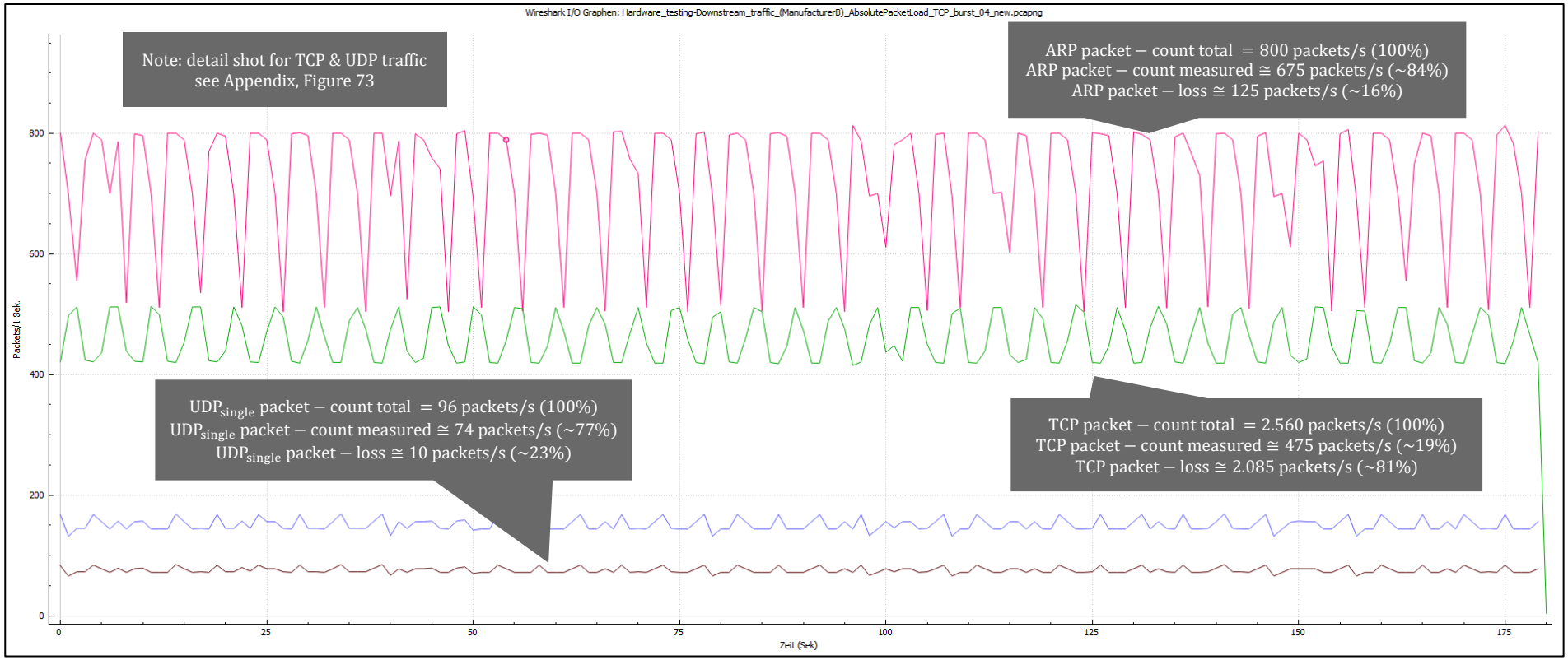

Figure 72: Downstream traffic analysis **@ increasing TCP packet count (Alternative 1), Manufacturer B** - Measurement results (1280 TCP packets / 500 ms) – total packet count

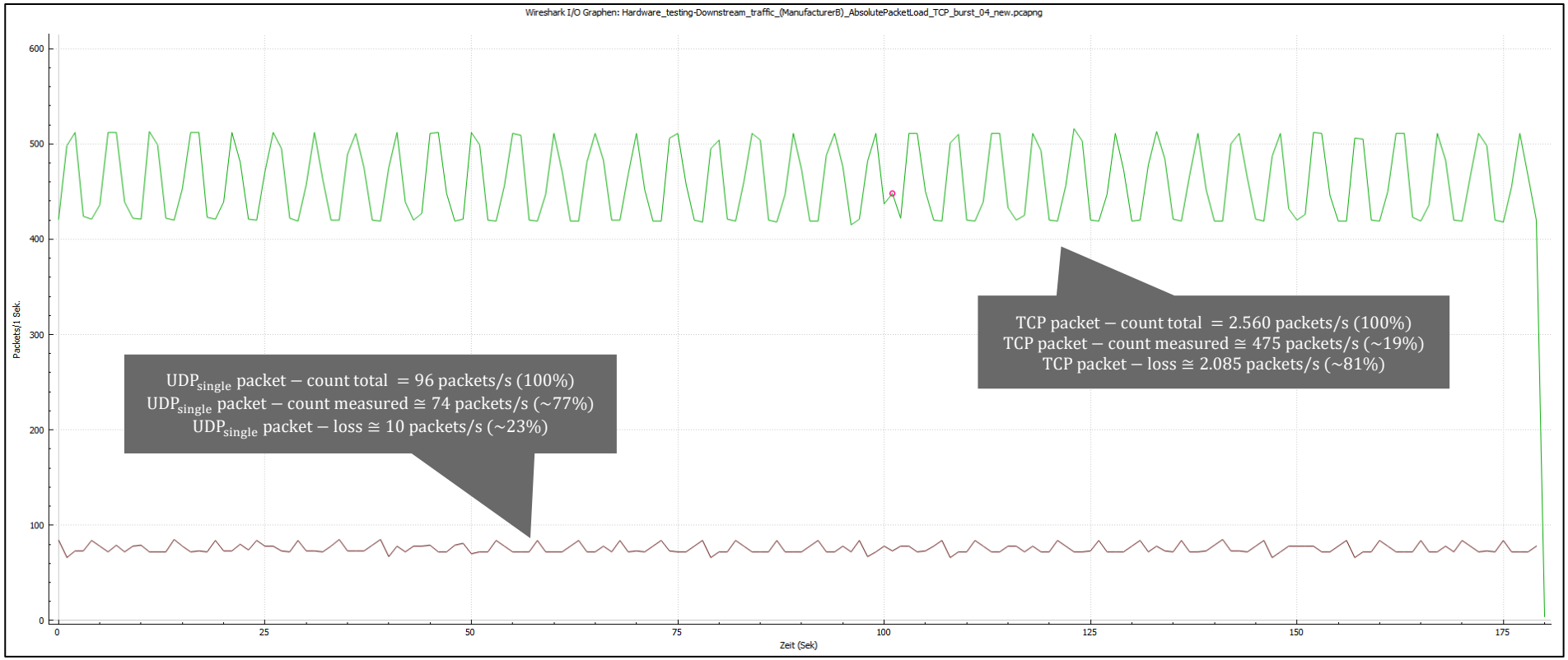

Figure 73: Downstream traffic analysis **@ increasing TCP packet count (Alternative 1), Manufacturer B** - Measurement results (1280 TCP packets / 500 ms) – single source packet count

Figure 72 shows that by quadrupling the total PCS ('*Packet Count per Second*') of TCP traffic from ~ 640 packets/s to ~ 2.560 packets/s, packet loss increases from 49% to 81%, while ARP traffic also drops from 0% to 16%. Additionally, UDP traffic packet loss further increases to 23%. This shows that the same phenomenon which can be observed in the previous Figure 68 is also apparent in Figure 72, but in a more pronounced form.

In addition to the exceeding the '*bufferCount*' limit of the switch hardware, regarding packet load now the packet count limit of 1.024 packets/cycle also gets surpassed by sending with an overall PCC of 1.404 packets/cycle with all traffic types combined, according to Table 10 and Table 14. By even further increasing TCP traffic, not only stable packet-throughput of TCP packets themselves, but also ARP and UDP packets is no longer ensured and leads to packet loss, regardless of packet priority.

This is acting against the packet processing behavior according to packet prioritization and leads to the conclusion, that TCP unicasting should not be performed above the '*bufferCount*' limit of said APL switch, by increasing the packet count above said limit.

Summary (Figure 68 to Figure 73): The measurements show that stable packet-throughput of high-priority traffic, by gradual increase of TCP traffic up to $\sim 2.560\ \text{packets/s}$ is always ensured, while working inside the respective hardware limitations of the APL switch.

Exceeding the total '*bufferCount*' limit of the switch, regarding packet load as well as packet count, leads to discarding of excessive packets arriving at the packet buffer inside the switch (see chapter 5.2.1). This leads to faulty behavior regarding packet processing of the APL switch and can lead to packet loss of high-priority UDP traffic if not limited accordingly.

In conclusion, the APL switch of Manufacturer B fulfills its packet-throughput requirements according to Table 14 regarding the desired packet processing behavior, while working in the constraints of its respective hardware limitations.

A.4 Hardware test – Upstream traffic (Alternative 1), Manufacturers A&B

Disclaimer: This chapter documents traffic measurement results in full detail and is meant to provide extensive background information on the results summarized in chapter 7.2 and 7.5. However, this level of detail is not necessary for understanding the essence of said results and serves only for the profound understanding of the measurement evaluation.

The following hardware measurement follows the network setup according to Figure 17 (chapter 6.1).

Note: A detailed description for better understanding and interpreting the measurement result figures is presented in chapter 6.5

A.4.1 Packet-throughput analysis - Packet processing @ *'SimultaneousTrafficBurst*' condition

The given Ethernet-APL switches handle packet data for up to 24 field devices send via the Ethernet-APL spur ports in both packet stream directions. For handling multiple incoming packet streams, while avoiding packet loss, the buffer must be able to manage a required number of packets without packet discarding.

The minimum buffer size needed for avoiding packet loss will be determined via the *'SimultaneousTrafficBurst*' condition, stating up to $384\ \mathrm{packets/cycle}$ calculated for 24 ports (one packet send each 62,5 ms) at $10\ \mathrm{Mbit/s}$ transmission speed.

Hardware test shall validate if the switches manage to uphold stable packet-throughput at packet loads up to the *'SimultaneousTrafficBurst*' condition, while ensuring no packet loss of higher priority traffic types at the given hardware limitations.

Note: While analyzing the packet processing of a switch, packet load as well as packet count must be considered for accurate validation of potential packet loss.

For example, packet loss can already occur at packet loads smaller than $128\ \mathrm{kByte}$ by sending more than 128 packets to one packet queue of the switch from Manufacturer A, thereby exceeding its '*queueLength*' limit and provoking packet loss. Said packets might only have a packet load of $64\ \mathrm{Byte}$ per packet, resulting in a total load of $8\ \mathrm{kByte}$,which is much smaller than the packet load of most test scenarios of the '*SimultaneousTrafficBurst*' condition.

Thus, the measurement results of the conducted test scenarios will be analyzed based on packet load as well as packet count.

The following packet load parameters stated in Table 15 have been used for testing:

Table 15: Upstream traffic analysis (Alternative 1) Manufacturers A&B – Traffic parameters

	UDP real-time data, single sensor			
number of field devices	24 (Manufacturer A)		4 (Manufacturer B)	
user priority	6			
total packet payload **(TPP)**	46 Byte (data) + 42 Byte (framing/transmission) = 88 Byte			
packet cycle time **(PCT)**	62,5 ms			
Packet Count per Cycle **(PCC)**	**4 · 1 / 8 · 1 / 16 · 1 / 24 · 1 packets/cycle** (**Manufacturer A**)		**1 · 4 / 2 · 4 / 4 · 4 / 6 · 4 packets/cycle** (**Manufacturer B**)	
Packet data Payload per Cycle **(PPC)**	~184 / ~368 / ~736 / ~1.104 Byte/cycle			
total frame payload per cycle **(FPC)**	~352 / ~704 / ~1.408 / ~2.112 Byte/cycle			
Packet Count per Second **(PCS)**	**64 / packets/s**	**128 packets/s**	**256 packets/s**	**384 packets/s**
packet data payload per second **(PPS)**	64 packets / s · 46 Byte = 2.944 Byte/s (**~2,9 kByte/s**)	128 packets / s · 46 Byte = 5.888 Byte/s (**~5,75 kByte/s**)	256 packets / s · 46 Byte = 11.776 Byte/s (**~11,5 kByte/s**)	384 packets / s · 46 Byte = 17.664 Byte/s (**~17,3 kByte/s**)
total frame payload per second **(FPS)**	64 packets / s · 88 Byte = 5.632 Byte/s (**~5,5 kByte/s**)	128 packets / s · 88 Byte = 11.264 Byte/s (**~11,0 kByte/s**)	256 packets / s · 88 Byte = 22.528 Byte/s (**~22,0 kByte/s**)	384 packets / s · 88 Byte = 33.792 Byte/s (**~33,0 kByte/s**)

Note: Due to the varying amount of ETH/APL media converters for testing Manufacturer A and B, the number of field device emulators and thus the distribution of PCC ('*Packet Count per Cycle*') may vary. Regardless the total sum of UDP traffic stays the same for both.

Table 15 shows all relevant traffic scenario parameters for the follow-up measurement. The PCS ('*Packet Count per Second*') shows that up to 384 packets of simultaneous UDP traffic gets generated complying to the '*SimultaneousTrafficBurst*' test condition.

The following figures show the packet-throughput behavior of the switch when using the stated in Table 15. Figure 74 to Figure 77 show the packet-throughput behavior of Manufacturer A in upstream direction. Figure 78 to Figure 81 show the packet-throughput behavior of Manufacturer B in upstream direction.

The blue line shows the entire captured UDP traffic send by the field device emulators to the workstation at $\sim 64 \ldots 384$ packets/s. The brown line shows the UDP traffic from one field device emulator at 16 packets/s for Manufacturer A and 96 packets/s for Manufacturer B.

Note: The total packet count of recorded UDP traffic in upstream direction resembles all field device emulators sending packets.

The TAP device which was used for traffic measurement was directly placed at the APL switch Fast Ethernet egress port (according to Figure 17). Thus, all outgoing UDP packets of the field device emulators, which are processed and forwarded by the APL switch are captured.

Besides measuring the packet count of one filed device the purpose of additional capturing the total UDP traffic is for better recognition of stability issues, regarding packet-throughput and potential packet loss.

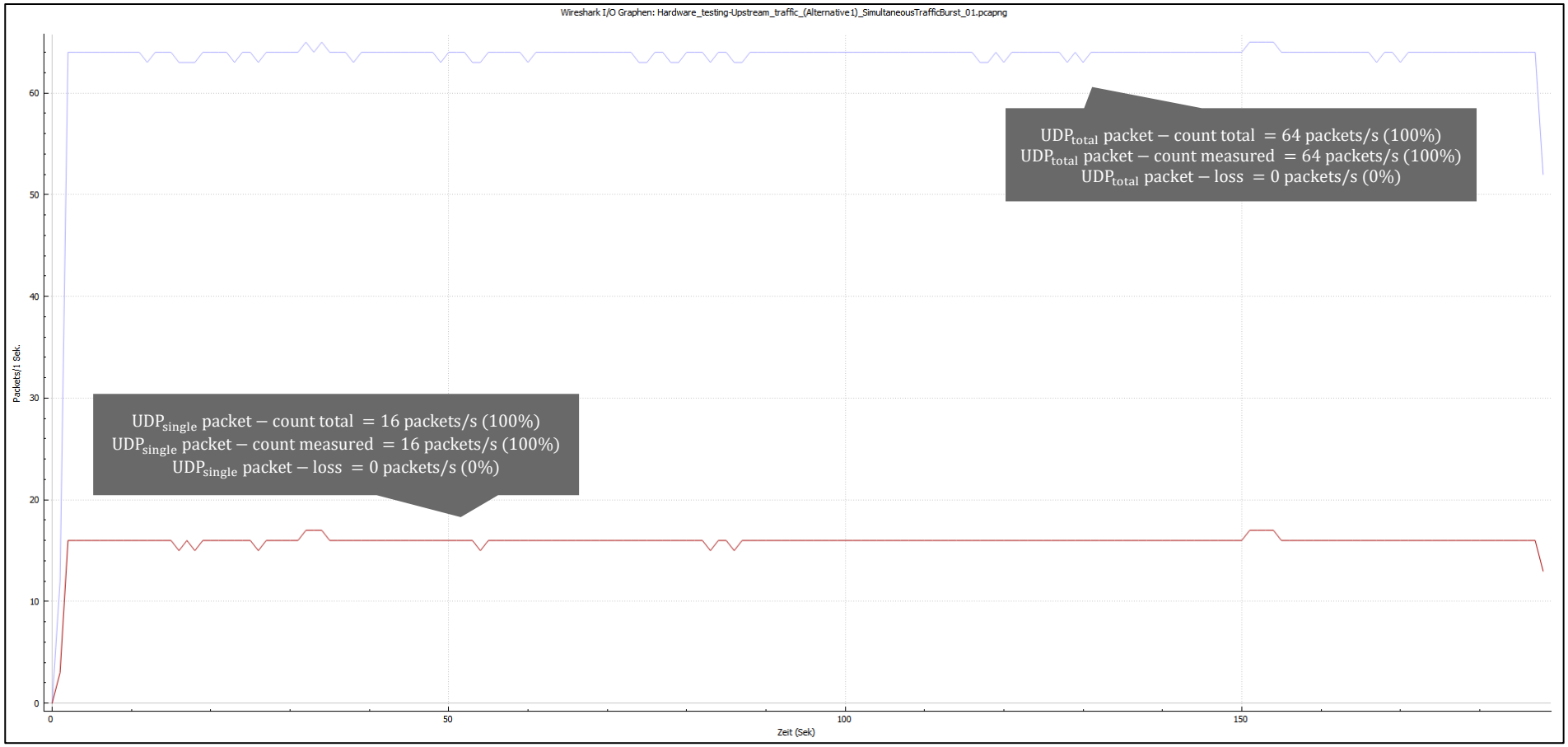

Figure 74: Upstream traffic analysis **@ *'SimultaneousTrafficBurst'* condition (Alternative 1), Manufacturer A** - Measurement results (4·1 UDP packets / 62,5 ms) – total packet count

Figure 74 shows that UDP traffic does not struggle with packet loss at a total PCS ('*Packet Count per Second*') of 64 packets/s . Hence, all UDP packets are forwarded successfully by the APL switch.

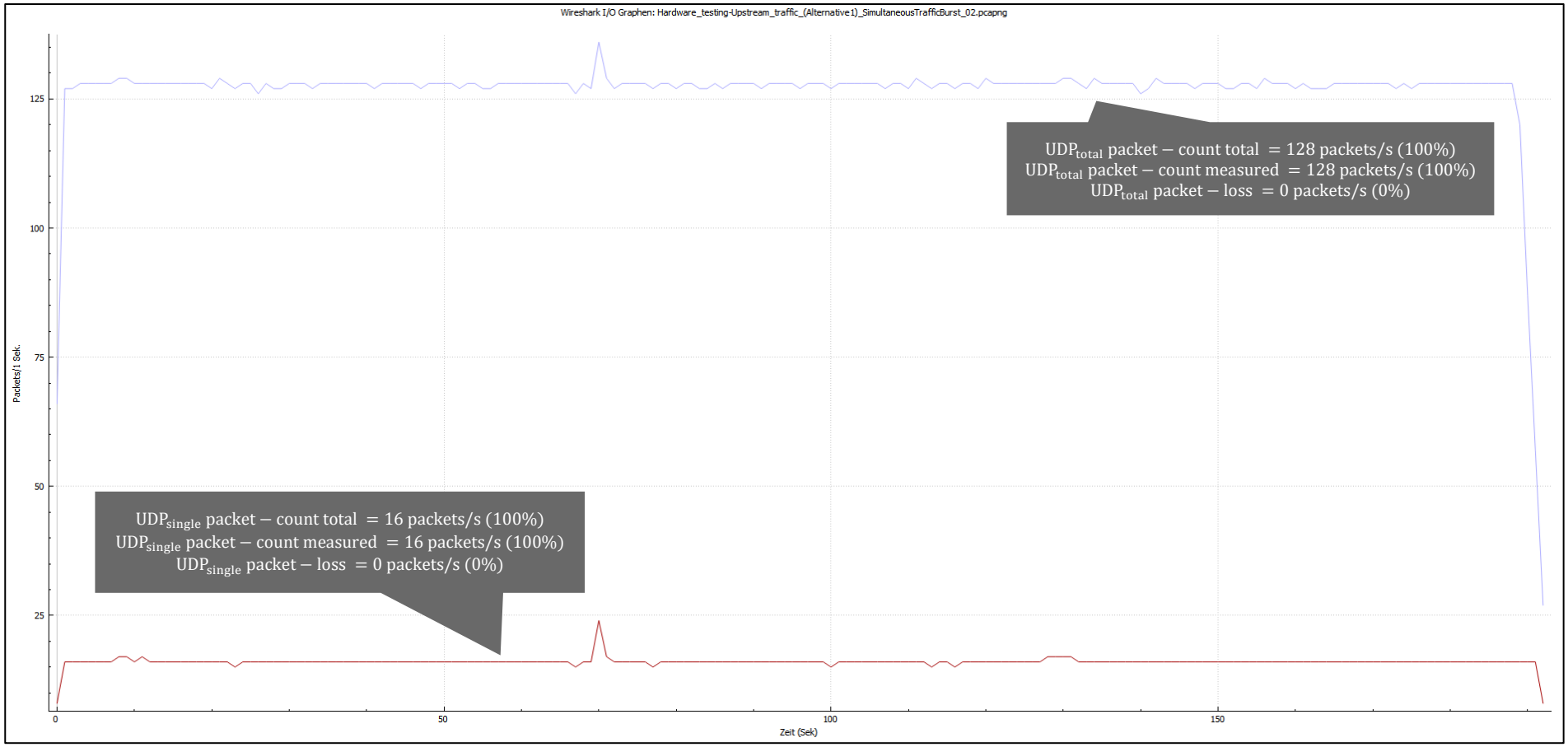

Figure 75: Upstream traffic analysis **@ *'SimultaneousTrafficBurst'* condition (Alternative 1), Manufacturer A** - Measurement results (8·1 UDP packets / 62,5 ms) – total packet count

Figure 75 shows that by doubling the total PCS ('*Packet Count per Second*') of UDP traffic from ~64 packets/s to ~128 packets/s, packet loss of UDP traffic still does not occur. Hence, all UDP packets are forwarded successfully by the APL switch.

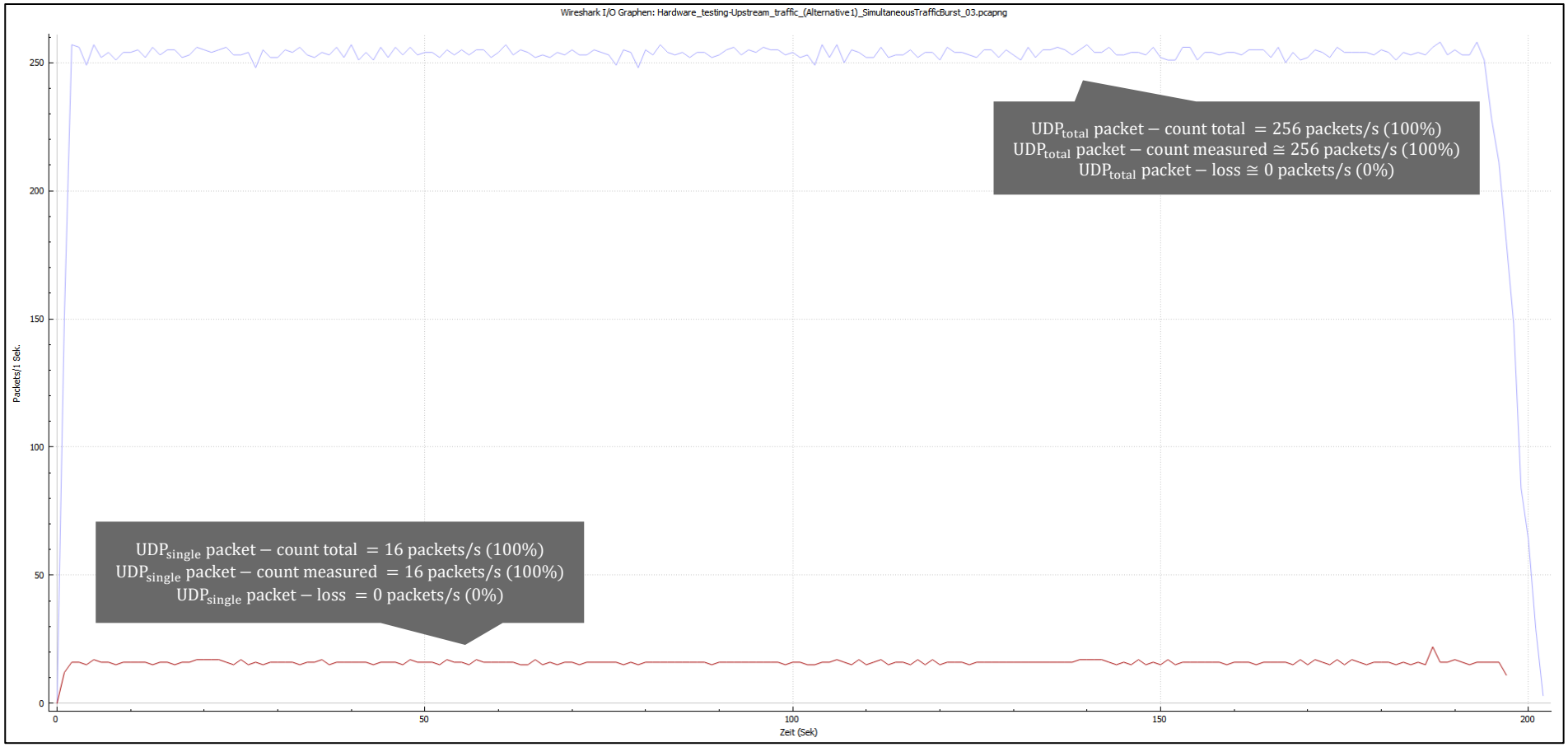

Figure 76: Upstream traffic analysis **@ *'SimultaneousTrafficBurst'* condition (Alternative 1), Manufacturer A** - Measurement results (16·1 UDP packets / 62,5 ms) – total packet count

Figure 76 shows that by quadrupling the total PCS ('*Packet Count per Second*') of UDP traffic from ~ 64 packets/s to ~ 256 packets/s, packet loss of UDP traffic still does not occur. Hence, all UDP packets are forwarded successfully by the APL switch.

However, by comparing Figure 74 to Figure 76, the packet-throughput behavior of UDP traffic shows increasing instabilities, which hints at a potential packet loss by further increasing the packet load by increasing the number of used field devices.

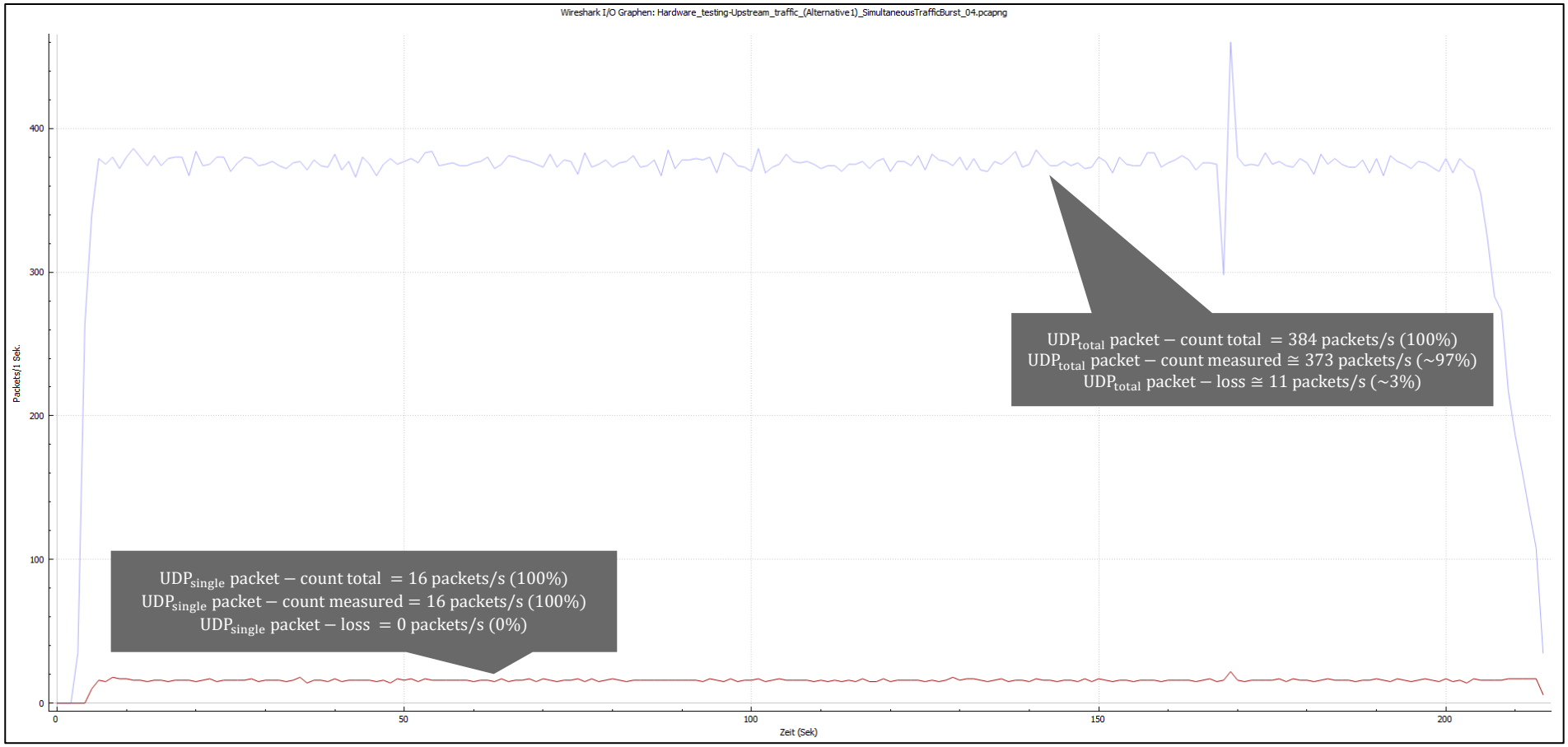

Figure 77: Upstream traffic analysis **@ *'SimultaneousTrafficBurst'* condition (Alternative 1), Manufacturer A** - Measurement results (24·1 UDP packets / 62,5 ms) - total packet count

Figure 77 shows that by sixfold the total PCS ('*Packet Count per Second*') of UDP traffic from ~64 packets/s to ~384 packets/s, packet loss increases from 0% to 3%. The total packet count of UDP traffic is 384 packets/cycle, stated in Table 15. This value is below the '*bufferLength*' limit of the APL switch (1.800 packets/cycle). Therefore, no packet discarding occurs in the packet buffer of the APL switch (see chapter 5.1.1).

Besides APL switch parameters regarding packet size (packet load and packet count), the PPT ('*Packet Processing Time*') also needs to be considered while analyzing packet-throughput behavior of Manufacturer A, regarding the '*queueLength*' limit of the APL switch (see chapter 5.1.2). If the APL switch does not forward incoming packets fast enough, subsequent packets of the next cycle can overlap with already queued packets resulting in potential packet discards if the packet queues consistently get filled until they're at their full storing capacity.

The following PPTs have been calculated with the help of the hardware delay time stated in Table 2 (chapter 5.4) and the traffic parameters stated in Table 15.

$$PPT_{\text{type}} = x_{\text{packets,type}} \cdot \left(t_{\text{bridge}} + t_{\text{port}} + t_{\text{cable}} + t_{\text{prop}}\right) \quad (53)$$

Note: The store and forward bridge delay is calculated, based on the datarate of the internal Ethernet bridge of the APL switch, handling the transition between its Fast Ethernet ingress port, working at $100\ \text{Mbit/s}$, and Ethernet-APL egress spur ports, working at $10\ \text{Mbit/s}$.

To empty all packet queues for one cycle, the following PPT derives:

$$\rightarrow PPT_{\text{UDP}} = x_{\text{packets,UDP}} \cdot \left(6\ \mu s + \frac{88\ Byte}{100\ Mbit/s} + 4\ \mu s + 1{,}33\ \mu s\right) \quad (54)$$

$$\rightarrow PPT_{\text{UDP}} = 24 \cdot \left(6\ \mu s + \frac{88\ Byte}{100\ Mbit/s} + 4\ \mu s + 1{,}33\ \mu s\right) \cong 0{,}44\ ms \quad (55)$$

$$\rightarrow (PPT_{\text{total}} = PPT_{\text{UDP}} \cong 0{,}44\ ms) < T_{UDP} = 62{,}5\ ms \quad (56)$$

The calculation shows that theoretical PPT ('*Packet Processing Time*') needed for emptying all ingress queues stays below the packet cycle time, thus preventing packet discarding by not exceeding the '*queueLength*' limit of the APL switch via packet overlapping inside the packet queues.

By dividing the UDP cycle time with the UDP PPT the following total UDP packet count derives:

$$\rightarrow\ x_{\text{packets,UDP,max}} = \frac{T_{\text{UDP}}}{PPT_{\text{UDP}}} \quad (57)$$

$$\rightarrow x_{\text{packets,UDP,max}} = \frac{62{,}5\ ms/\ cycle}{\mathbf{6}\ \mu s + \frac{88\ Byte/packet}{100\ Mbit/s} + 4\ \mu s + 1{,}33\ \mu s} = \frac{62{,}5\ ms}{\mathbf{6}\ \mu s + 7{,}04\ \mu s + 4\ \mu s + 1{,}33\ \mu s}\frac{packets}{cycle} \cong 3.402\frac{packets}{cycle} \quad (58)$$

The calculation shows that theoretical total packet count processable by the APL switch grants a UDP PPC of 3.402 packets/cycle, thus yielding a total UDP PPS of 54.437 packets/s in terms of packet processing capability. However, the actual TCP PCS only states a packet count of ~ 373 packets/s, resulting in a packet

loss of ~ 11 packets/s. According to the calculations, the APL switch of Manufacturer A should fulfill all test scenarios, regarding the packet load stated in Table 15 without packet loss.

The previous mentioned increase of UDP packet-throughput instability, which are visible while working with 16 to 24 field devices, are not related to a faulty behavior of the switches but rather to the missing synchronization of the emulated field devices in Ostinato. Unfortunately, Ostinato showed significant performance problems when managing more than 8 Ethernet interfaces, sending packet load at the same time, which resulted in strong fluctuations of the packet-throughput.

Hence, not all UDP packets could be generated in time for subsequent processing at the APL switch, which caused packet loss of system relevant UDP traffic. Basically, it can be said that the APL switch nevertheless successfully receives and forwards all packets sent via the field devices, without being responsible for said packet loss.

Summary (Figure 74 to Figure 77): The measurements show, that stable packet-throughput of simultaneous incoming traffic, by gradual increase of UDP traffic up to ~ 384 packets/s, according to the '*SimulaneousTrafficBurst*' condition, is always ensured.

The instabilities regarding UDP packet-throughput happen due to missing synchronization of the field device emulators sending packets (see note in chapter 6.1). Exceeding the number of field devices used for generating packet load above 8 resulted in increasing packet-throughput instabilities, resulting in packet loss, caused by the missing synchronization of packet load generation via multiple packet streams by the frame generator tool Ostinato. However, the total PCS ('*Packet Count per Second*') of UDP traffic stating ~ 373 packets/s shows that the APL switch is able to successfully process all remaining UDP packets without packet loss, according to its packet processing hardware limitations (54.437 packets/s) calculated in Figure 77.

In conclusion, the APL switch of Manufacturer A fulfills its packet-throughput requirements according to Table 15 regarding the desired packet processing behavior.

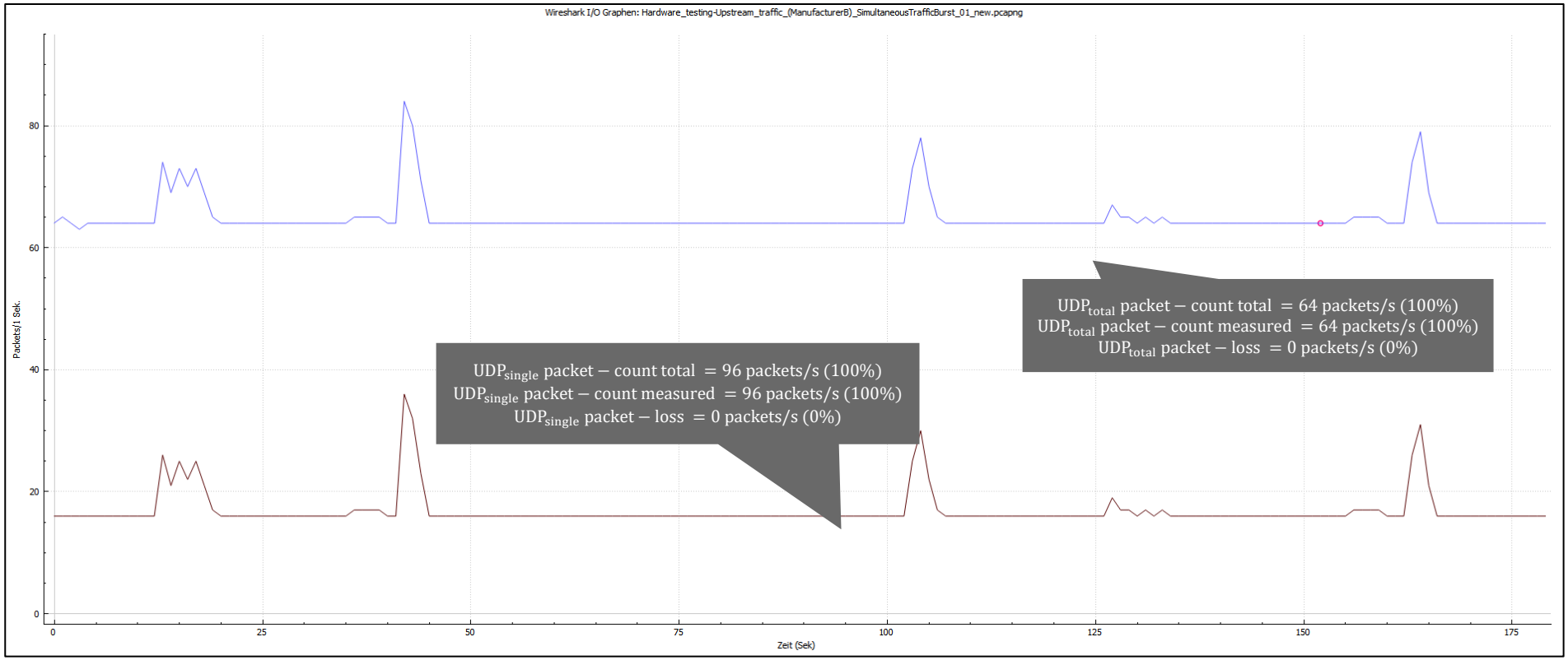

Figure 78: Upstream traffic analysis **@ *'SimultaneousTrafficBurst'* condition (Alternative 1), Manufacturer B** - Measurement results (4·1 UDP packets / 62,5 ms) – total packet count

Figure 78 shows that UDP traffic does not struggle with packet loss at a total PCS ('*Packet Count per Second*') of 64 packets/s . Hence, all UDP packets are forwarded successfully by the APL switch.

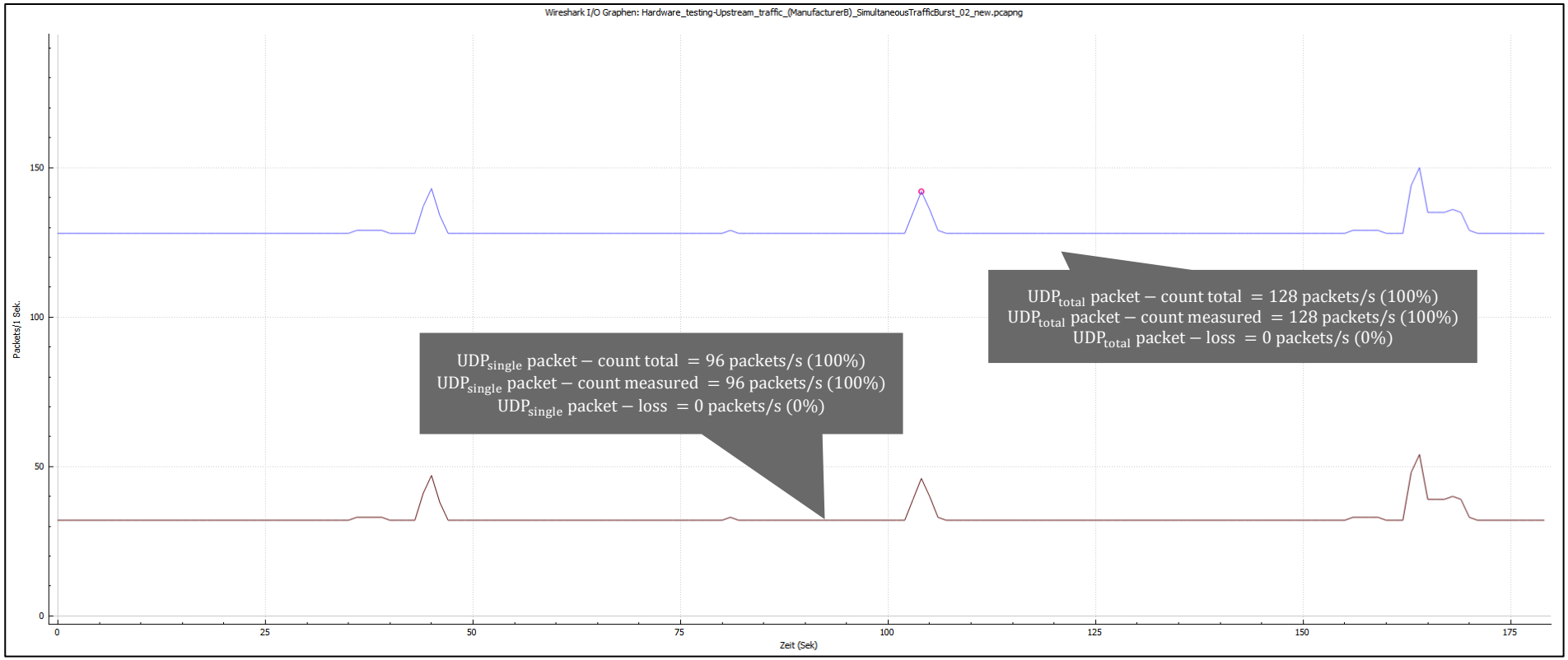

Figure 79: Upstream traffic analysis **@ *'SimultaneousTrafficBurst'* condition (Alternative 1), Manufacturer B** - Measurement results (8·1 UDP packets / 62,5 ms) – total packet count

Figure 79 shows that by doubling the total PCS ('*Packet Count per Second*') of UDP traffic from ~ 64 packtes/s to ~ 128 packets/s, packet loss of UDP traffic still does not occur. Hence, all UDP packets are forwarded successfully by the APL switch.

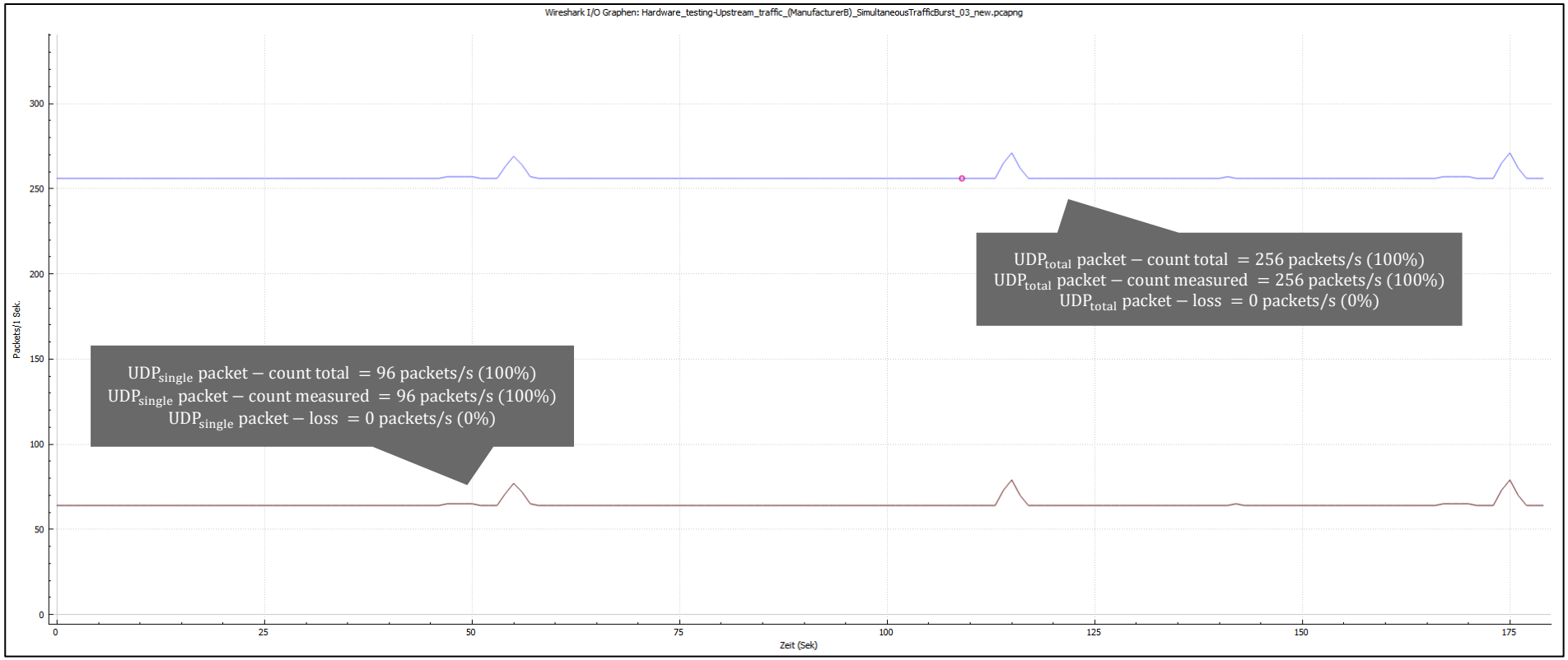

Figure 80: Upstream traffic analysis **@ *'SimultaneousTrafficBurst'* condition (Alternative 1), Manufacturer B** - Measurement results (16·1 UDP packets / 62,5 ms) – total packet count

Figure 80 shows, at by quadrupling the total PCS ('*Packet Count per Second*') of UDP traffic from ~ 64 packets/s to ~ 256 packets/s, packet loss of UDP traffic still does not occur. Hence, all UDP packets are forwarded successfully by the APL switch.

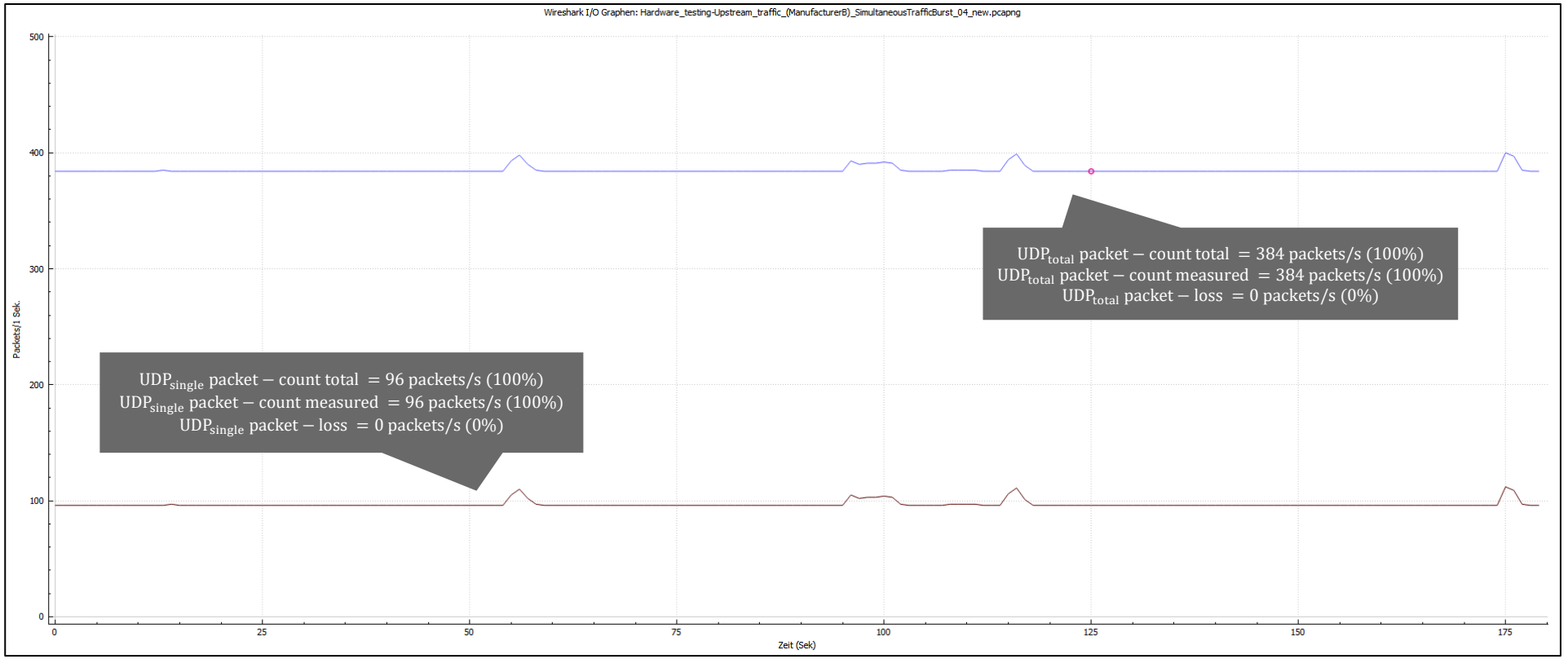

Figure 81: Upstream traffic analysis **@ *'SimultaneousTrafficBurst'* condition (Alternative 1), Manufacturer B** - Measurement results (24·1 UDP packets / 62,5 ms) - total packet count

Figure 81 shows that by sixfold the total PCS ('*Packet Count per Second*') of UDP traffic from ~64 packets/s to ~384 packets/s, packet loss of UDP traffic still does not occur. Hence, all UDP packets are forwarded successfully by the APL switch.

Summary: (Figure 78 to Figure 81): The measurements show, that stable packet-throughput of simultaneous incoming traffic, by gradual increase of UDP traffic up to ~ 384 packets/s, according to the '*SimulaneousTrafficBurst*' condition, is always ensured. The instabilities regarding UDP packet-throughput happen due to missing synchronization of the field device emulators sending packets (see note in chapter 6.2).

However, by staying below 8 field devices used for generating packet load, packet loss is prevented (refer to measurement summary of Figure 74 to Figure 77). The total packet count of UDP traffic is 24 packets/cycle, stated in Table 15. This value is below the '*bufferCount*' limit of the APL switch (1.024 packets/cycle). Therefore, no packet discarding occurs in the packet buffer of the APL switch (see chapter 5.2.1).

In conclusion, the APL switch of Manufacturer B fulfills its packet-throughput requirements according to Table 15 regarding the desired packet processing behavior.

A.4.2 Packet-throughput analysis – absolute packet processing limit

Similar to downstream traffic, hardware tests in upstream direction shall validate if the switches manage to uphold stable packet-throughput of higher priority traffic while gradually increasing the packet load.

However, this time only high priority traffic gets gradually increased to verify the absolute maximum packet processing limitation.

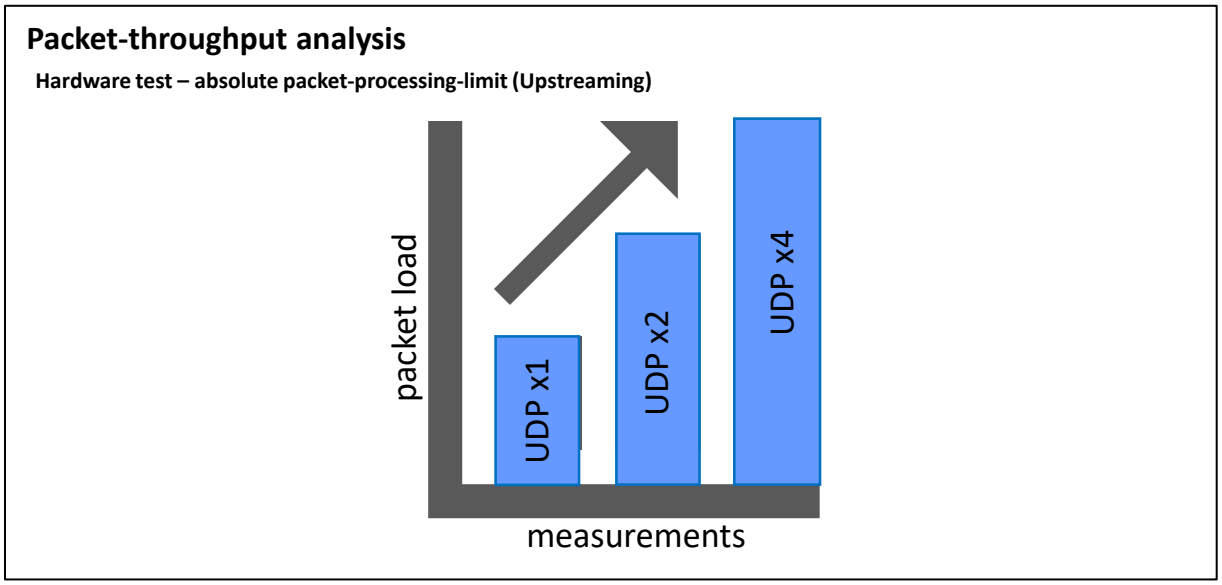

Figure 82: Gradual increase of high-priority traffic for investigation of absolute packet processing limit

For easier understanding Figure 82 resembles the gradual increase of high-priority UDP traffic packet load.

A.4.2.1 Packet processing @ decreasing UDP cycle time

The following tests have been conducted by increasing UDP traffic through decreasing its cycle time. The traffic parameters used in these tests are stated in Table 16.

Table 16: Downstream traffic analysis (Alternative 1), Manufacturers A&B – Traffic parameters, increasing UDP traffic @ varying cycle time

	UDP real-time data, *single sensor*		
user priority	6		
total packet payload **(TPP)**	46 Byte (data) + 42 Byte (framing/transmission) = 88 Byte		
packet cycle time **(PCT)**	**31,25 ms / 15,625 ms / 7,8125 ms**		
Packet Count per Cycle **(PCC)**	24 · 1 packets/cycle (Manufacturer A)	6 · 4 packets/cycle (Manufacturer B)	
Packet data Payload per Cycle **(PPC)**	24 packets / cycle · 46 Byte = 1.104 Byte/cycle (~1,1 kByte/c)		
total frame payload per cycle **(FPC)**	24 p/c · 88 Byte = 2.112 Byte/cycle (~2 kByte/cycle)		
Packet Count per Second **(PCS)**	**768 packets/s**	**1536 packets/s**	**3072 packets/s**
packet data payload per second **(PPS)**	768 p/s · 46 Byte = 35.328 $Byte/s$ (~**34,5** $\boldsymbol{kByte/s}$)	1536 p/s · 46 Byte = 70.656 $Byte/s$ (~**69,0** $\boldsymbol{kByte/s}$)	3072 p/s · 46 Byte = 141.312 $Byte/s$ (~**138,0** $\boldsymbol{kByte/s}$)
total frame payload per second **(FPS)**	768 p/s · 88 Byte = 67.584 $Byte/s$ (~**66** $\boldsymbol{kByte/s}$)	1536 p/s · 88 Byte = 135.168 $Byte/s$ (~**132,0** $\boldsymbol{kByte/s}$)	3072 p/s · 88 Byte = 270.336 $Byte/s$ (~**264** $\boldsymbol{kByte/s}$)

The following figures show the packet-throughput behavior of the switch when using the stated in Table 16. Figure 83 to Figure 88 show the packet-throughput behavior of Manufacturer A in upstream direction. Figure 89 to Figure 94 show the packet-throughput behavior of Manufacturer B in upstream direction.

The blue line represents the entire captured UDP traffic send by all field device emulators at ~768 … 3.072 packets/s for both Manufacturers, while the brown line shows the UDP traffic from one field device emulator at 32 … 128 packets/s for Manufacturer A and 192 … 768 packets/s for Manufacturer B.

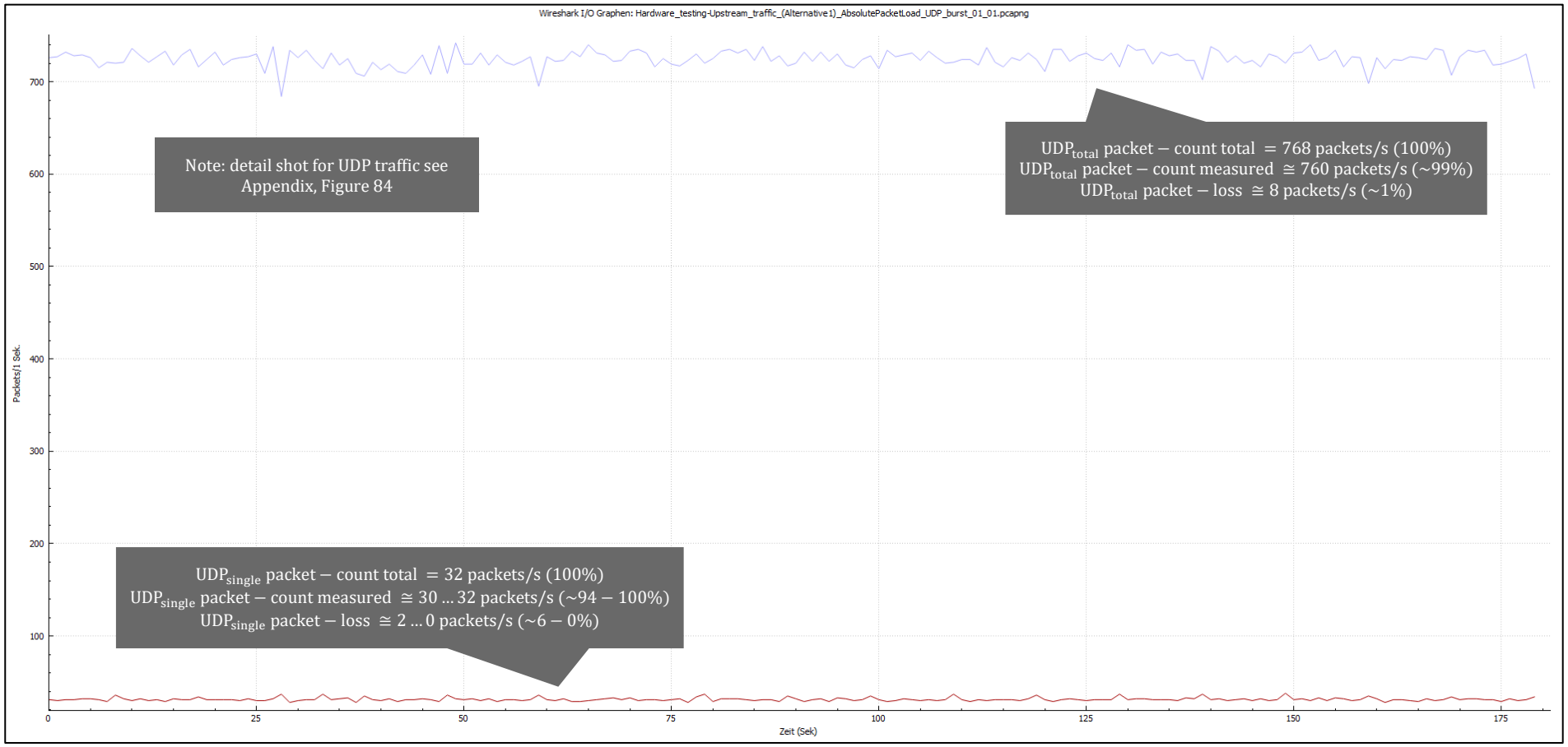

Figure 83: Upstream traffic analysis **@ decreasing UDP cycle time (Alternative 1), Manufacturer A** - Measurement results (1·24 UDP packets / 31,25 ms) - total packet count

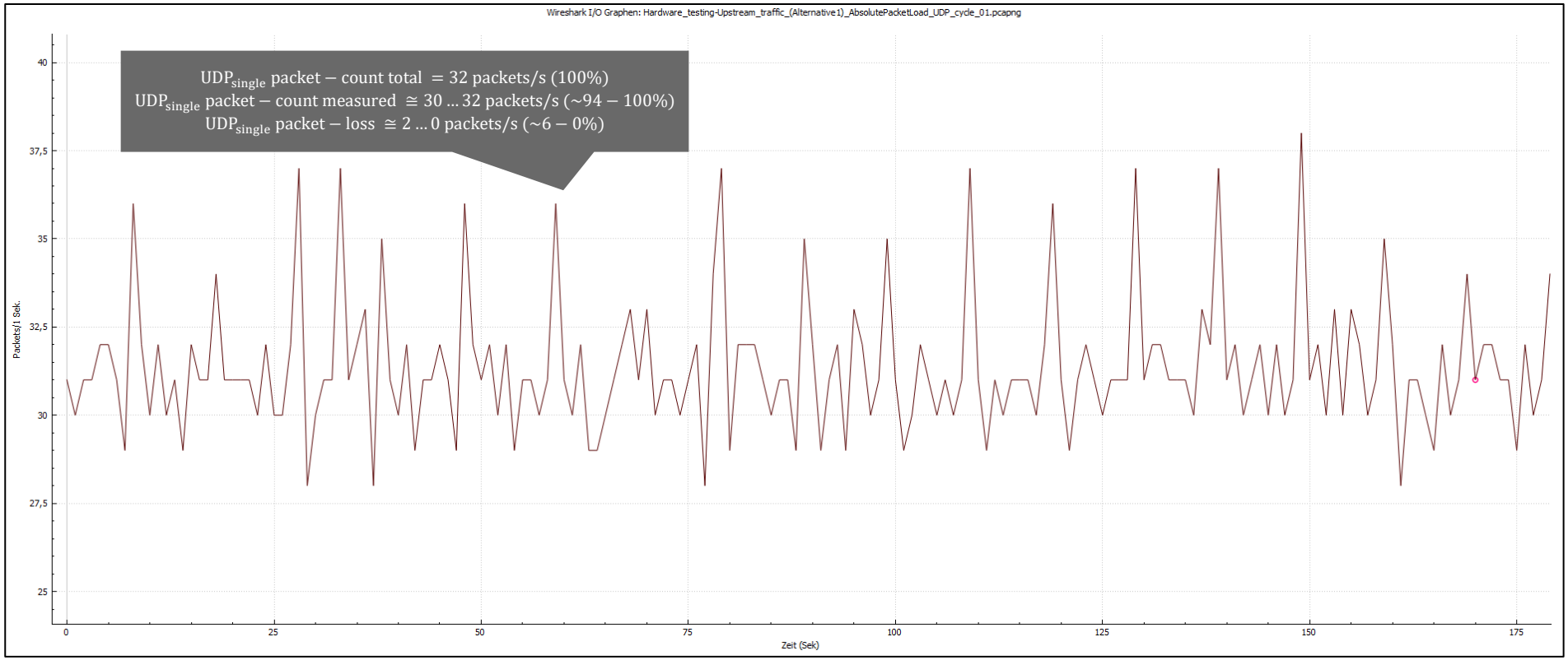

Figure 84: Upstream traffic analysis **@ decreasing UDP cycle time (Alternative 1), Manufacturer A** - Measurement results (1·24 UDP packets / 31,25 ms) – single source packet count

Figure 83 shows that UDP traffic has a packet loss of 1% at a total PCS (‘*Packet Count per Second*’) of $768\ \text{packets/s}$. Said packet loss of UDP packets happens due to the same phenomenon which was already described in the measurement summary of chapter A.4.1. Hence, the packet prioritization of the APL switch works correctly.

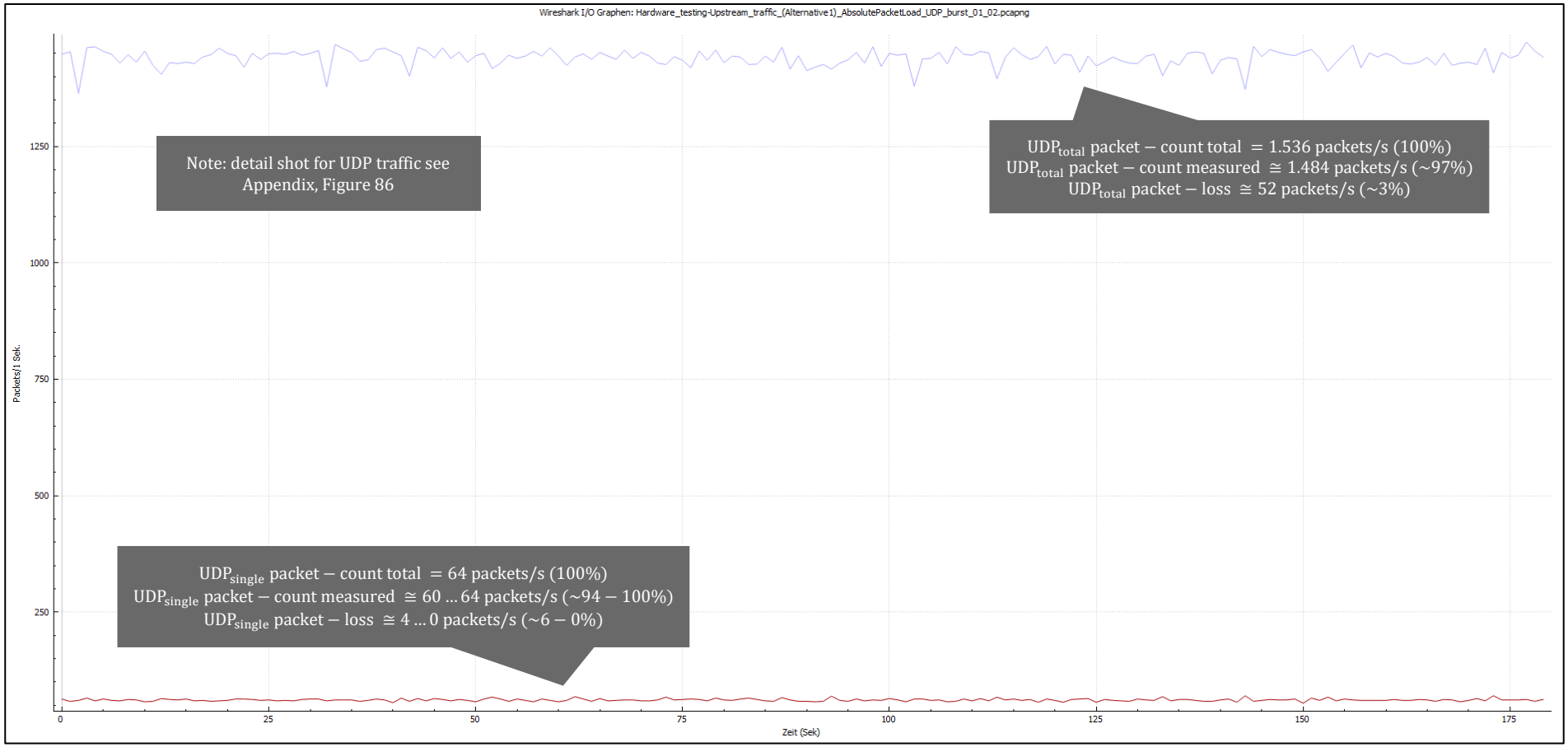

Figure 85: Upstream traffic analysis **@ decreasing UDP cycle time (Alternative 1), Manufacturer A** - Measurement results (1·24 UDP packets / 15,625 ms) - total packet count

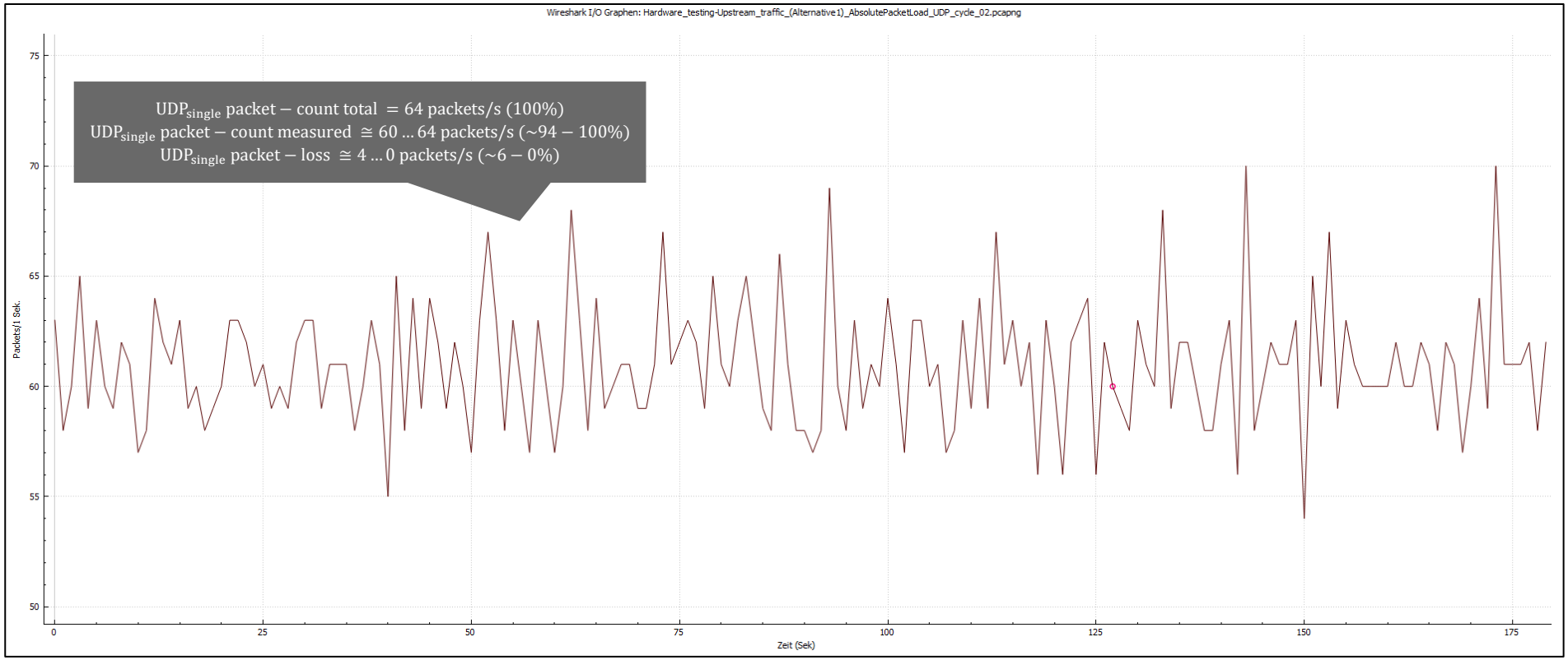

Figure 86: Upstream traffic analysis **@ decreasing UDP cycle time (Alternative 1), Manufacturer A** - Measurement results (1·24 UDP packets / 15,625 ms) – single source packet count

Figure 85 shows that by doubling the total PCS ('*Packet Count per Second*') of UDP traffic from ~768 packets/s to ~1.536 packets/s, packet loss increases from 1% to 3%. The same phenomenon which was referred to in the previous Figure 83 is also apparent in Figure 85, but in a more pronounced form. Hence, the packet prioritization of the APL switch works correctly.

Figure 87: Upstream traffic analysis **@ decreasing UDP cycle time (Alternative 1), Manufacturer A** - Measurement results (**1·24** UDP packets **/ 7,8125 ms**) - total packet count

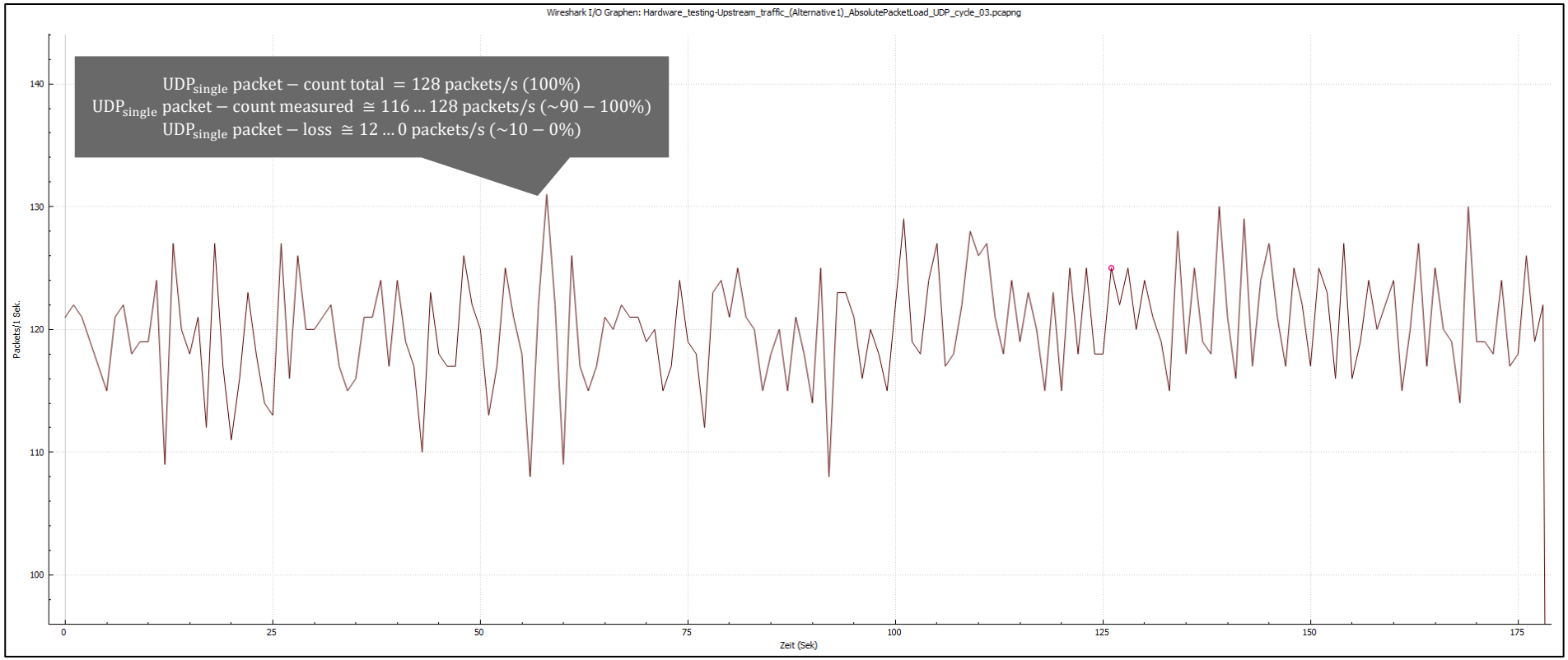

Figure 88: Upstream traffic analysis **@ decreasing UDP cycle time (Alternative 1), Manufacturer A** - Measurement results (**1·24** UDP packets **/ 7,8125 ms**) – single source packet count

Figure 87 shows that by quadrupling the total PCS ('*Packet Count per Second*') of UDP traffic from ~768 packets/s to ~3.072 packets/s, packet loss increases from 3% to 6%. The same phenomenon which was referred to in the previous Figure 83 is also apparent in Figure 87, but in a more pronounced form. Hence, the packet prioritization of the APL switch works correctly.

Summary (Figure 83 to Figure 88): The measurements show, that stable packet-throughput of simultaneous incoming traffic, by gradual increase of UDP traffic up to ~3.072 packets/s, is always ensured. The UDP packet loss happens due to missing synchronization of field device, generating simultaneous packet load and is not related to the APL switch.

Due to UDP traffic's total PCS ('*Packet Count per Cycle*') staying below the internal total '*bufferLength*' limit (1800 packets/cycle), packet loss does not occur, regarding packet count (see chapter 5.2.1). Furthermore, the total PPT needed for emptying all packet ques once per cycle (≅ 1,74 ms), stays below the decreased UDP cycle time (7,8125 ms), thus preventing packet loss does due to exceeding the '*queueLength*' limit (128 packets/cycle) caused by packet overlapping inside the respective packet queues.

In conclusion, the APL switch of Manufacturer A fulfills its packet-throughput requirements according to Table 16 regarding the desired packet processing behavior.

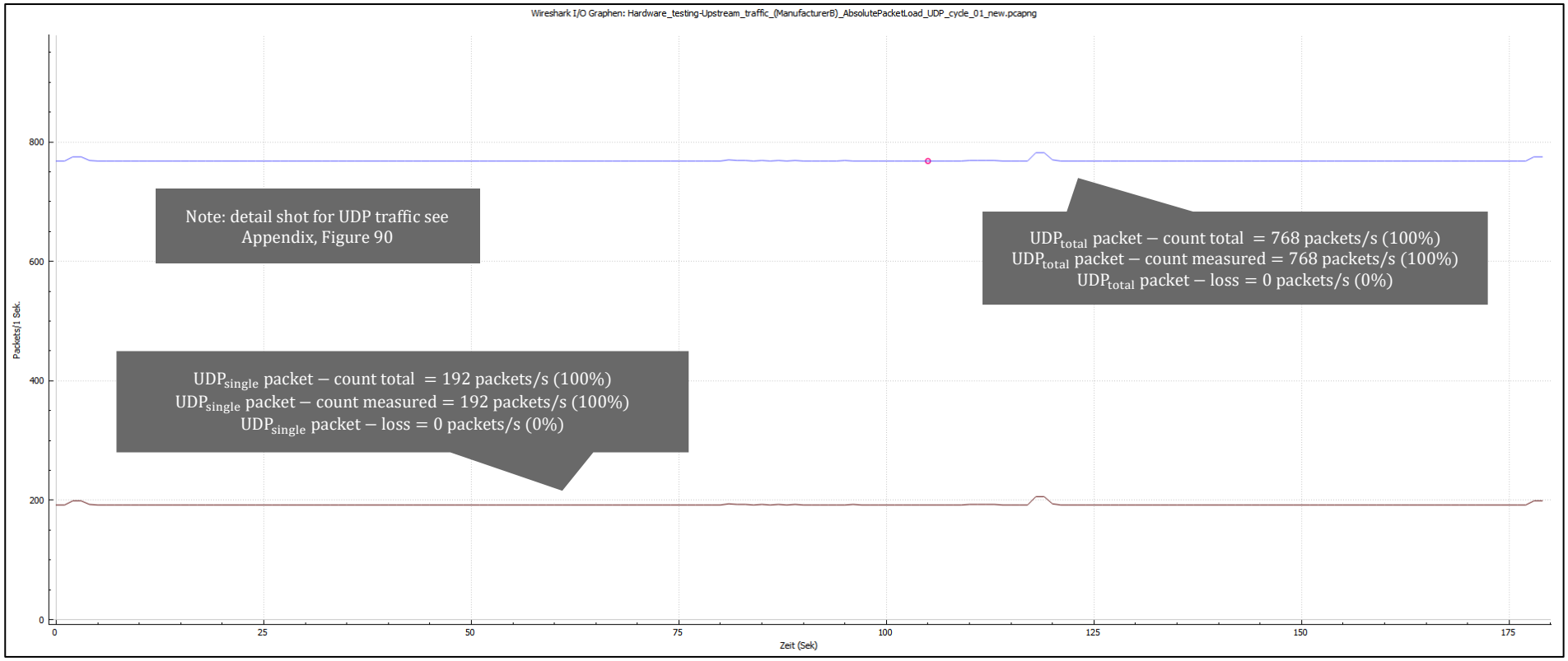

Figure 89: Upstream traffic analysis **@ decreasing UDP cycle time (Alternative 1), Manufacturer B** - Measurement results (6·4 UDP packets / 31,25 ms) - total packet count

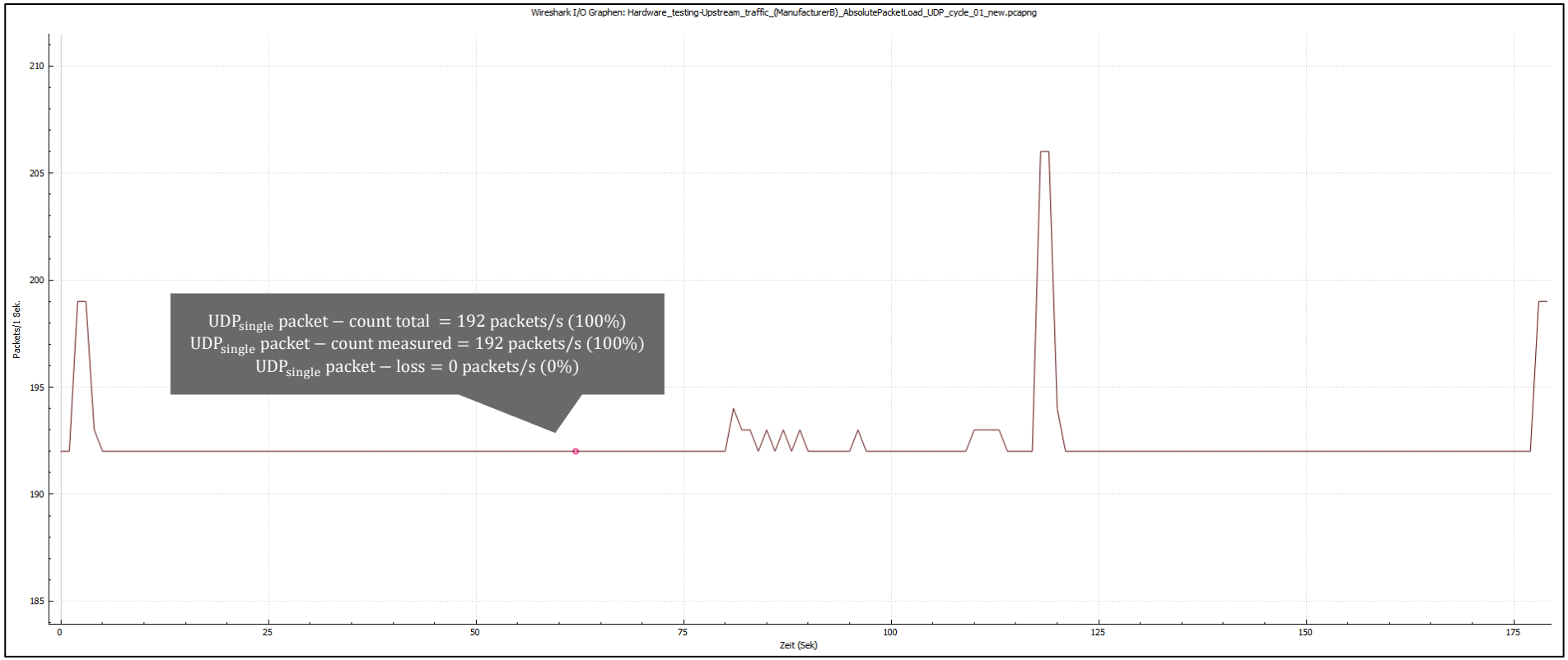

Figure 90: Upstream traffic analysis **@ decreasing UDP cycle time (Alternative 1), Manufacturer B** - Measurement results (6·4 UDP packets / 31,25 ms) – single source packet count

Figure 90 shows that UDP traffic does not struggle with packet loss at a total PCS (‘*Packet Count per Second*’) of $768\ \text{packets/s}$. Hence, all UDP packets are forwarded successfully by the APL switch.

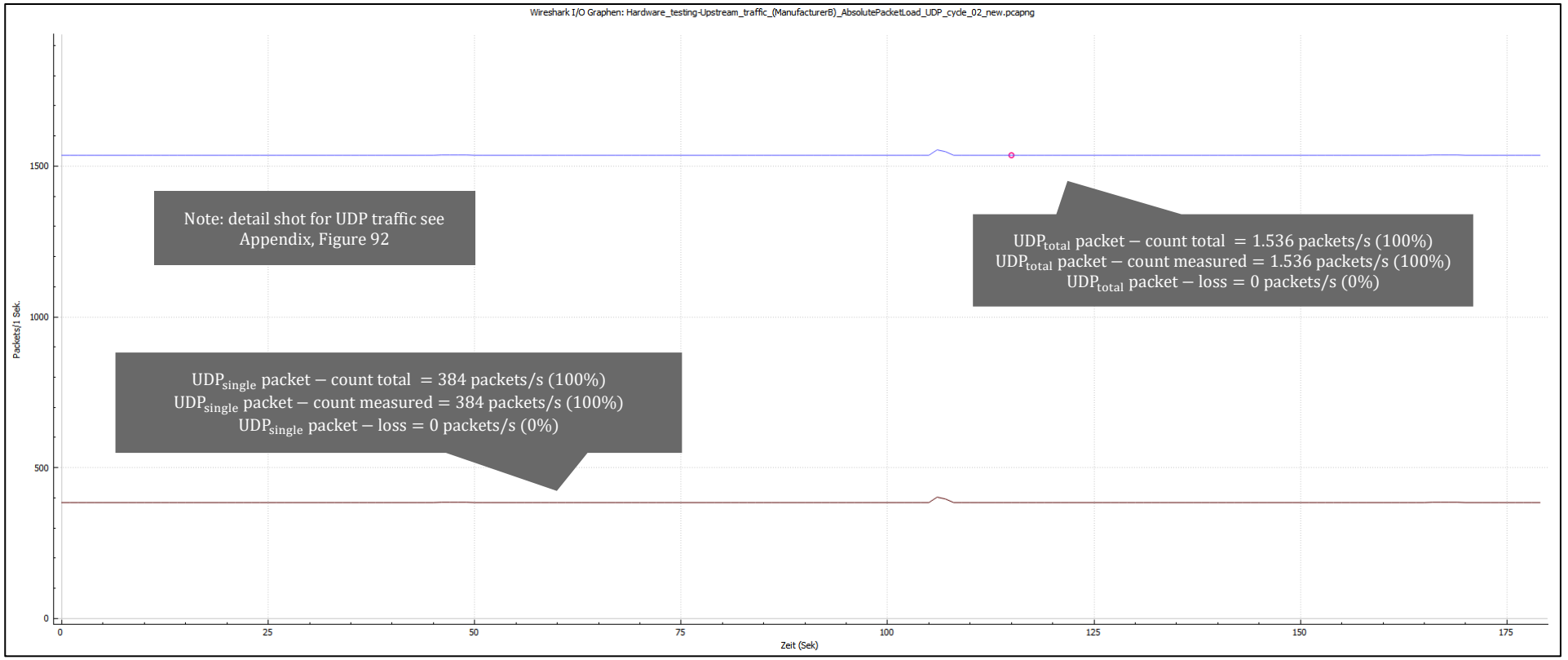

Figure 91: Upstream traffic analysis **@ decreasing UDP cycle time (Alternative 1), Manufacturer B** - Measurement results (6·4 UDP packets / 15,625 ms) - total packet count

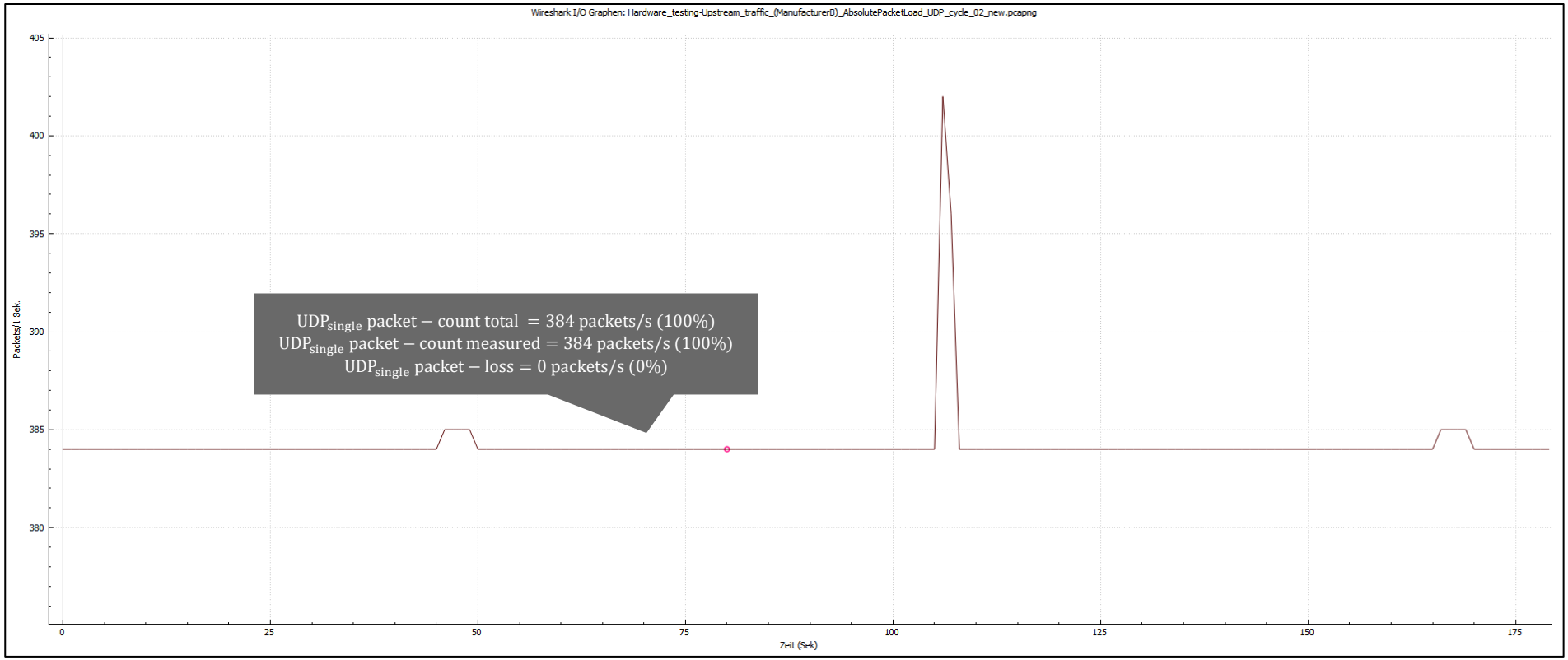

Figure 92: Upstream traffic analysis **@ decreasing UDP cycle time (Alternative 1), Manufacturer B** - Measurement results (6·4 UDP packets / 15,625 ms) – single source packet count

Figure 91 shows that by doubling the total PCS (‘*Packet Count per Second*’) of UDP traffic from ~ 768 packets/s to ~ 1.536 packets/s, packet loss still does not occur. Hence, all packets are forwarded successfully by the APL switch.

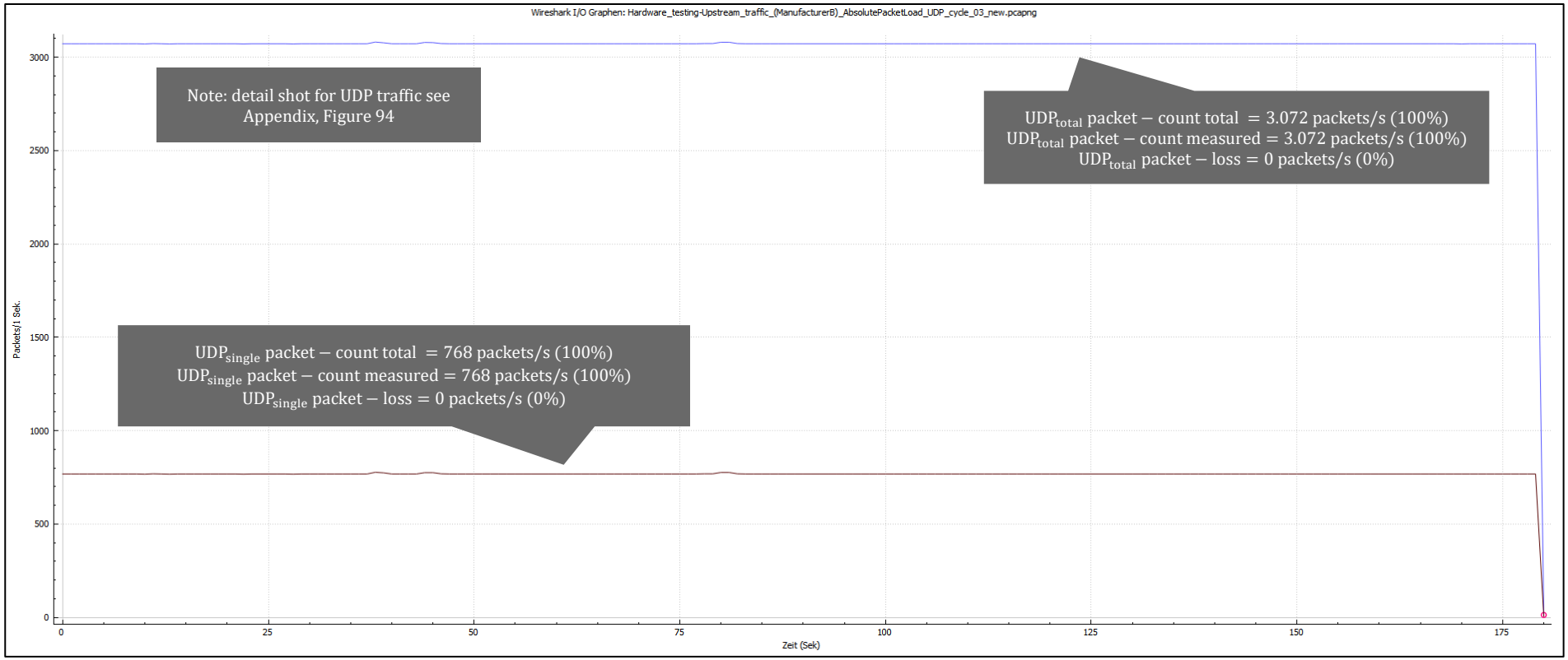

Figure 93: Upstream traffic analysis **@ decreasing UDP cycle time (Alternative 1), Manufacturer B** - Measurement results (6·4 UDP packets / 7,8125 ms) - total packet count

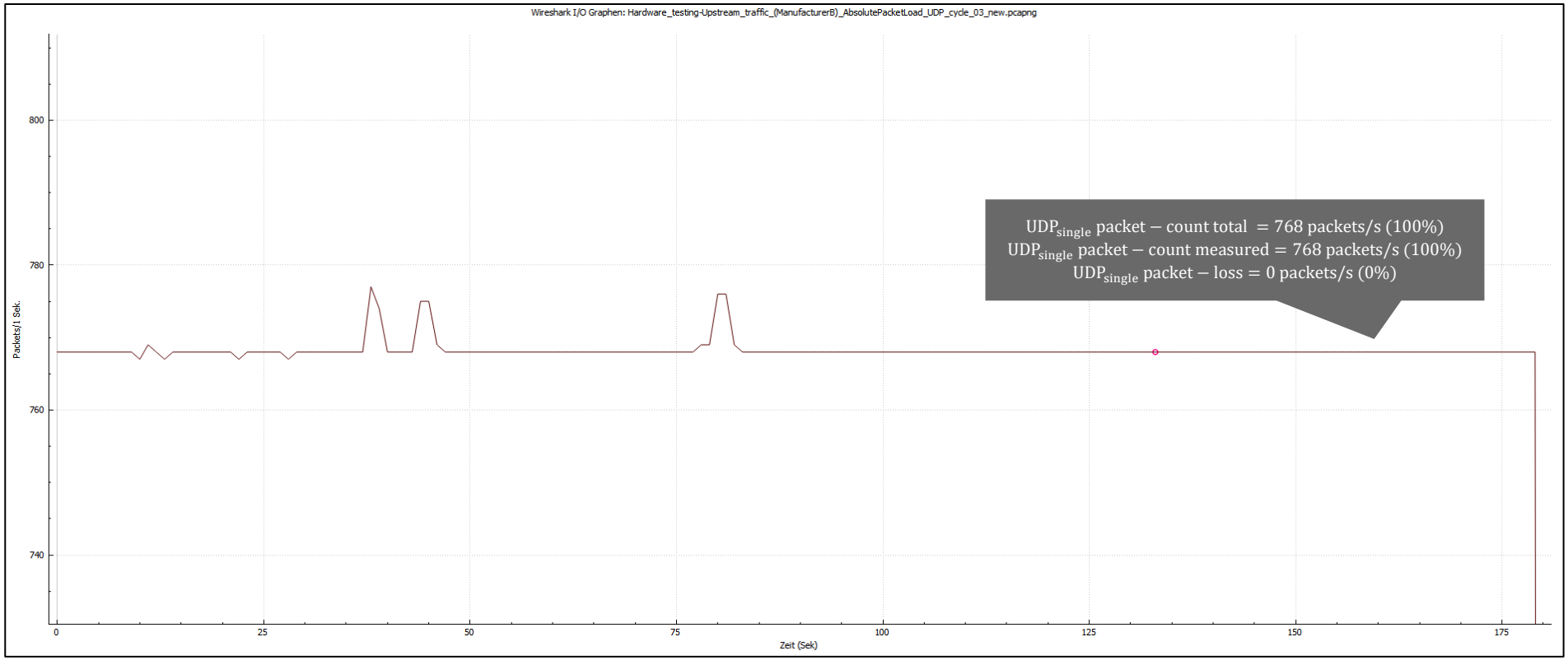

Figure 94: Upstream traffic analysis **@ decreasing UDP cycle time (Alternative 1), Manufacturer B** - Measurement results (6·4 UDP packets / 7,8125 ms) – single source packet count

Figure 93 shows that by quadrupling the total PCS ('*Packet Count per Second*') of UDP traffic from ~ 768 packets/s to ~ 3.072 packets/s, packet loss still does not occur. Hence, all packets are forwarded successfully by the APL switch.

Summary: (Figure 89 to Figure 94): The measurements show, that stable packet-throughput of simultaneous incoming traffic, by gradual increase of UDP traffic up to ~ 3.072 packets/s, is always ensured. The total packet count of UDP traffic 204 packets/cycle, stated in Table 16. This value is below the '*bufferCount*' limit of the APL switch (1.024 packets/cycle). Therefore, no packet discarding occurs in the packet buffer of the APL switch (see chapter 5.2.1). Furthermore, the total PPT needed for emptying all packet ques once per cycle ($\cong 1{,}74$ ms), stays below the decreased UDP cycle time ($7{,}8125$ ms), thus preventing packet loss does due to exceeding the '*bufferCount*' limit caused by packet overlapping inside the packet buffer. In conclusion, the APL switch of Manufacturer B fulfills its packet-throughput requirements according to Table 16 regarding the desired packet processing behavior.

A.4.2.2 Packet processing @ increasing UDP packet count:

The following tests have been conducted by increasing UDP traffic through increasing its packet count. The traffic parameters used in these tests are stated in Table 17.

Table 17: Downstream traffic analysis (Alternative 1), Manufacturers A&B – Traffic parameters, increasing UDP traffic @ varying *Packet Count per Cycle*

	UDP real-time data, *single sensor*					
user priority	6					
total packet payload **(TPP)**	46 Byte (data) + 42 Byte (framing/transmission) = 88 Byte					
packet cycle time **(PCT)**	62,5 ms					
Packet Count per Cycle **(PCC)**	**24 · 32 / 24 · 64 / 24 · 128 packets/cycle (Manufacturer A)**			**4 · 128 / 4 · 256 / 4 · 512 packets/cycle) (Manufacturer B, 100 Mbit/s)**		
Packet data Payload per Cycle **(PPC)**	~34,5 / ~69,0 / ~138,0 kByte/cycle (Manufacturer A)			~23,0 / ~46,0 / ~92,0 kByte/cycle (Manufacturer B)		
total frame payload per cycle **(FPC)**	~68 / ~135 / ~270 kByte/cycle (Manufacturer A)			~45 / ~90 / ~180 kByte/cycle (Manufacturer B)		
Packet Count per Second **(PCS)**	**12.288 / packets/s (Manufacturer A)**	**24.576 packets/s (Manufacturer A)**	**49.152 packets/s (Manufacturer A)**	**8.192 / packets/s (Manufacturer B)**	**16.384 / packets/s (Manufacturer B)**	**32.768 / packets/s (Manufacturer B)**
packet data payload per second **(PPS)**	12.288 p/s · 46 Byte = 565.248 Byte/s (~**552, 0 kByte/s**)	24.576 p/s · 46 Byte = 1.130.496 Byte/s (~**1, 08 MByte/s**)	49.152 p/s · 46 Byte = 2.260.992 Byte/s (~**2, 16 MByte/s**)	8.192 p/s · 46 Byte = 376.832 Byte/s (~**368, 0 kByte/s**)	16.384 p/s · 46 Byte = 753.664 Byte/s (~**736, 0 kByte/s**)	32.768 p/s · 46 Byte = 1.507.328 Byte/s (~**1, 44 MByte/s**)
total frame payload per second **(FPS)**	12.288 p/s · 88 Byte = 1.081.344 Byte/s (~**1, 03 MByte/s**)	24.576 p/s · 88 Byte = 2.162.688 Byte/s (~**2, 06 MByte/s**)	49.152 p/s · 88 Byte = 4.325.376 Byte/s (~**4, 13 MByte/s**)	8.192 p/s · 88 Byte = 720.896 Byte/s (~**704, 0 kByte/s**)	16.384 p/s · 88 Byte = 1.441.792 Byte/s (~**1, 38 MByte/s**)	32.768 p/s · 88 Byte = 2.883.584 Byte/s (~**2, 75 MByte/s**)

The following figures show the packet-throughput behavior of the switch when using the stated in Table 17. Figure 95 to Figure 100 show the packet-throughput behavior of Manufacturer A in upstream direction. Figure 101 to Figure 106 show the packet-throughput behavior of Manufacturer B in upstream direction.

The blue line represents the entire captured UDP traffic send by all field device emulators at ~12.288 … 49.152 packets/s for Manufacturer A and ~8.192 … 32.768 packets/s for Manufacturer B. The brown line shows the UDP traffic from one field device emulator at 512 … 2.048 packets/s for Manufacturer A and 2.048 … 8.192 p/s for Manufacturer B.

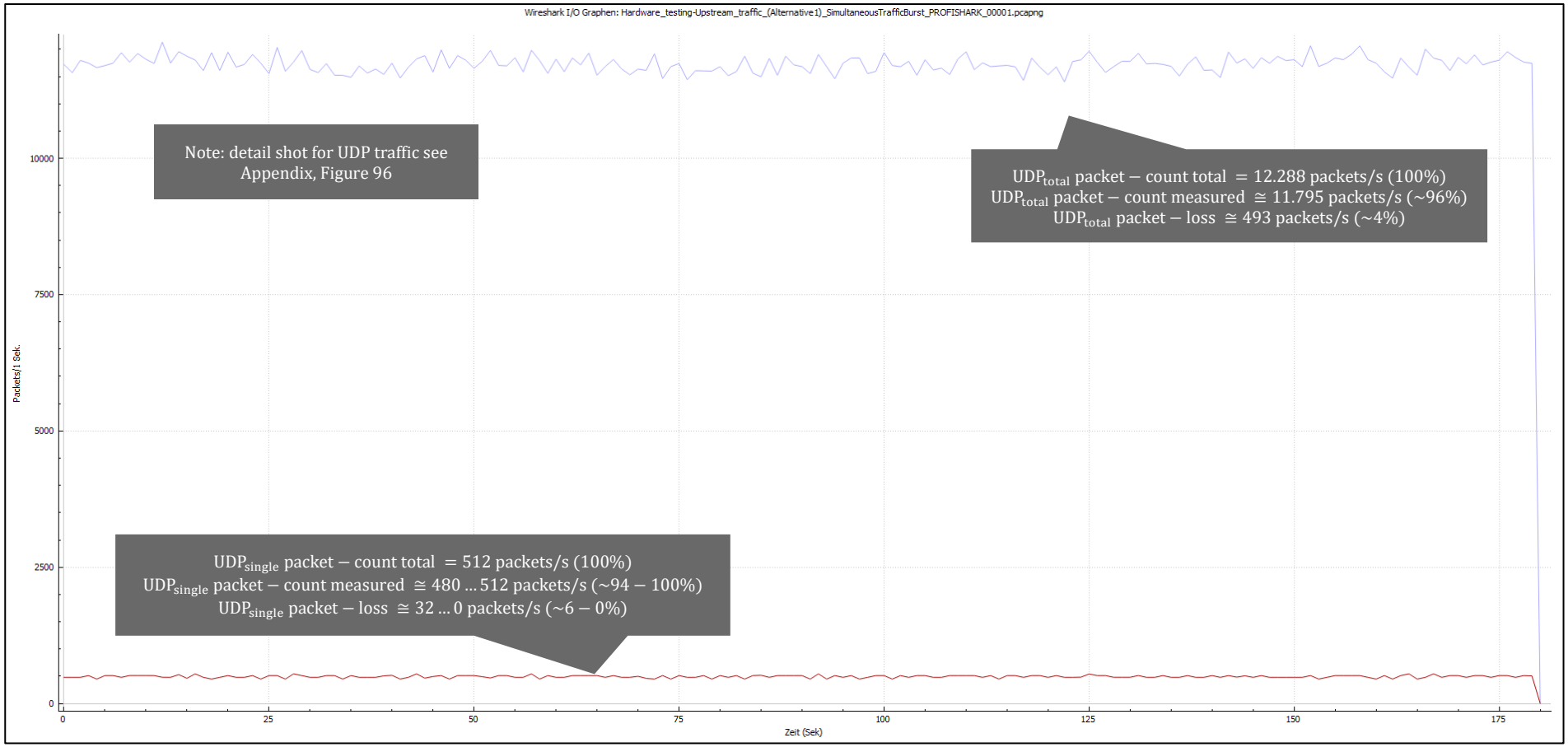

Figure 95: Upstream traffic analysis **@ increasing UDP packet count (Alternative 1), Manufacturer A** - Measurement results (24·32 UDP packets / 62,5 ms) - total packet count

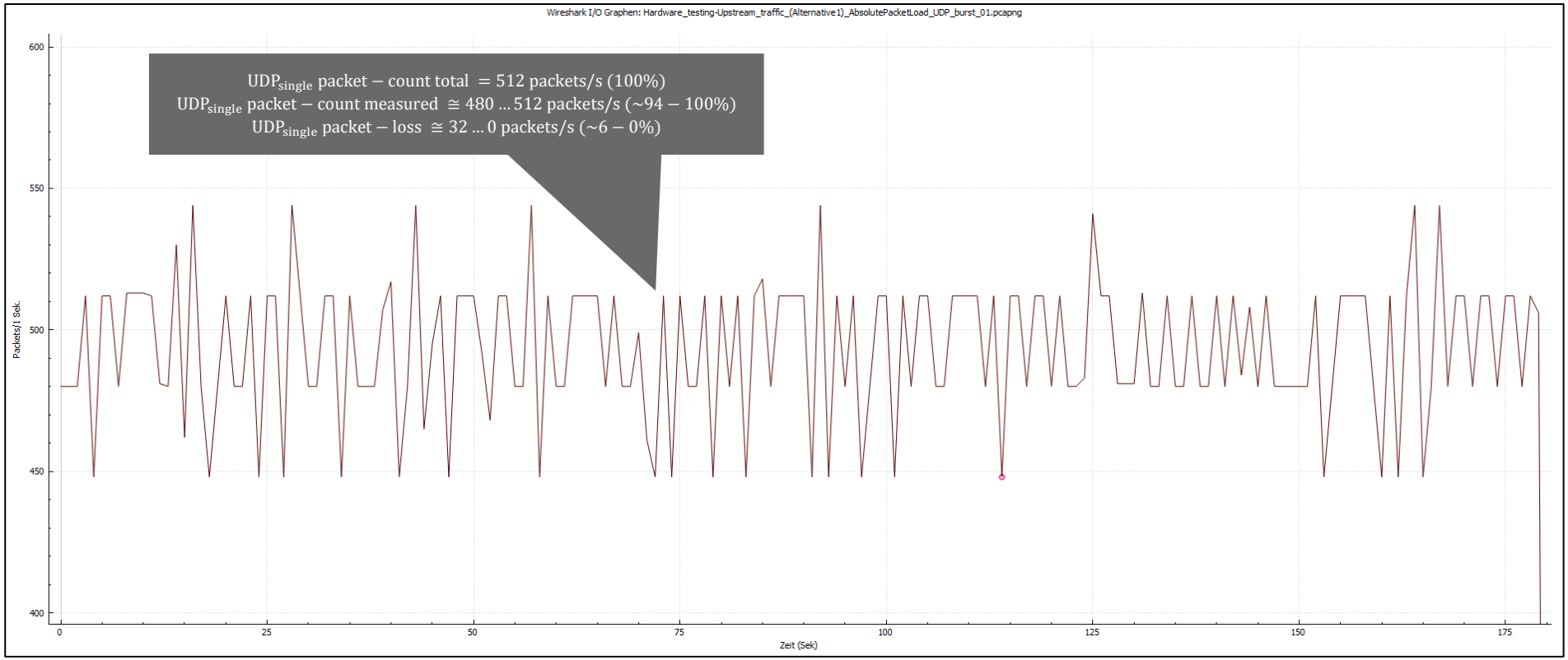

Figure 96: Upstream traffic analysis **@ increasing UDP packet count (Alternative 1), Manufacturer A** - Measurement results (24·32 UDP packets / 62,5 ms) – single source packet count

Figure 95 shows that UDP traffic has a packet loss of 4% at a total PCS ('*Packet Count per Second*') of $12.288\ \text{packets/s}$. Said packet loss of UDP packets happens due to the same phenomenon which was already described in the measurement summary of chapter A.4.1. Hence, the packet prioritization of the APL switch works correctly.

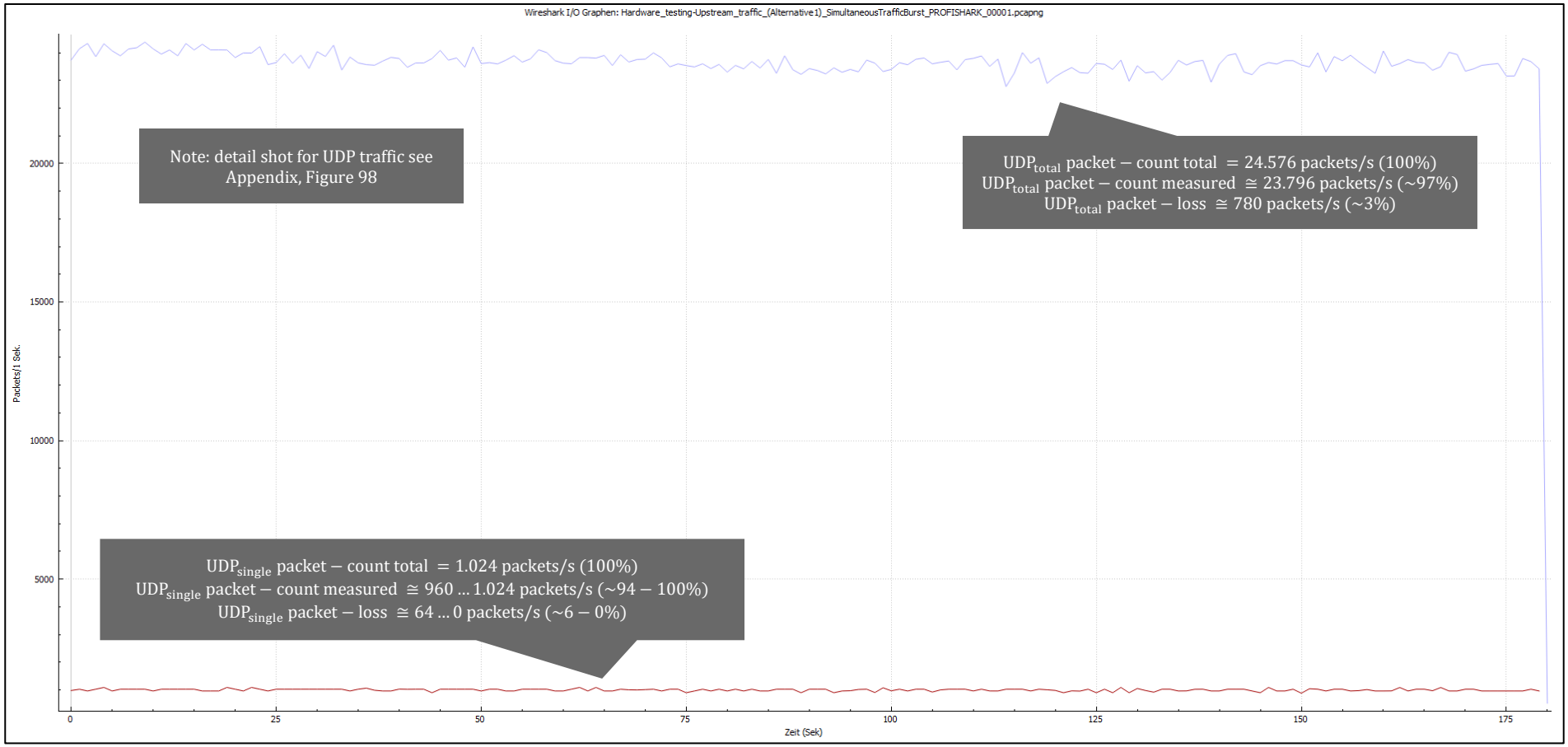

Figure 97: Upstream traffic analysis **@ increasing UDP packet count (Alternative 1), Manufacturer A** - Measurement results (24·64UDP packets / 62,5 ms) – total packet count

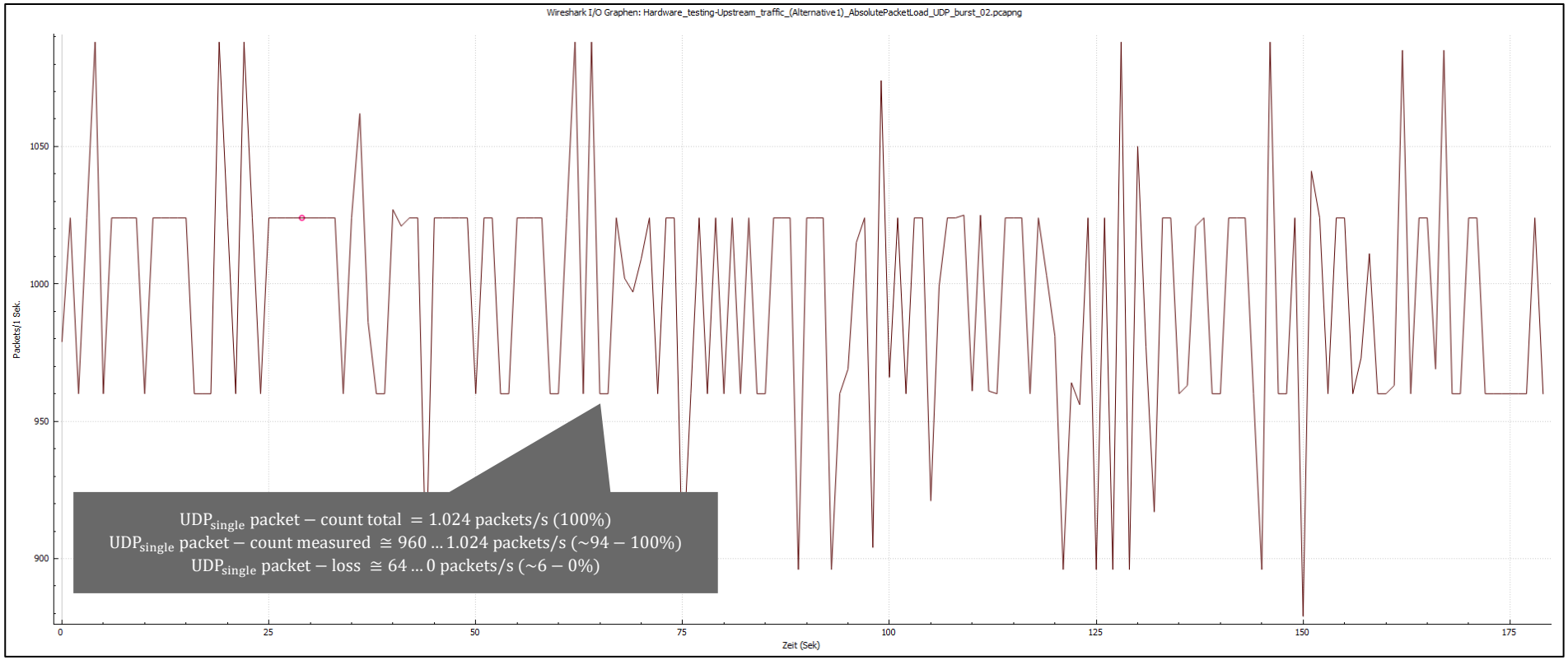

Figure 98: Upstream traffic analysis **@ increasing UDP packet count (Alternative 1), Manufacturer A** - Measurement results (24·64UDP packets / 62,5 ms) – single source packet count

Figure 97 shows that by doubling the total PCS ('*Packet Count per Second*') of UDP traffic from ~12.288 packets/s to ~24.576 packets/s, packet loss increases from 1% to 3%. The same phenomenon which was referred to in the previous Figure 95 is also apparent in Figure 97, but in a more pronounced form. Hence, the packet prioritization of the APL switch works correctly.

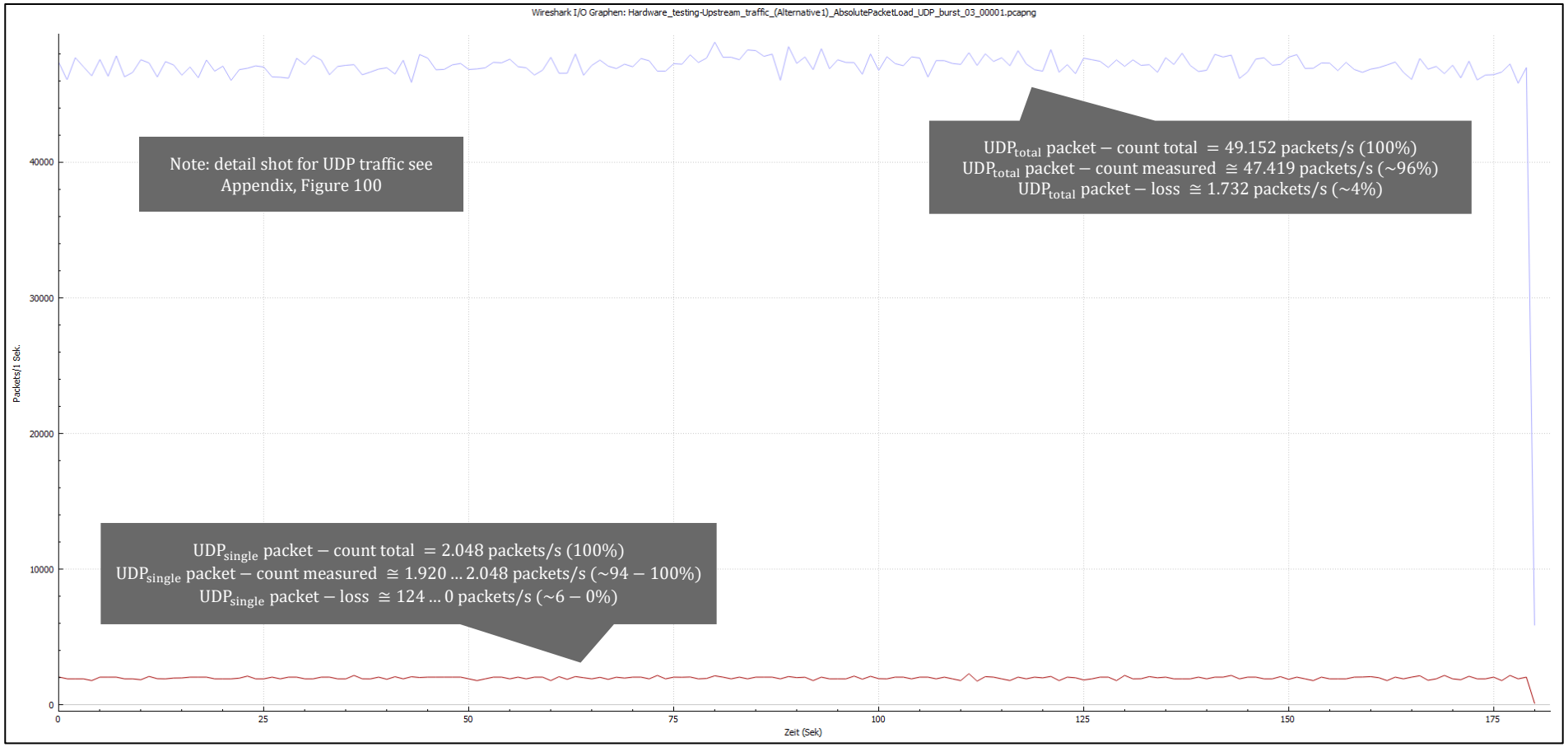

Figure 99: Upstream traffic analysis **@ increasing UDP packet count (Alternative 1), Manufacturer A** - Measurement results (24·128UDP packets / 62,5 ms) – total packet count

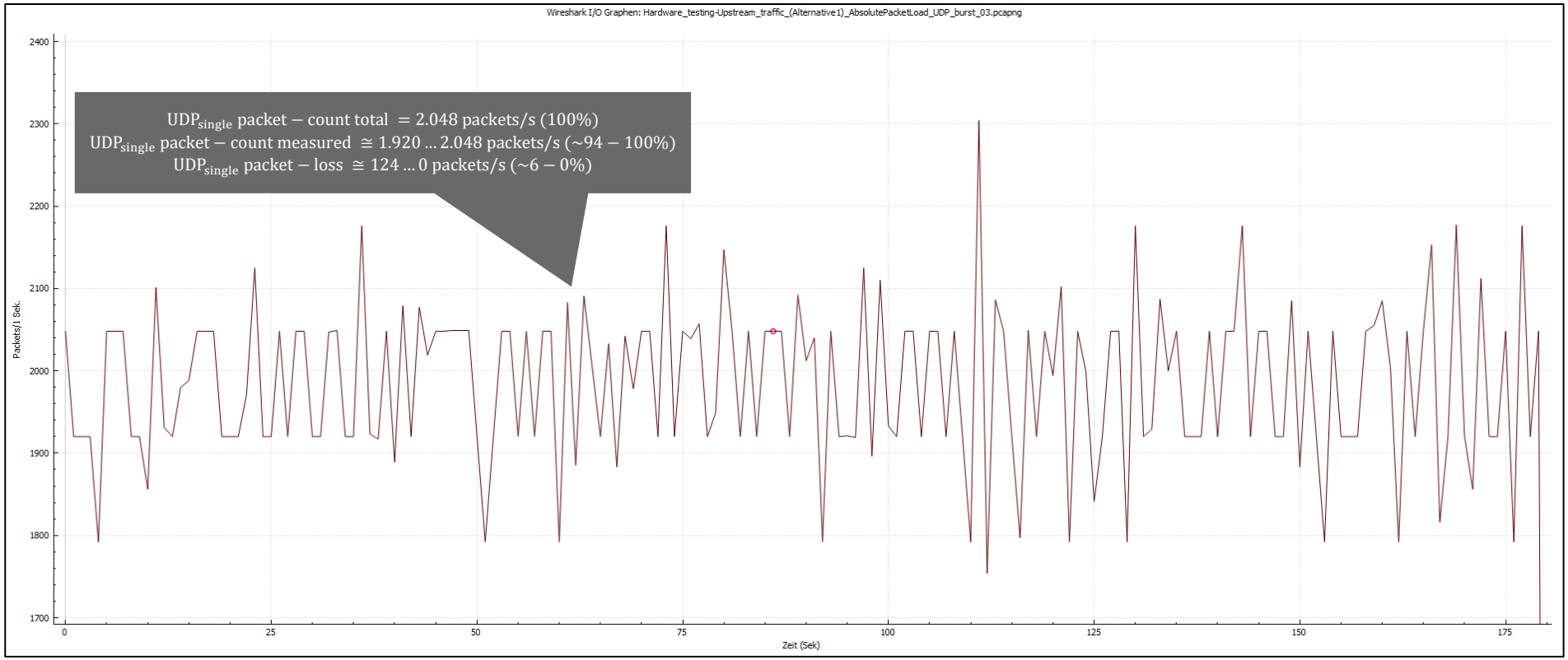

Figure 100: Upstream traffic analysis **@ increasing UDP packet count (Alternative 1), Manufacturer A** - Measurement results (24·128UDP packets / 62,5 ms) – single source packet count

Figure 99 shows that by quadrupling the total PCS ('*Packet Count per Second*') of UDP traffic from ~12.288 packets/s to ~49.152 packets/s, packet loss increases from 3% to 6%. The same phenomenon which was referred to in the previous Figure 95 carries over Figure 99, but in a more pronounced form. Hence, the packet prioritization of the APL switch works correctly.

Summary (Figure 95 to Figure 100): The measurements show, that stable packet-throughput of simultaneous incoming traffic, by gradual increase of UDP traffic up to ~49.152 packets/s, is always ensured. The UDP packet loss happens due to missing synchronization of field device, generating simultaneous packet load and is not related to the APL switch (see measurement summary chapter of A.4.1).

Due to UDP traffic's PCS ('*Packet Count per Cycle*') staying below the internal total '*bufferLength*' limit (1800 packets/cycle), as well as the '*queueLength*' limit (128 packets/cycle), packet loss does not occur, regarding packet count (see chapter 5.1.2 and 5.1.2). Furthermore, the total PPT needed for emptying all packet queues once per cycle (3.072 packets/cycle), stays below total packet count processable by the switch hardware (3.402 packets/cycle) at a cycle time of 62,5 ms, according to formula (58) (see measurement summary of chapter A.4.1).

In conclusion, the APL switch of Manufacturer A fulfills its packet-throughput requirements according to Table 17 regarding the desired packet processing behavior.

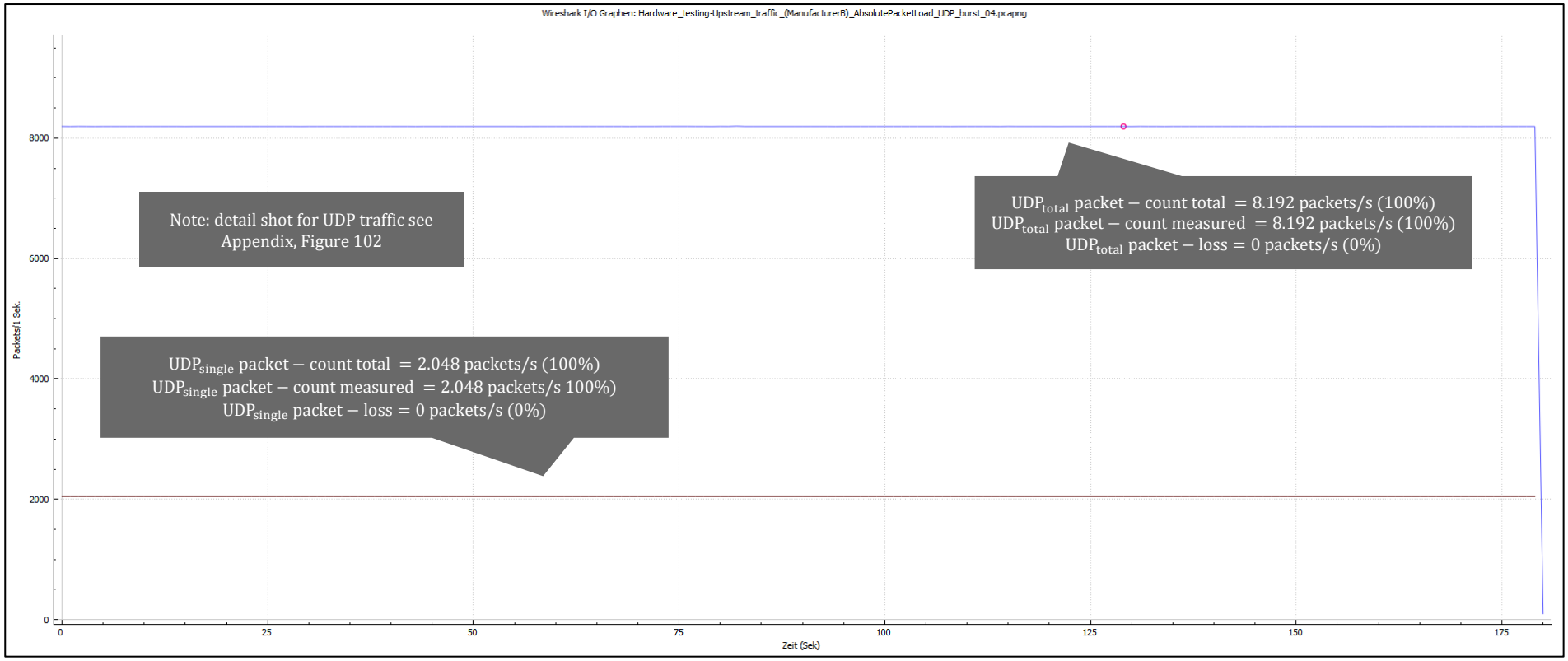

Figure 101: Upstream traffic analysis **@ increasing UDP packet count (Alternative 1), Manufacturer B** - Measurement results (4·128 UDP packets / 62,5 ms) - total packet count

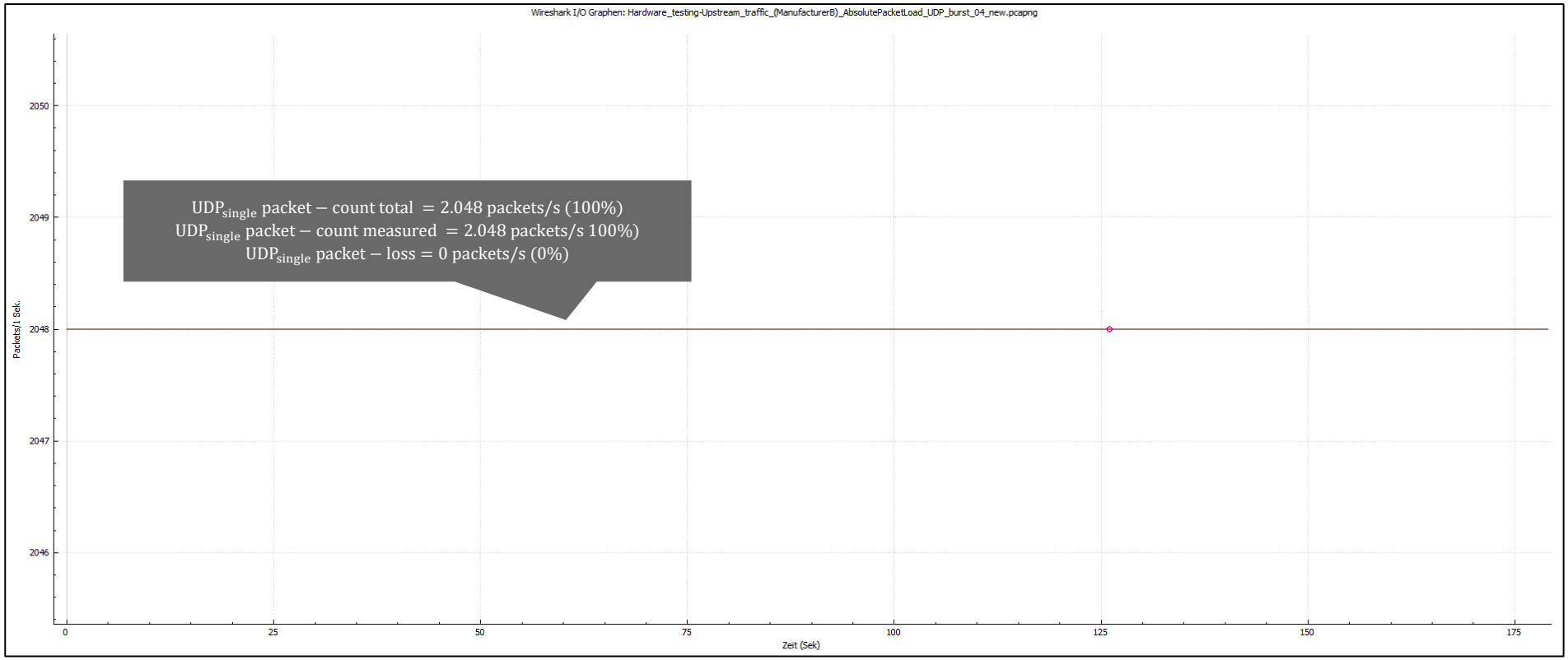

Figure 102: Upstream traffic analysis **@ increasing UDP packet count (Alternative 1), Manufacturer B** - Measurement results (4·128 UDP packets / 62,5 ms) – single source packet count

Figure 101 shows that UDP traffic does not struggle with packet loss at a total PCS (‘*Packet Count per Second*’) of 8.192 packets/s . Hence, all UDP packets are forwarded successfully by the APL switch.

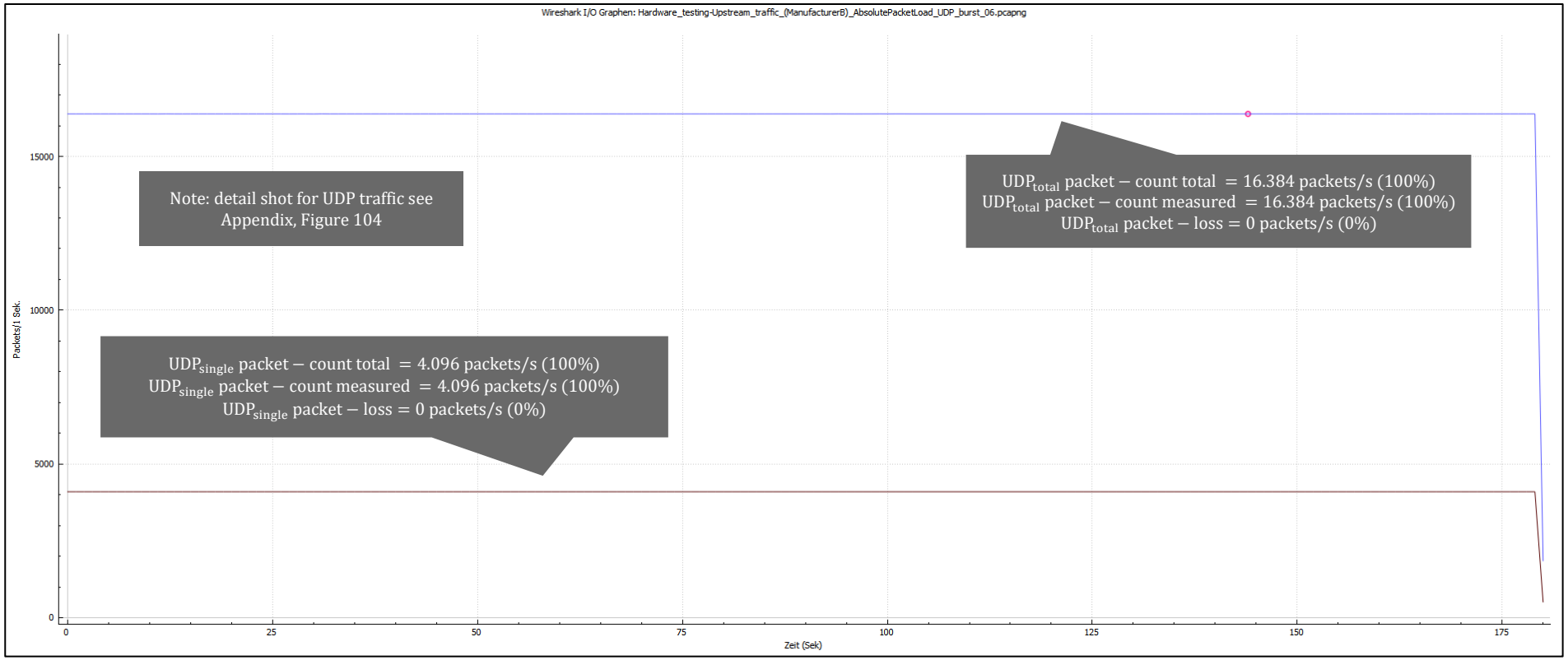

Figure 103: Upstream traffic analysis **@ increasing UDP packet count (Alternative 1), Manufacturer B** - Measurement results (4x256 UDP packets / 62,5 ms) – total packet count

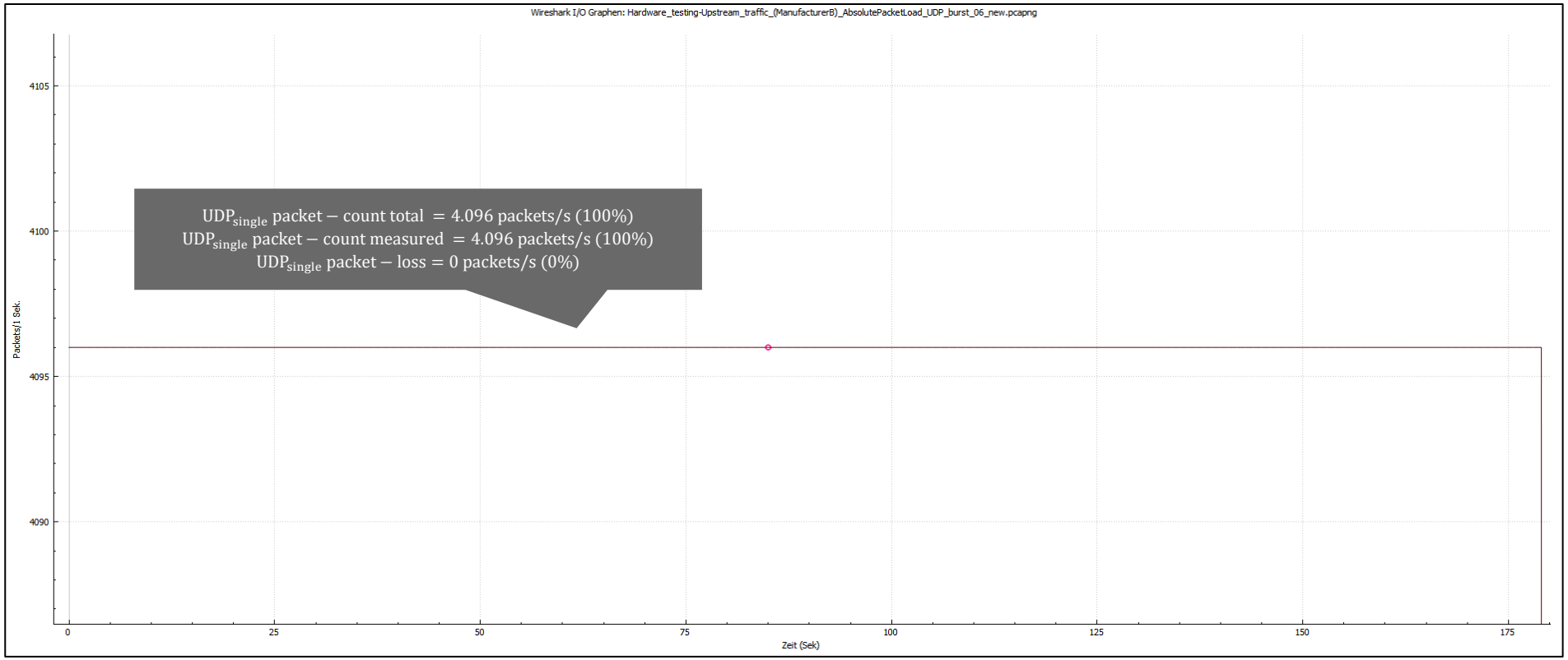

Figure 104: Upstream traffic analysis **@ increasing UDP packet count (Alternative 1), Manufacturer B** - Measurement results (4x256 UDP packets / 62,5 ms) – single source packet count

Figure 103 shows that by doubling the total PCS ('*Packet Count per Second*') of UDP traffic from ~8.192 packets/s to ~16.384 packets/s, packet loss still does not occur. Hence, all packets are forwarded successfully by the APL switch.

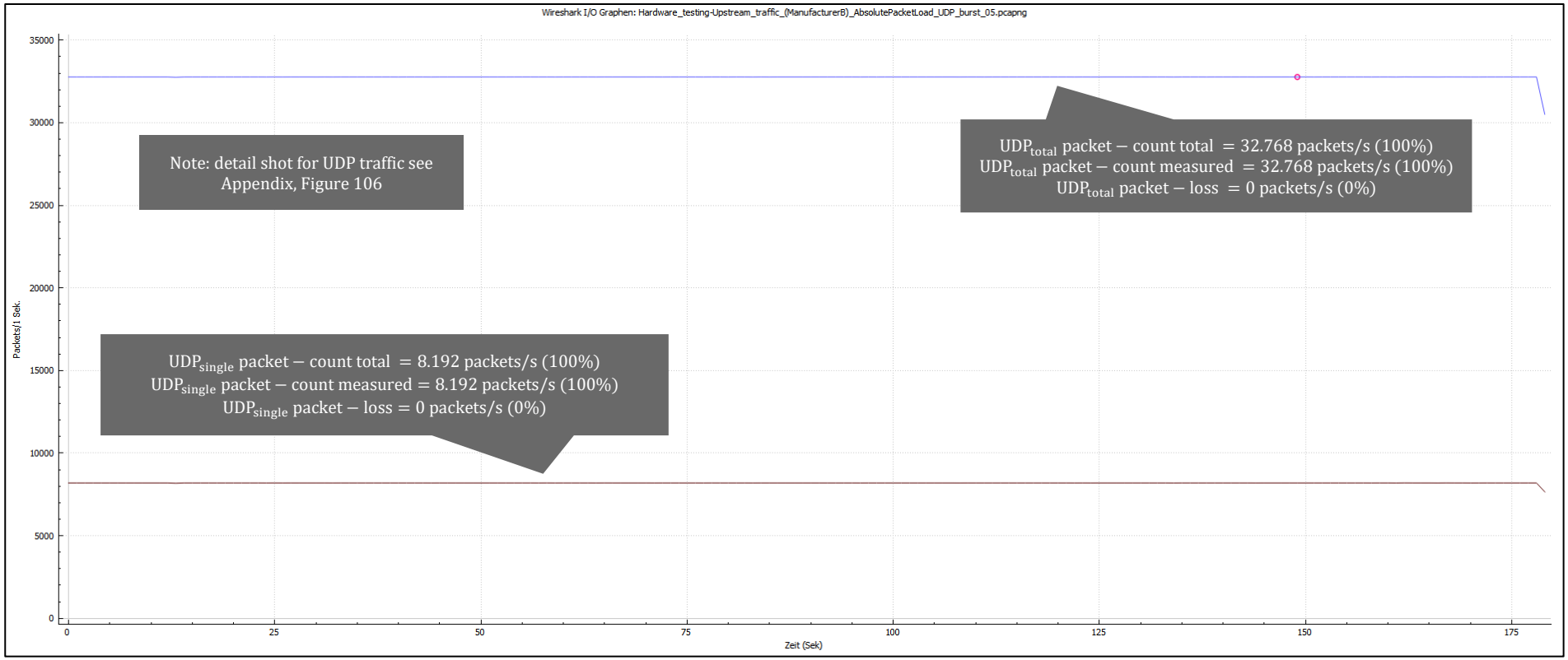

Figure 105: Upstream traffic analysis **@ increasing UDP packet count (Alternative 1), Manufacturer B** - Measurement results (4x512 UDP packets / 62,5 ms) – total packet count

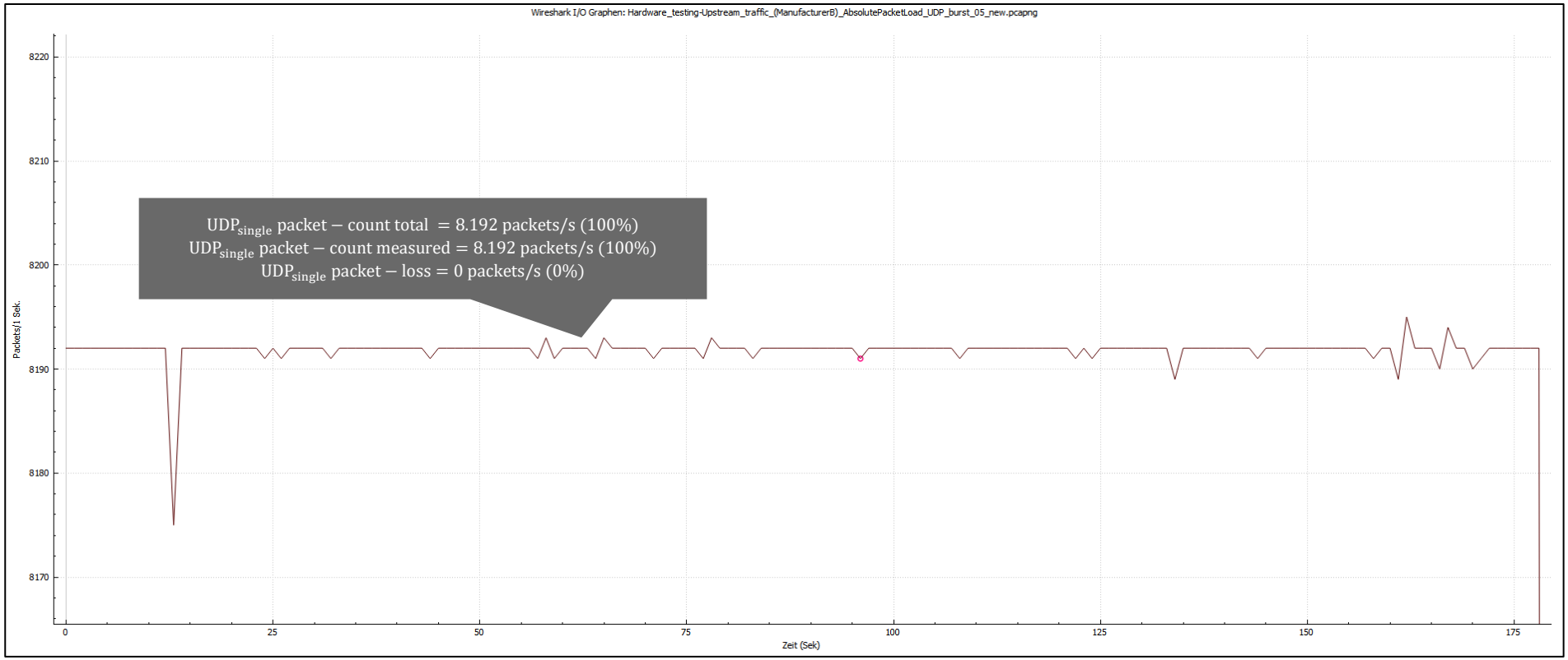

Figure 106: Upstream traffic analysis **@ increasing UDP packet count (Alternative 1), Manufacturer B** - Measurement results (4x512 UDP packets / 62,5 ms) – single source packet count

Figure 105 shows that by quadrupling the total PCS ('*Packet Count per Second*') of UDP traffic from ~ 8.192 packets/s to ~ 32.768 packets/s, packet loss still does not occur. Hence, all packets are forwarded successfully by the APL switch.

Summary: (Figure 101 to Figure 106): The measurements show, that stable packet-throughput of simultaneous incoming traffic, by gradual increase of UDP traffic up to ~ 32.768 packets/s, is always ensured. The total packet count of UDP traffic is 1.024 packets/cycle, stated in Table 17. This value is within the '*bufferCount*' limit of the APL switch (1.024 packets/cycle). Therefore, no packet discarding occurs in the packet buffer of the APL switch (see chapter 5.2.1). Furthermore, the total PPT needed for emptying all packet queues once per cycle (2.048 packets/cycle), stays below total packet count processable by the switch hardware (3.402 packets/cycle) at a cycle time of $62{,}5$ ms, according to formula (58) (see measurement summary chapter of A.4.1). In conclusion, the APL switch of Manufacturer B fulfills its packet-throughput requirements according to Table 17 regarding the desired packet processing behavior.

A.5 Hardware test – Upstream traffic (Alternative 2), Manufacturers A&B

Disclaimer: This chapter documents traffic measurement results in full detail and is meant to provide extensive background information on the results summarized in chapter 7.3 and 7.6. However, this level of detail is not necessary for understanding the essence of said results and serves only for the profound understanding of the measurement evaluation.

The following hardware measurement follows the network setup according to Figure 19 (chapter 6.4).

Note: A detailed description for better understanding and interpreting the measurement result figures is presented in chapter 6.5

A.5.1 Packet-throughput analysis - Packet processing @ *'SimultaneousTrafficBurst'*

Note: The following upstream traffic analysis follows the same measurement principles as conducted in chapter A.4.

The only difference is the change in transmission speed at the Fast Ethernet egress port of the APL switch from 100 Mbit/s to 10 Mbit/s amplifying the bottleneck problem, which will be analyzed in regards of packet-throughput behavior of the APL field switch.

The following packet load parameters stated in Table 18 have been used for testing:

Table 18: Upstream traffic analysis (Alternative 1) Manufacturers A&B – Traffic parameters

	UDP real-time data, single sensor			
number of field devices	24 (Manufacturer A)		4 (Manufacturer B)	
user priority	6			
total packet payload **(TPP)**	42 Byte (data) + 46 Byte (framing/transmission) = 88 Byte			
packet cycle time **(PCT)**	62,5 ms			
Packet Count per Cycle **(PCC)**	**4 · 1 / 8 · 1 / 16 · 1 / 24 · 1 packets/cycle** (**Manufacturer A**)		**1 · 4 / 2 · 4 / 4 · 4 / 6 · 4 packets/cycle** (**Manufacturer B**)	
Packet data Payload per Cycle **(PPC)**	~184 / ~368 / ~736 / ~1.104 Byte/cycle			
total frame payload per cycle **(FPC)**	~352 / ~704 / ~1.408 / ~2.112 Byte/cycle			
Packet Count per Second **(PCS)**	**64 / packets/s**	**128 packets/s**	**256 packets/s**	**384 packets/s**
packet data payload per second **(PPS)**	64 packets / s · 46 Byte = 2.944 Byte/s (**~2,9 kByte/s**)	128 packets / s · 46 Byte = 5.888 Byte/s (**~5,75 kByte/s**)	256 packets / s · 46 Byte = 11.776 Byte/s (**~11,5 kByte/s**)	384 packets / s · 46 Byte = 17.664 Byte/s (**~17,3 kByte/s**)
total frame payload per second **(FPS)**	64 packets / s · 88 Byte = 5.632 Byte/s (**~5,5 kByte/s**)	128 packets / s · 88 Byte = 11.264 Byte/s (**~11,0 kByte/s**)	256 packets / s · 88 Byte = 22.528 Byte/s (**~22,0 kByte/s**)	384 packets / s · 88 Byte = 33.792 Byte/s (**~33,0 kByte/s**)

Note: Due to the varying amount of ETH/APL media converters for testing Manufacturer A and B, the number of field device emulators and thus the distribution of PCC ('*Packet Count per Cycle*') may vary. Regardless the total sum of UDP traffic stays the same for both.

Table 18 shows all relevant traffic scenario parameters for the follow-up measurement. The *Packet Count per Second* (PCC) shows that up to 384 packets of simultaneous UDP traffic gets generated complying to the '*SimultaneousTrafficBurst*' test condition. Additionally, the *Packet Count per Cycle* (PCC) always stays below 128 packets to not exceed the '*queueLength*' limit of the switch and cause accidental packet loss. The total frame payload per second (PFS) which states the total amount of transmission payload via the APL line stays below the '*dataRate*' limit of $10\ \mathrm{Mbit/s} \cong 1{,}19\ \mathrm{MByte/s}$.

The following figures show the packet-throughput behavior of the switch when using the stated in Table 18. Figure 107 to Figure 110 show the packet-throughput behavior of Manufacturer A in upstream direction. Figure 111 to Figure 114 show the packet-throughput behavior of Manufacturer B in upstream direction.

The blue line shows the entire captured UDP traffic send by the field device emulators to the workstation at $\sim 64 \ldots 384\ \mathrm{packets/s}$. The brown line shows the UDP traffic from one field device emulator at $16\ \mathrm{packets/s}$ for Manufacturer A and $96\ \mathrm{packets/s}$ for Manufacturer B.

Note: The total packet count of recorded UDP traffic in upstream direction resembles all field device emulators sending packets.

The TAP device which was used for traffic measurement was directly placed at the APL switch Fast Ethernet egress port (according to Figure 19). Thus, all outgoing UDP packets of the field device emulators, which are processed and forwarded by the APL switch are captured.

Besides measuring the packet count of one filed device the purpose of additional capturing the total UDP traffic is for better recognition of stability issues, regarding packet-throughput and potential packet loss.

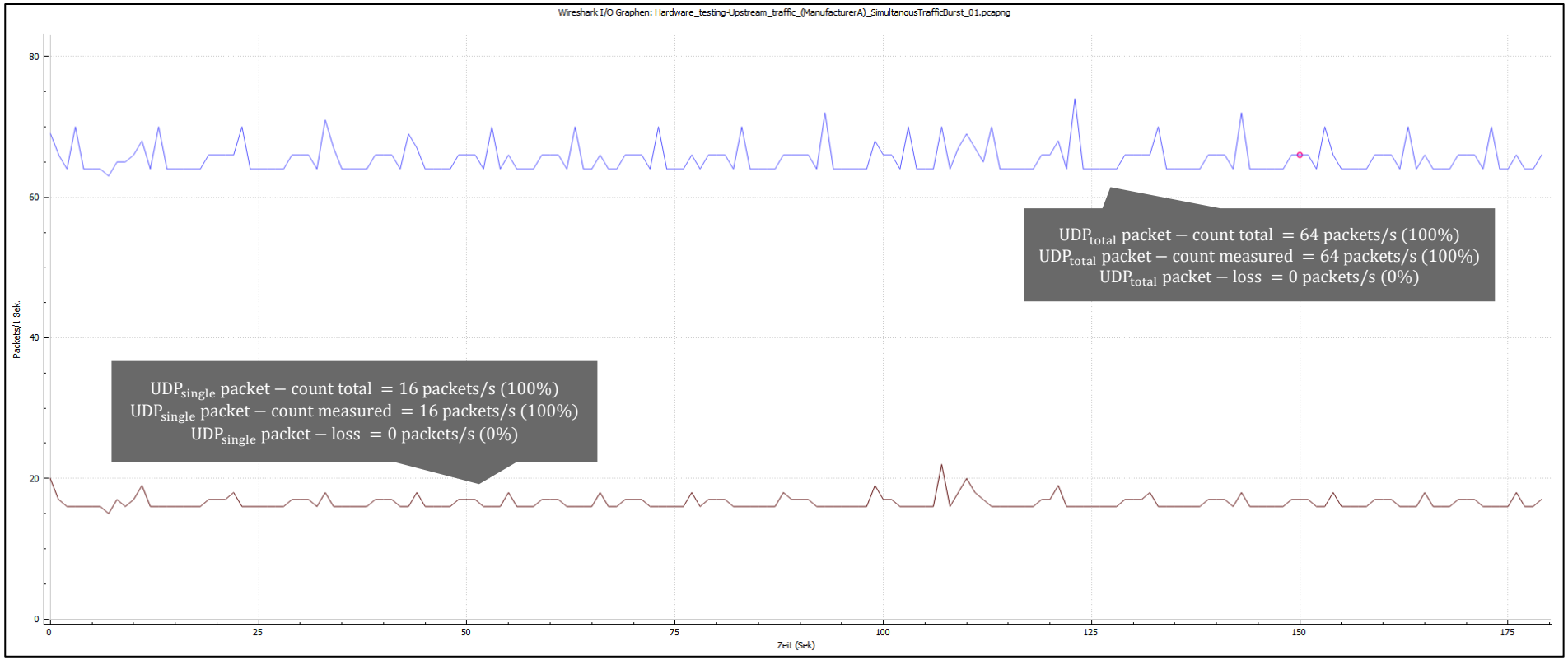

Figure 107: Upstream traffic analysis **(Alternative2), Manufacturer A** - Measurement results (4·1 UDP packets / 62,5 ms) – total packet count

Figure 107 shows that UDP traffic does not struggle with packet loss at a total PCS (‘*Packet Count per Second*’) of 64 packets/s . Hence, all UDP packets are forwarded successfully by the APL switch.

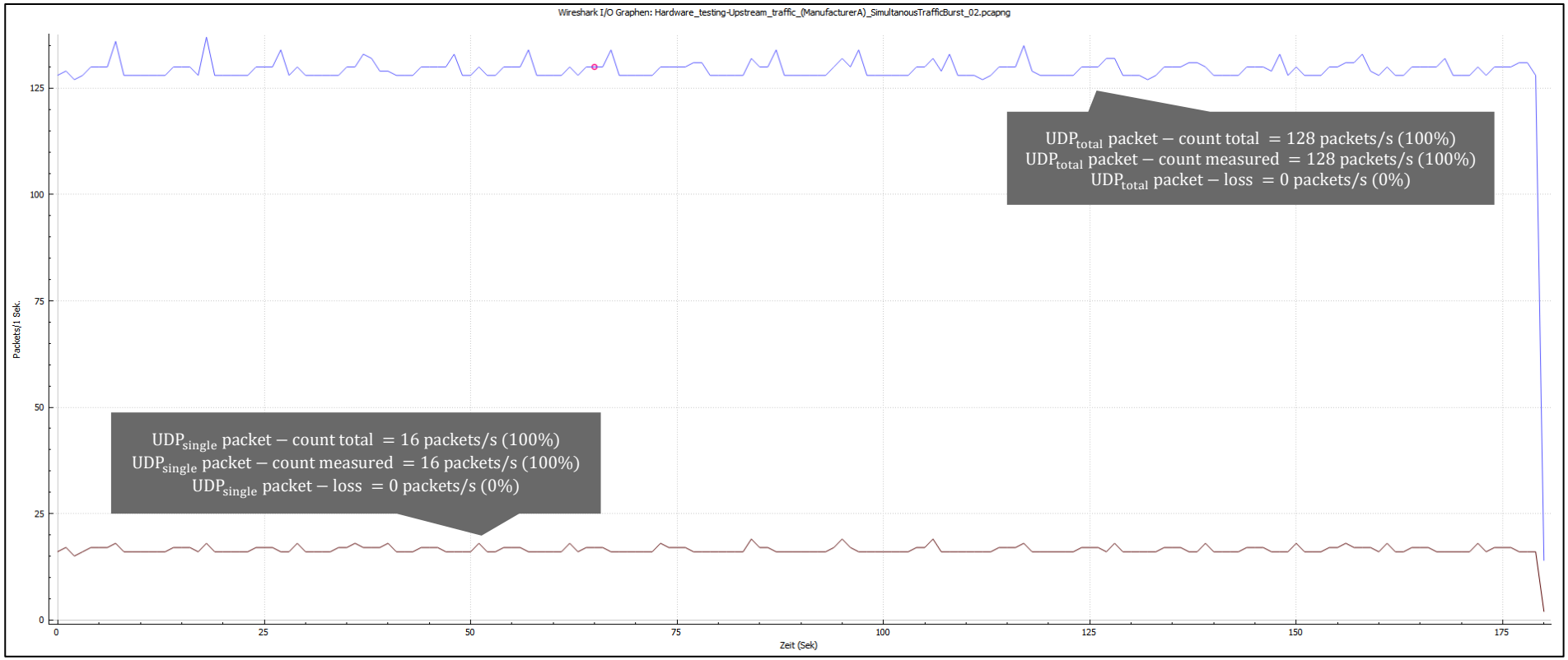

Figure 108: Upstream traffic analysis **(Alternative2), Manufacturer A** - Measurement results (8·1 UDP packets / 62,5 ms) – total packet count

Figure 108 shows that by doubling the total PCS (‘*Packet Count per Second*’) of UDP traffic from ~ 64 packets/s to ~ 128 packets/s, packet loss of UDP traffic still does not occur. Hence, all UDP packets are forwarded successfully by the APL switch.

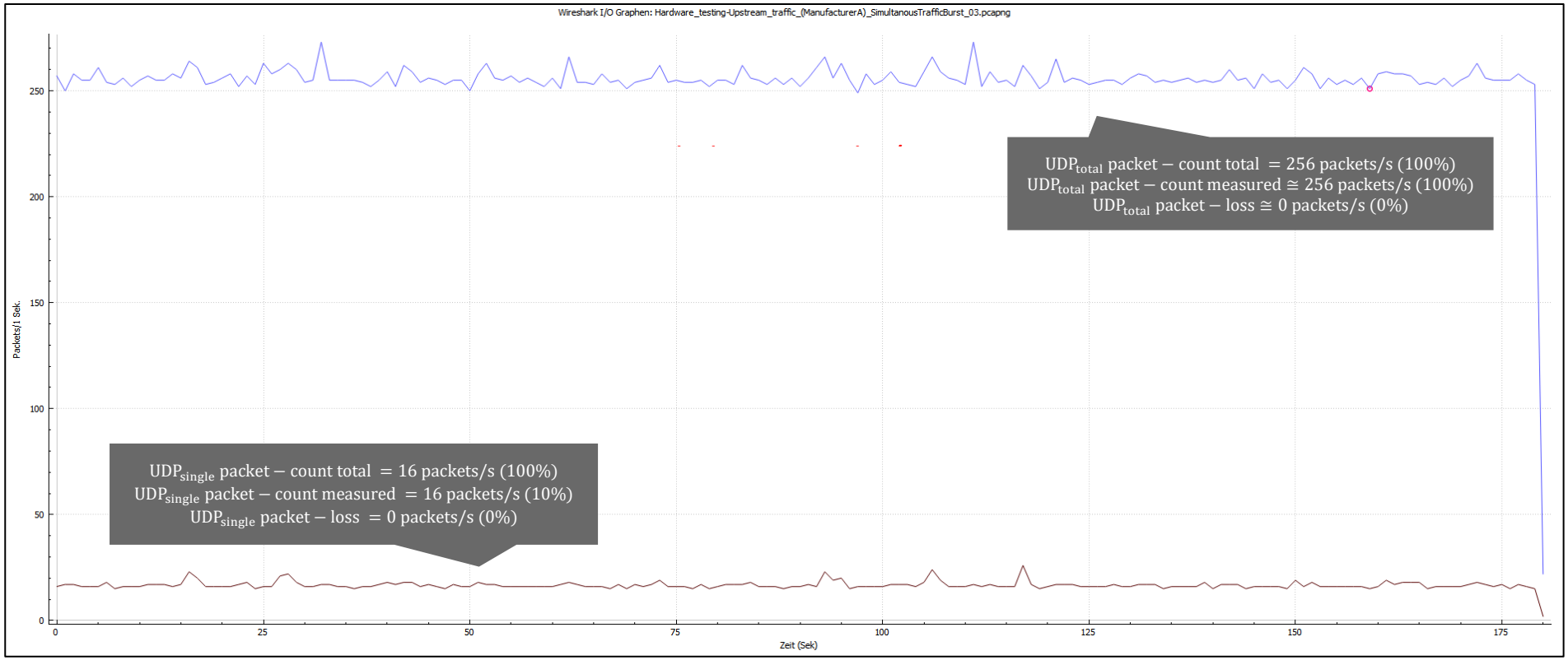

Figure 109: Upstream traffic analysis **(Alternative2), Manufacturer A** - Measurement results (16·1 UDP packets / 62,5 ms) – total packet count

Figure 109 shows that by quadrupling the total PCS ('*Packet Count per Second*') of UDP traffic from ~ 64 packets/s to ~ 256 packets/s, packet loss of UDP traffic still does not occur. Hence, all UDP packets are forwarded successfully by the APL switch.

However, by comparing Figure 107 to Figure 109, the packet-throughput behavior of UDP traffic shows increasing instabilities, which hints at a potential packet loss by further increasing the packet load by increasing the number of used field devices.

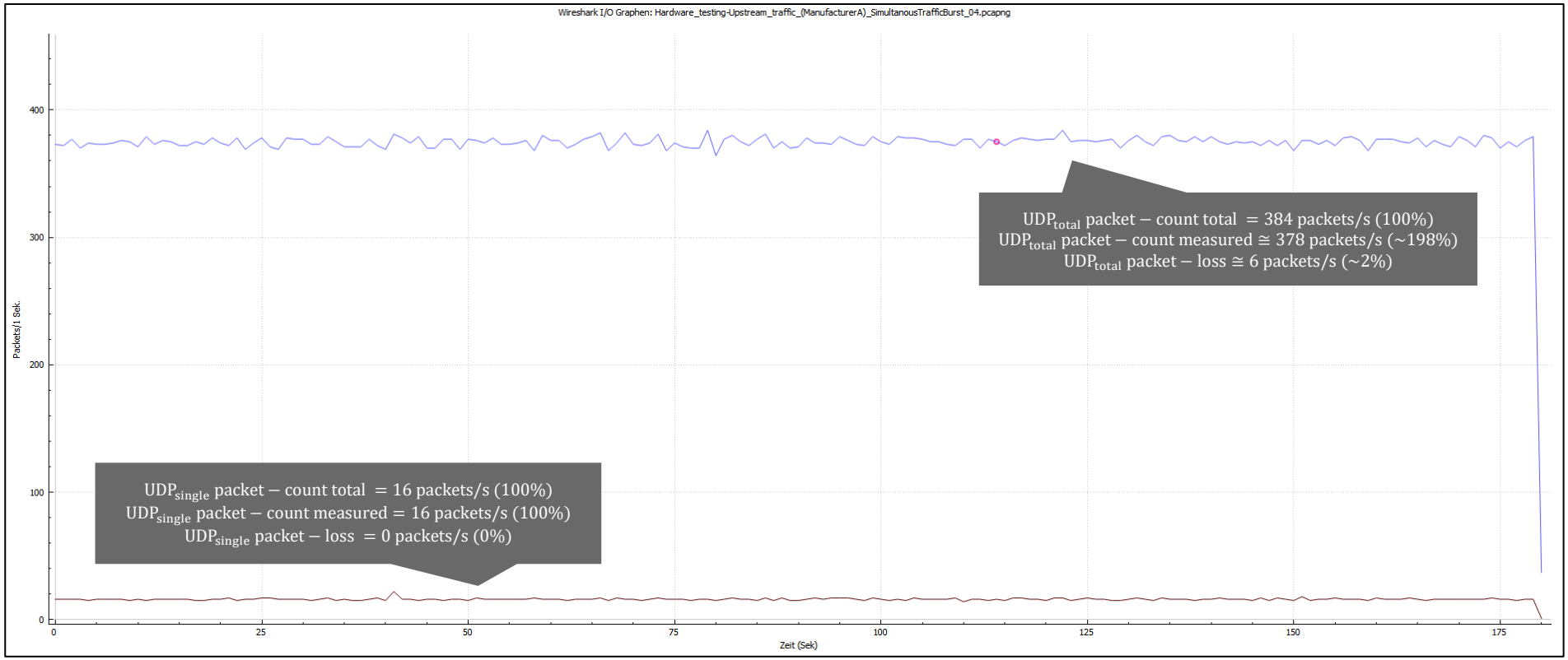

Figure 110: Upstream traffic analysis **(Alternative2), Manufacturer A** - Measurement results (24·1 UDP packets / 62,5 ms) - total packet count

Figure 110 shows that by sixfold the total PCS ('*Packet Count per Second*') of UDP traffic from ~64 packets/s to ~384 packets/s, packet loss increases from 0% to 2%. Said packet loss of UDP packets happens due to the same phenomenon which was already described in the measurement summary of chapter A.4.1. Hence, the packet prioritization of the APL switch works correctly.

However, in contrary to the measurements conducted in chapter A.4, the datarate of the APL egress trunk port has changed from 100 Mbit/s to 10 Mbit/s. Thus, the PPT must be reevaluated to gain insight about the total packet count processable by the switch hardware while working with a tenfold limited egress port datarate. The following PPTs have been calculated with the help of the hardware delay time stated in Table 2 (chapter 5.4) and the traffic parameters stated in Table 18.

$$PPT_{type} = x_{packets,type} \cdot \left(t_{bridge} + t_{port} + t_{cable} + t_{prop}\right) \quad (59)$$

Note: The store and forward bridge delay is calculated, based on the datarate of the internal Ethernet bridge of the APL switch, handling the transition between its Fast Ethernet ingress port, working at $100\ \mathrm{Mbit/s}$, and Ethernet-APL egress spur ports, working at $10\ \mathrm{Mbit/s}$.

To empty all packet queues for one cycle, the following PPT derives:

$$\rightarrow PPT_{\mathrm{UDP}} = x_{\mathrm{packets,UDP}} \cdot \left(6\ \mu s + \frac{88\ Byte}{10\ Mbit/s} + 4\ \mu s + 1{,}33\ \mu s\right) \quad (60)$$

$$\rightarrow PPT_{\mathrm{UDP}} = 24 \cdot \left(6\ \mu s + \frac{88\ Byte}{10\ Mbit/s} + 4\ \mu s + 1{,}33\ \mu s\right) \cong 1{,}96\ ms \quad (61)$$

$$\rightarrow (PPT_{\mathrm{total}} = PPT_{\mathrm{UDP}} \cong 1{,}96\ ms) < T_{UDP} = 62{,}5\ ms \quad (62)$$

The calculation shows that theoretical PPT ('*Packet Processing Time*') needed for emptying all ingress queues stays below the packet cycle time, thus preventing packet discarding by not exceeding the '*queueLength*' limit of the APL switch via packet overlapping inside the packet queues. By dividing the UDP cycle time with the UDP PPT the following total UDP packet count derives:

$$\rightarrow\ x_{\mathrm{packets,UDP,max}} = \frac{T_{\mathrm{UDP}}}{PPT_{\mathrm{UDP}}} \quad (63)$$

$$\rightarrow x_{\mathrm{packets,UDP,max}} = \frac{62{,}5\ ms/\ cycle}{\mathbf{6}\ \mu s + \frac{88\ Byte/packet}{10\ Mbit/s} + 4\ \mu s + 1{,}33\ \mu s} = \frac{62{,}5\ ms}{\mathbf{6}\ \mu s + 70{,}4\ \mu s + 4\ \mu s + 1{,}33\ \mu s}\frac{packets}{cycle} \cong 765\frac{packets}{cycle} \quad (64)$$

The calculation shows that theoretical total packet count processable by the APL switch grants a UDP PPC of $765\ \mathrm{packets/cycle}$, thus yielding a total UDP PPS of $12.235\ \mathrm{packets/s}$ in terms of packet processing capability. This shows that compared to the previous packet processing limit of $54.437\ \mathrm{packets/s}$, while working at a datarate of 100 Mbit/s, that the total packet processing capacity almost decreased by five times. Said limitation may lead to potential packet loss in the follow-up measurements by increasing the packet load.

Summary (Figure 107 to Figure 110): The measurements show, that stable packet-throughput of simultaneous incoming traffic, by gradual increase of UDP traffic up to $\sim 384\ \mathrm{packets/s}$, according to the '*SimulaneousTrafficBurst*' condition, is always ensured.

The direct comparison between the measurements conducted in chapter A.4.1 and A.5.1, regarding the packet processing behavior of Manufacturer A, shows no difference. In conclusion, the APL switch of Manufacturer A fulfills its packet-throughput requirements according to Table 18 regarding the desired packet processing behavior.

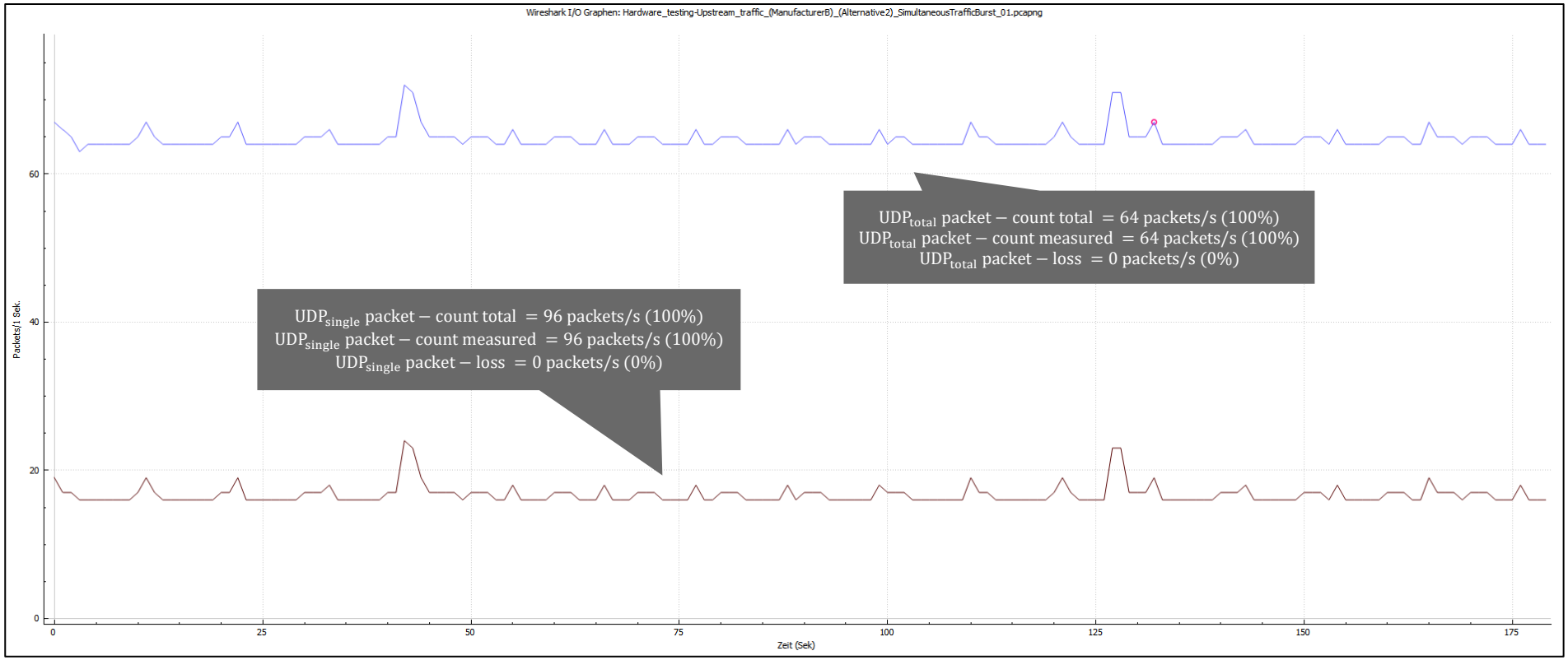

Figure 111: Upstream traffic analysis **(Alternative 1), Manufacturer B** - Measurement results (4·1 UDP packets / 62,5 ms) – total packet count

Figure 111 shows that UDP traffic does not struggle with packet loss at a total PCS ('*Packet Count per Second*') of $64\ \text{packets/s}$. Hence, all UDP packets are forwarded successfully by the APL switch.

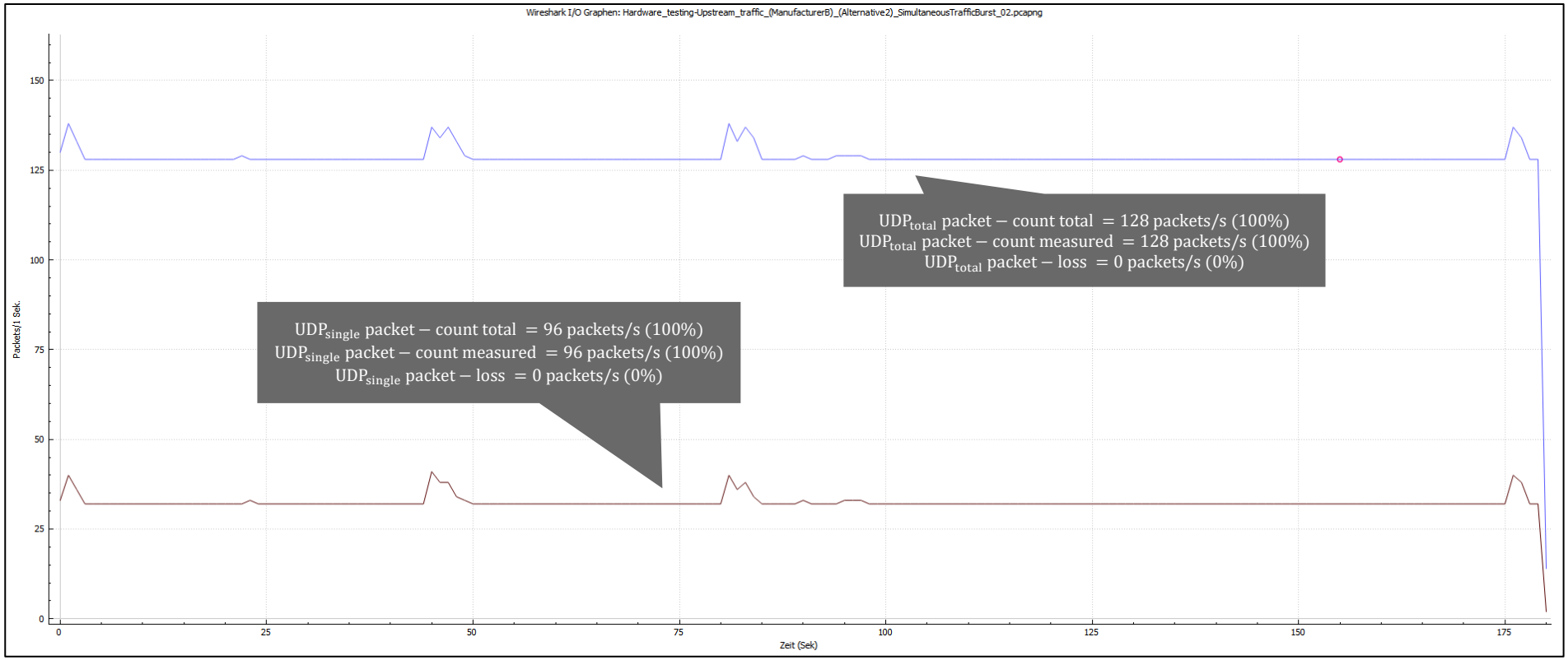

Figure 112: Upstream traffic analysis **(Alternative 1), Manufacturer B** - Measurement results (8·1 UDP packets / 62,5 ms) – total packet count

Figure 112 shows that by doubling the total PCS ('*Packet Count per Second*') of UDP traffic from $\sim 64\ packtes/s$ to $\sim 128\ packets/s$, packet loss of UDP traffic still does not occur. Hence, all UDP packets are forwarded successfully by the APL switch.

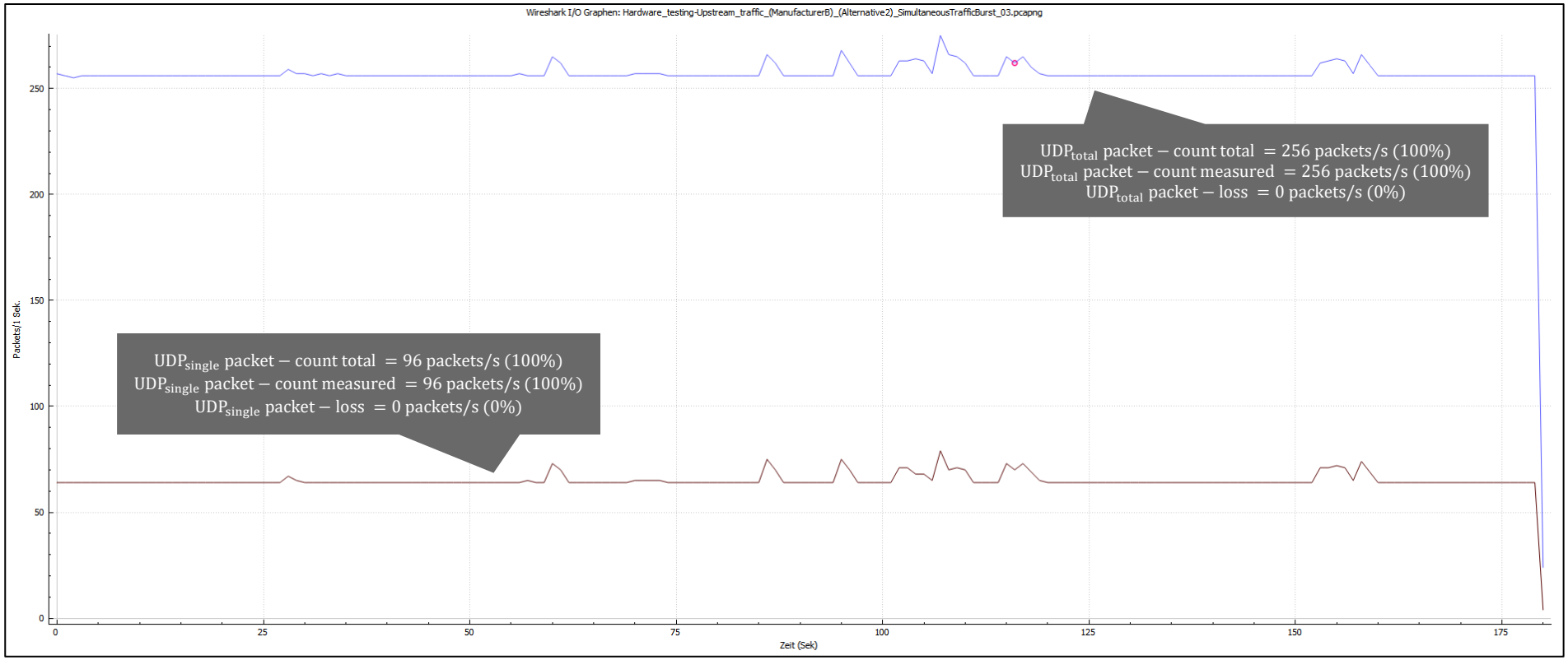

Figure 113: Upstream traffic analysis **(Alternative 1), Manufacturer B** - Measurement results (16·1 UDP packets / 62,5 ms) – total packet count

Figure 113 shows, at by quadrupling the total PCS ('*Packet Count per Second*') of UDP traffic from ~ 64 packets/s to ~ 256 packets/s, packet loss of UDP traffic still does not occur. Hence, all UDP packets are forwarded successfully by the APL switch.

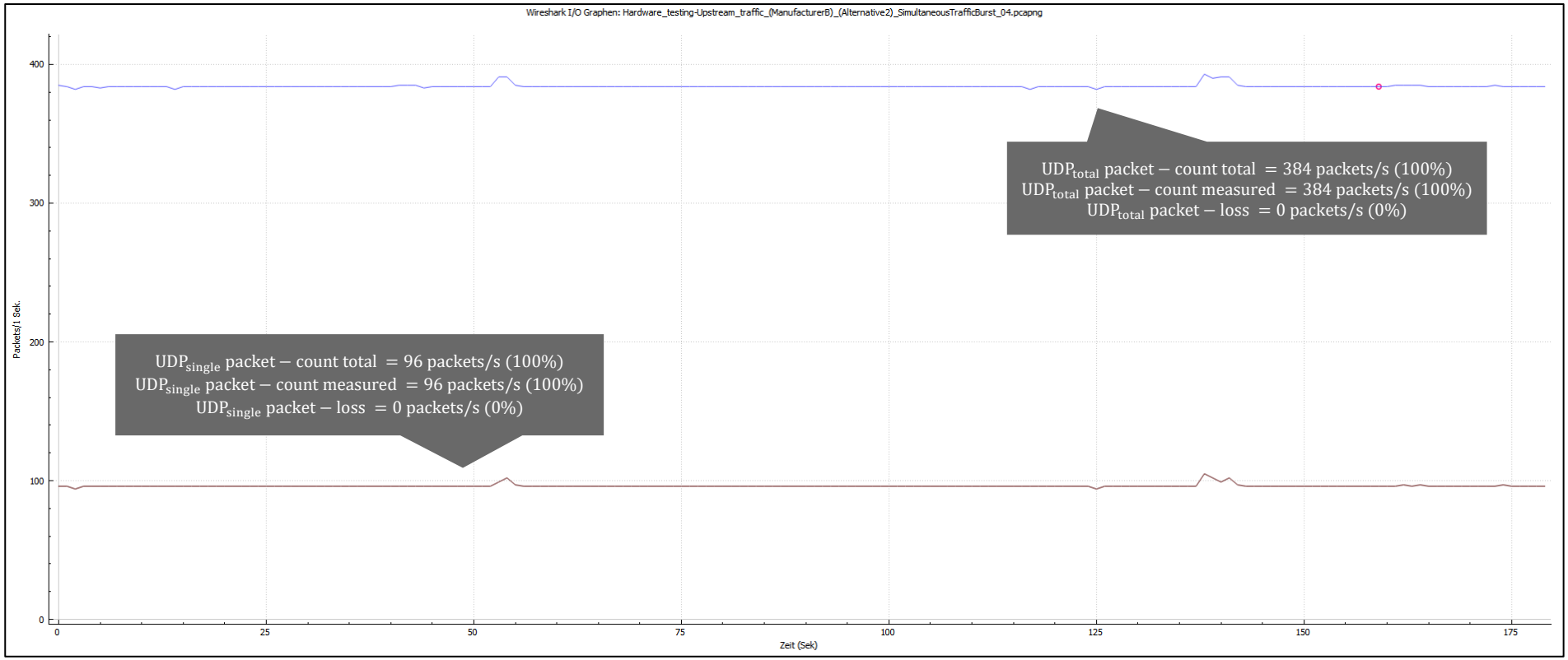

Figure 114: Upstream traffic analysis **(Alternative 1), Manufacturer B** - Measurement results (24·1 UDP packets / 62,5 ms) - total packet count

Figure 114 shows that by sixfold the total PCS ('*Packet Count per Second*') of UDP traffic from ~ 64 packets/s to ~ 384 packets/s, packet loss of UDP traffic still does not occur. Hence, all UDP packets are forwarded successfully by the APL switch.

Summary: (Figure 111 to Figure 114): The measurements show, that stable packet-throughput of simultaneous incoming traffic, by gradual increase of UDP traffic up to ~ 384 packets/s, according to the '*SimulaneousTrafficBurst*' condition, is always ensured. The direct comparison between the measurements conducted in chapter A.4.1 and A.5.1, regarding the packet processing behavior of Manufacturer B, shows no difference.

In conclusion, the APL switch of Manufacturer B fulfills its packet-throughput requirements according to Table 18 regarding the desired packet processing behavior.

A.5.2 Packet-throughput analysis – absolute packet processing limit

Similar to downstream traffic, hardware test shall validate if the switches manage to uphold stable packet-throughput of higher priority traffic while gradually increasing the packet load.

A.5.2.1 Packet processing @ decreasing UDP cycle time

The following tests have been conducted by increasing UDP traffic through decreasing its cycle time. The traffic parameters used in these tests are stated in Table 19.

Table 19: Downstream traffic analysis (Alternative 1), Manufacturers A&B – Traffic parameters, increasing UDP traffic @ varying cycle time

	UDP real-time data, *single sensor*		
user priority	6		
total packet payload **(TPP)**	46 Byte (data) + 42 Byte (framing/transmission) = 88 Byte		
packet cycle time **(PCT)**	**31,25 ms / 15,625 ms / 7,8125 ms**		
Packet Count per Cycle **(PCC)**	24 · 1 packets/cycle (Manufacturer A)	24 · 1 packets/cycle (Manufacturer A)	
Packet data Payload per Cycle **(PPC)**	24 packets / cycle · 46 Byte = 1.104 Byte/cycle (~1,1 kByte/c)		
total frame payload per cycle **(FPC)**	24 p/c · 88 Byte = 2.112 Byte/cycle (~2 kByte/cycle)		
Packet Count per Second **(PCS)**	**768 packets/s**	**1536 packets/s**	**3072 packets/s**
packet data payload per second **(PPS)**	768 p/s · 46 Byte = 35.328 $Byte/s$ (~**34,5 $kByte/s$**)	1536 p/s · 46 Byte = 70.656 $Byte/s$ (~**69,0 $kByte/s$**)	3072 p/s · 46 Byte = 141.312 $Byte/s$ (~**138,0 $kByte/s$**)
total frame payload per second **(FPS)**	768 p/s · 88 Byte = 67.584 $Byte/s$ (~**66 $kByte/s$**)	1536 p/s · 88 Byte = 135.168 $Byte/s$ (~**132,0 $kByte/s$**)	3072 p/s · 88 Byte = 270.336 $Byte/s$ (~**264 $kByte/s$**)

The following figures show the packet-throughput behavior of the switch when using the stated in Table 19. Figure 115 to Figure 120 show the packet-throughput behavior of Manufacturer A in upstream direction. Figure 121 to Figure 126 show the packet-throughput behavior of Manufacturer B in upstream direction.

The blue line shows the entire captured UDP traffic send by all field device emulators at $\sim 768 \ldots 3072$ packets/s for both Manufacturers, while the brown line shows the UDP traffic from one field device emulator at $32 \ldots 128$ packets/s for Manufacturer A and $192 \ldots 768$ packets/s for Manufacturer B.

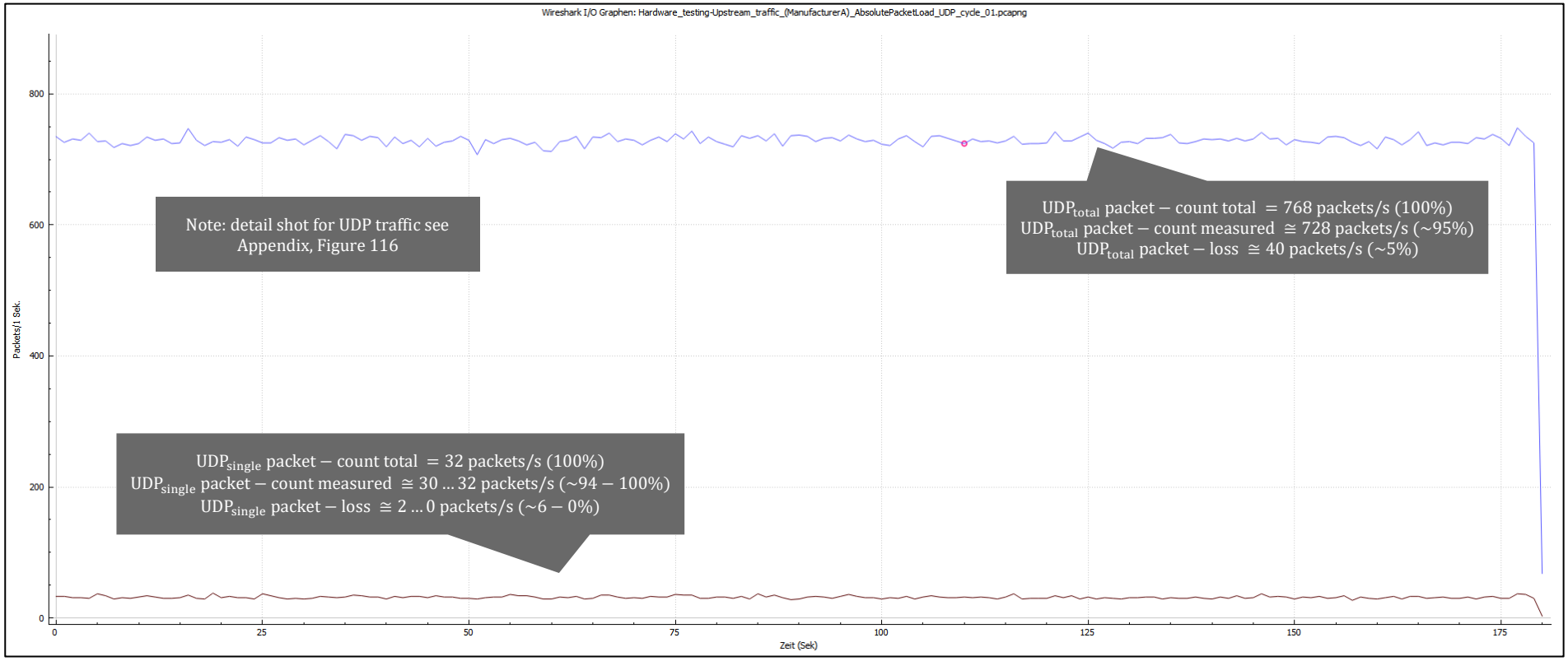

Figure 115: Upstream traffic analysis **(Alternative 2), Manufacturer A** - Measurement results (1·24 UDP packets / 31,25 ms) - total packet count

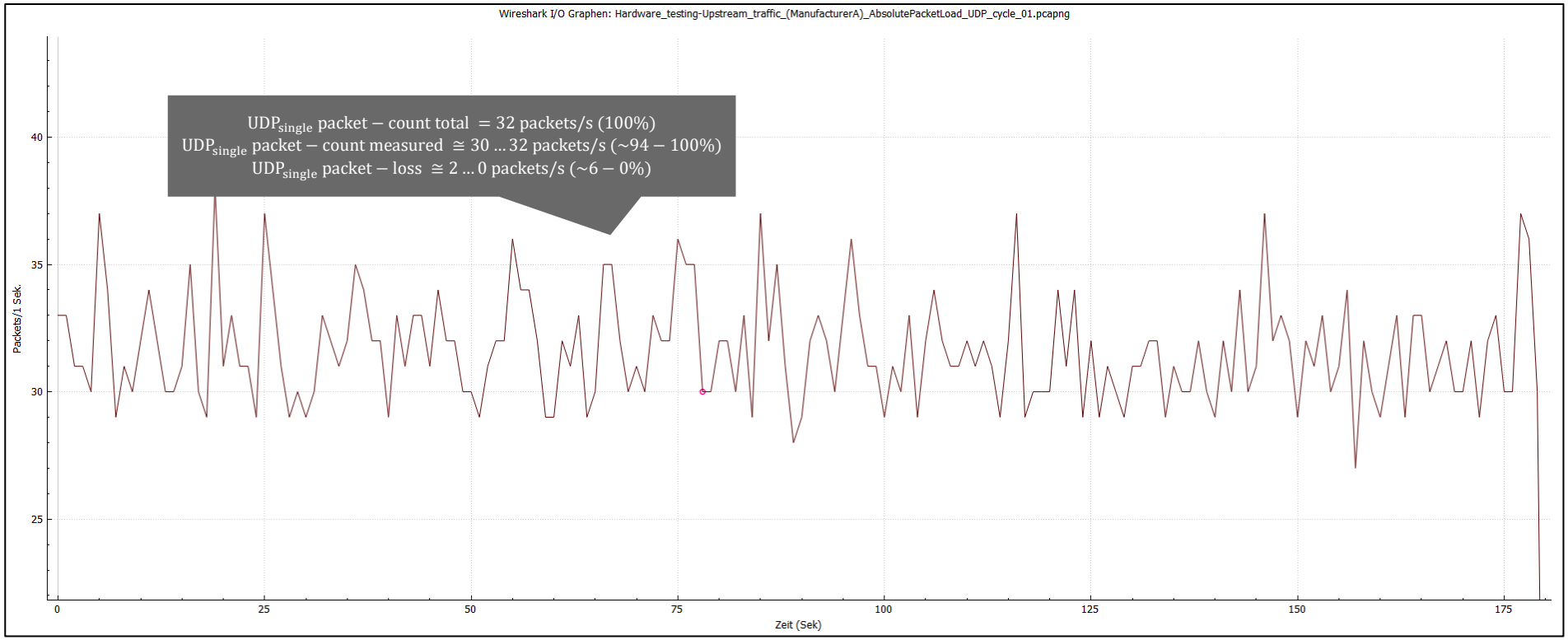

Figure 116: Upstream traffic analysis **(Alternative 2), Manufacturer A** - Measurement results (1·24 UDP packets / 31,25 ms) – single source packet count

Figure 115 shows that UDP traffic has a packet loss of 5% at a total PCS ('*Packet Count per Second*') of 768 packets/s . Said packet loss of UDP packets happens due to the same phenomenon which was already described in the measurement summary of chapter A.4.1. Hence, the packet prioritization of the APL switch works correctly.

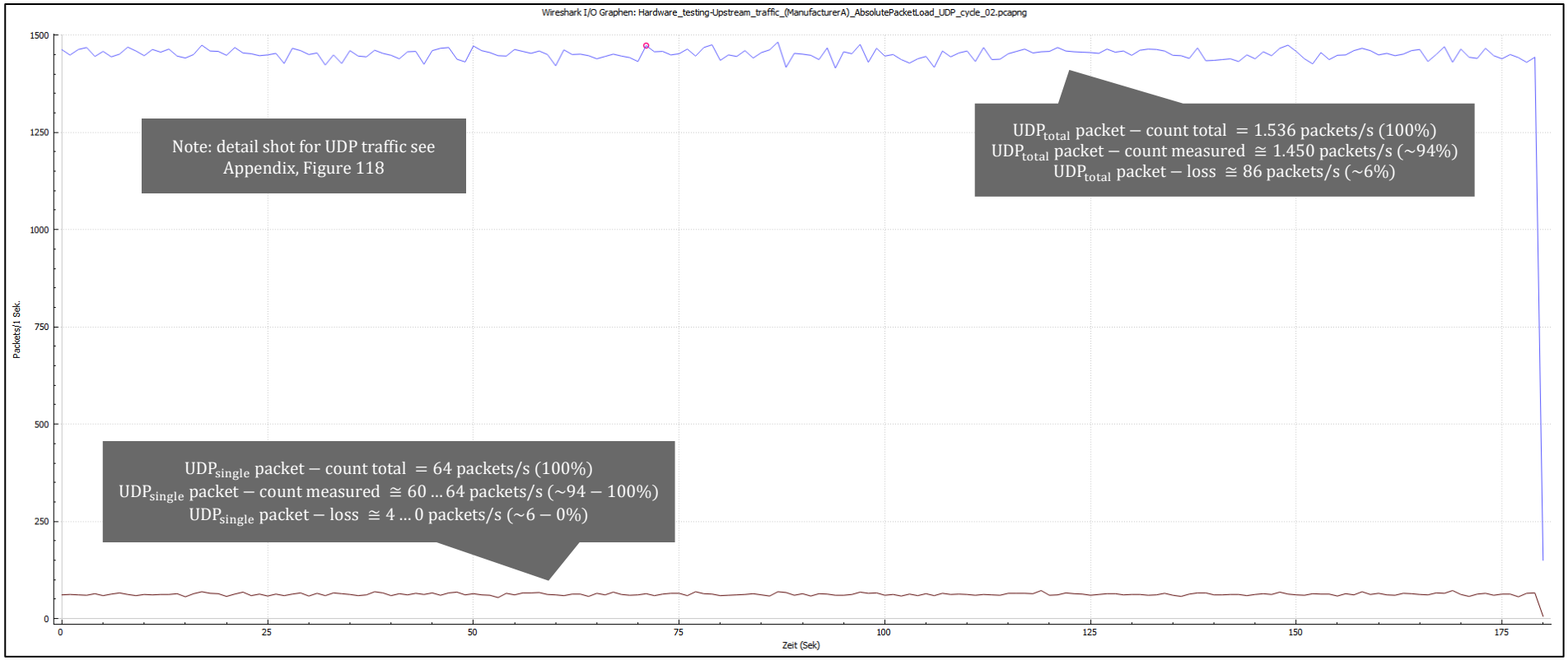

Figure 117: Upstream traffic analysis **(Alternative 2), Manufacturer A** - Measurement results (1·24 UDP packets / 15,625 ms) - total packet count

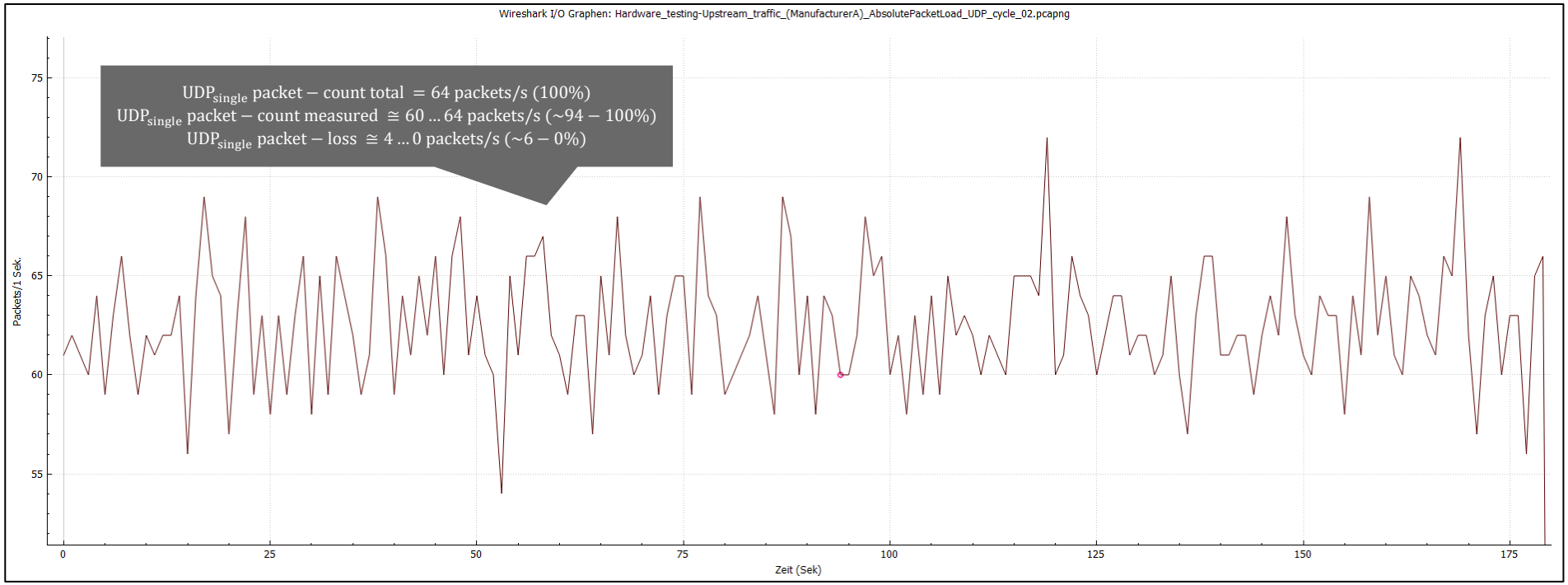

Figure 118: Upstream traffic analysis **(Alternative 2), Manufacturer A** - Measurement results (1·24 UDP packets / 15,625 ms) – single source packet count

Figure 117 shows that by doubling the total PCS ('*Packet Count per Second*') of UDP traffic from ~768 packets/s to ~1.536 packets/s, packet loss increases from 5% to 6%. The same phenomenon which was referred to in the previous Figure 115 is also apparent in Figure 117, but in a more pronounced form. Hence, the packet prioritization of the APL switch works correctly.

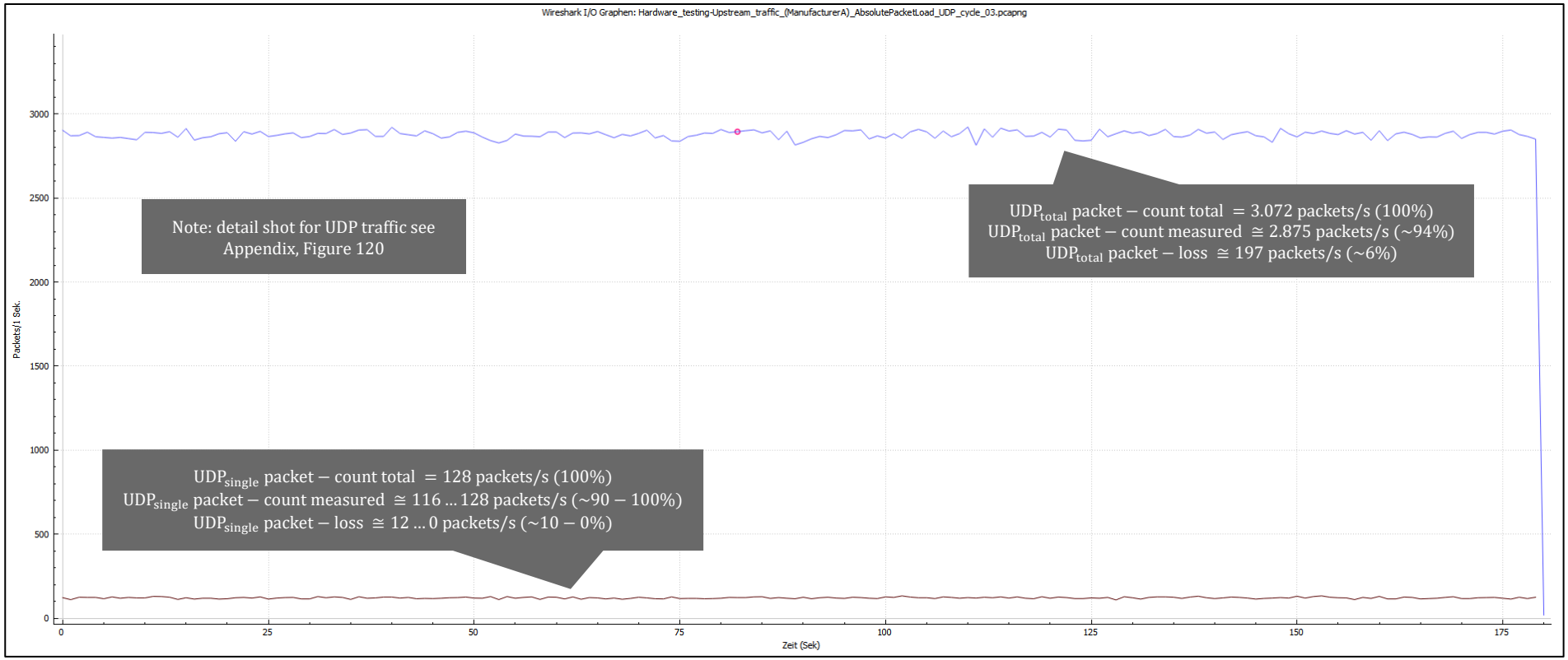

Figure 119: Upstream traffic analysis **(Alternative 2), Manufacturer A** - Measurement results (**1·24** UDP packets **/ 7,8125 ms**) - total packet count

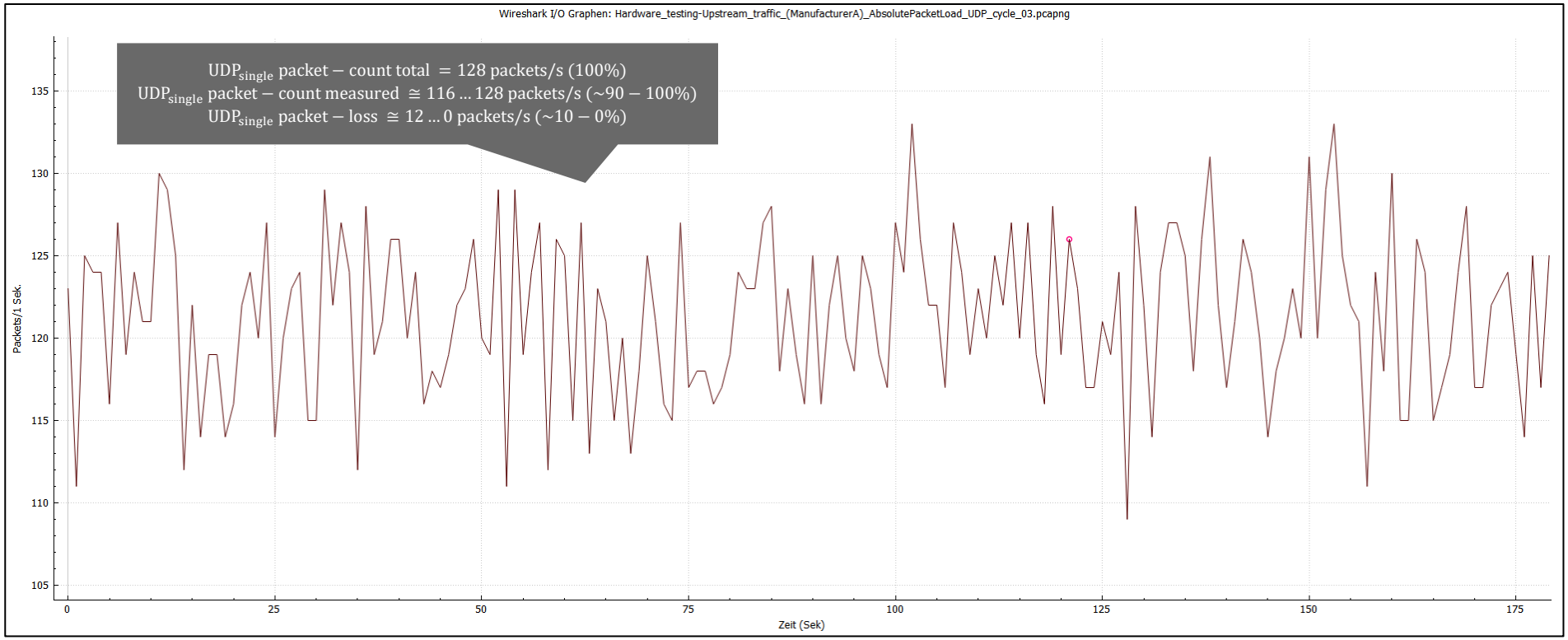

Figure 120: Upstream traffic analysis **(Alternative 2), Manufacturer A** - Measurement results (**1·24** UDP packets **/ 7,8125 ms**) – single source packet count

Figure 119 shows that by quadrupling the total PCS ('*Packet Count per Second*') of UDP traffic from ~768 packets/s to ~3.072 packets/s, packet loss stays at 6%. The same phenomenon which was referred to in the previous Figure 115 is also apparent in Figure 119, but in a more pronounced form. Hence, the packet prioritization of the APL switch works correctly.

Summary: (Figure 115 to Figure 120): The measurements show, that stable packet-throughput of simultaneous incoming traffic, by gradual increase of UDP traffic up to ~3.072 packets/s, is always ensured.

The direct comparison between the measurements conducted in chapter A.4.2.1 and A.5.2.1, regarding the packet processing behavior of Manufacturer A, shows no difference.

In conclusion, the APL switch of Manufacturer A fulfills its packet-throughput requirements according to Table 19 regarding the desired packet processing behavior.

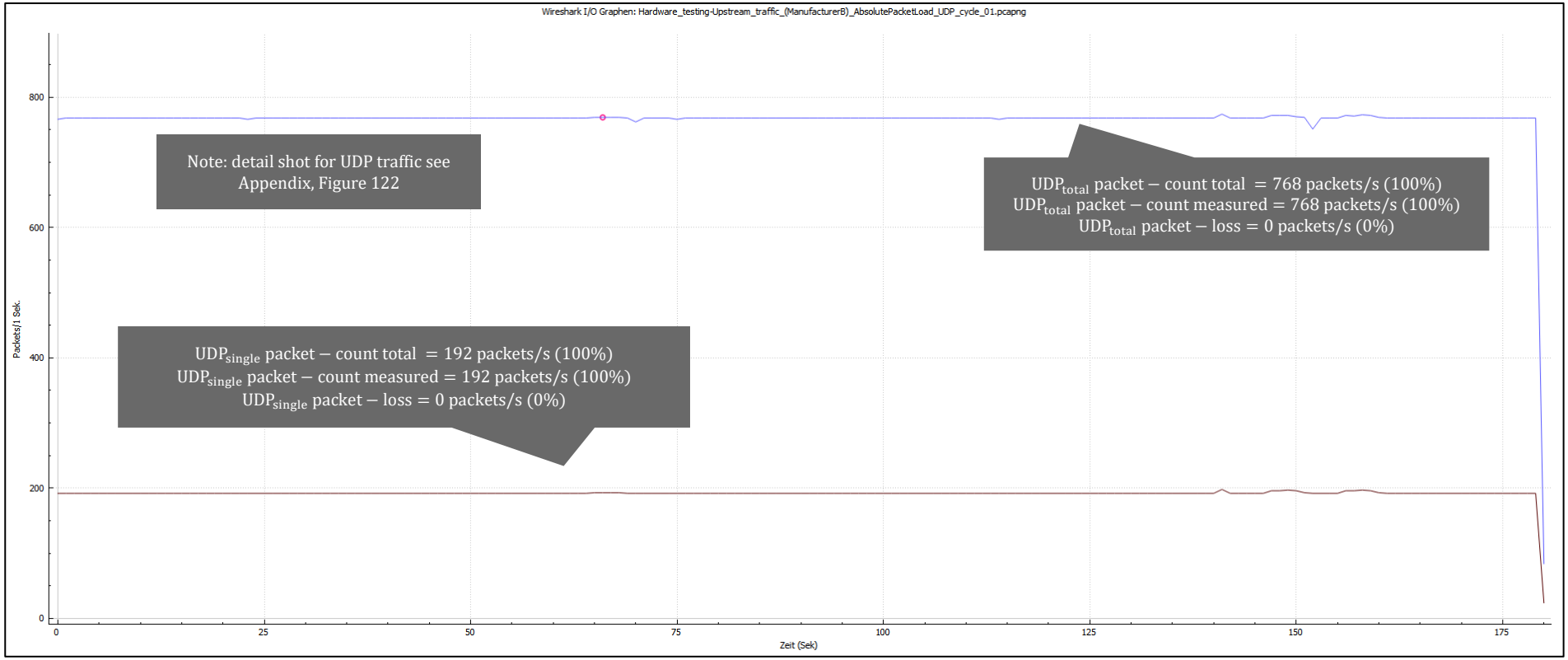

Figure 121: Upstream traffic analysis **(Alternative 2), Manufacturer B** - Measurement results (1·24 UDP packets / 31,25 ms) - total packet count

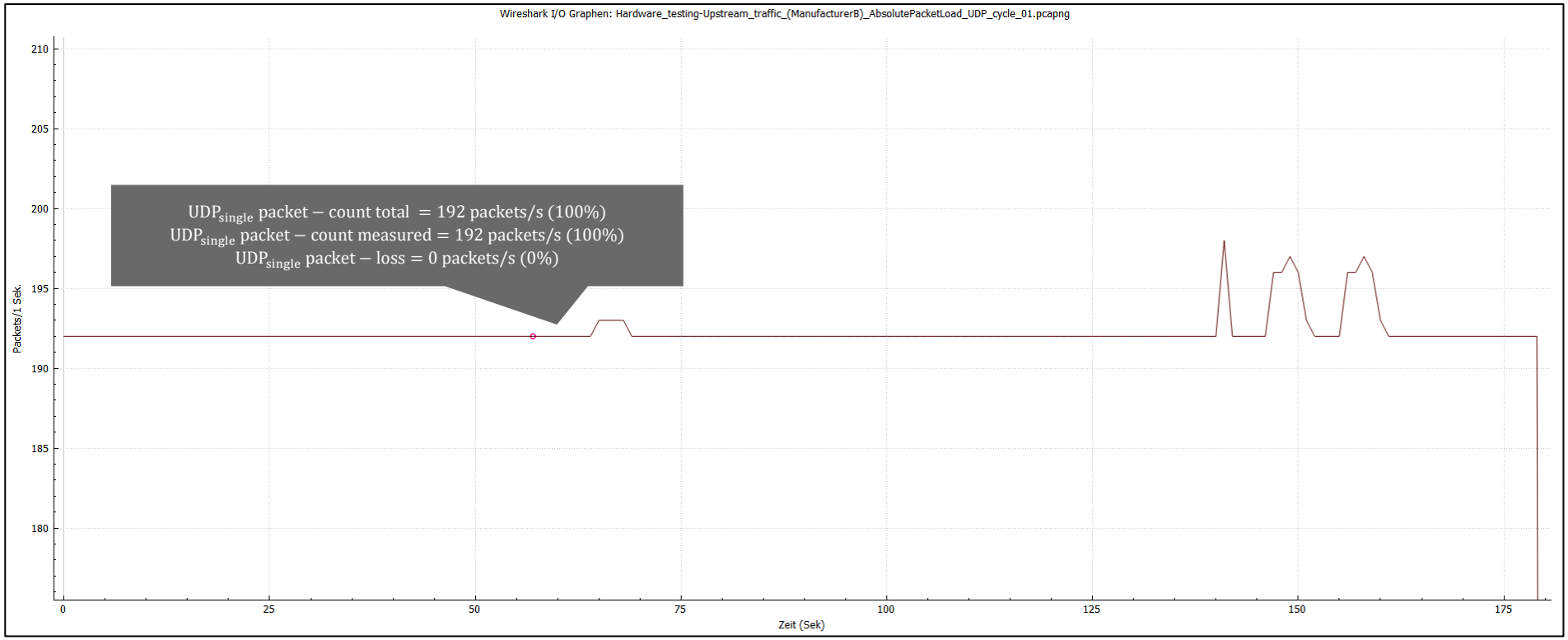

Figure 122: Upstream traffic analysis **(Alternative 2), Manufacturer B** - Measurement results (1·24 UDP packets / 31,25 ms) – single source packet count

Figure 121 shows that UDP traffic does not struggle with packet loss at a total PCS (‘*Packet Count per Second*’) of 768 packets/s . Hence, all UDP packets are forwarded successfully by the APL switch.

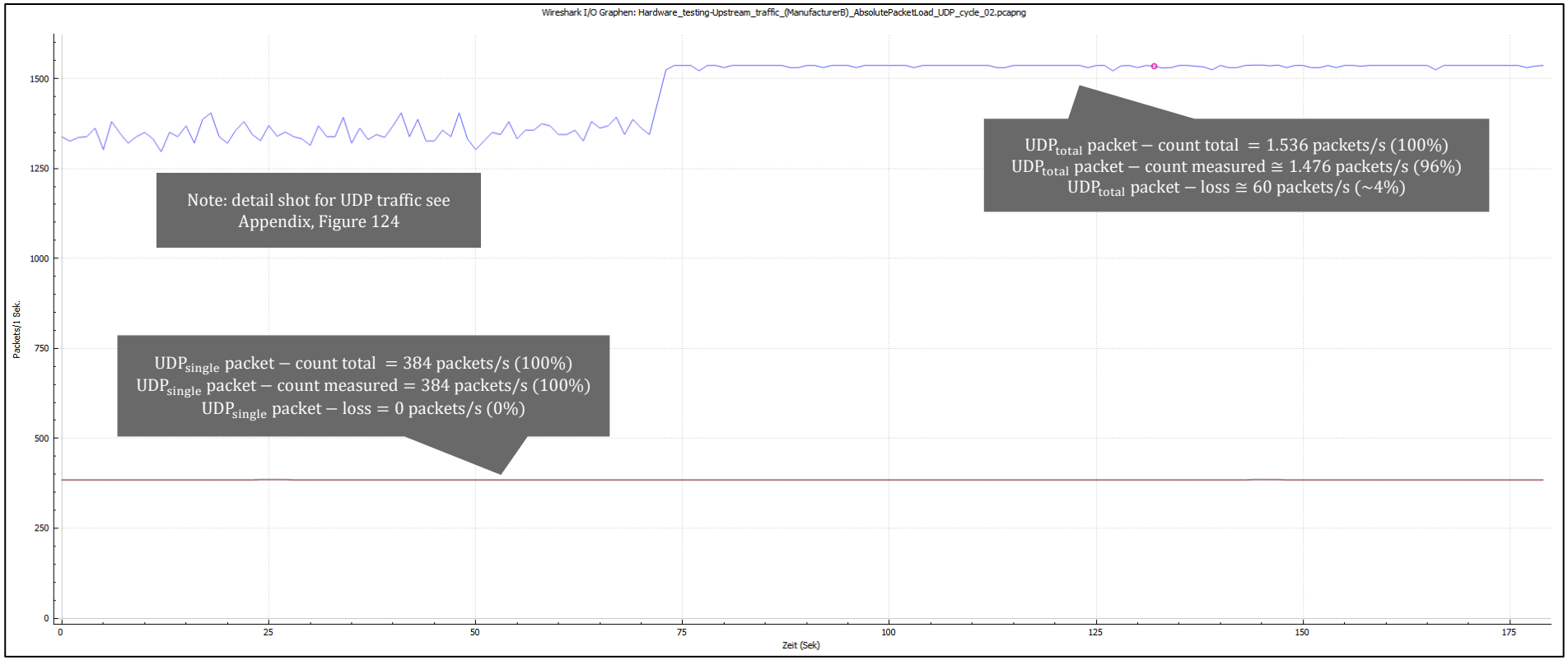

Figure 123: Upstream traffic analysis **(Alternative 2), Manufacturer B** - Measurement results (1·24 UDP packets / 15,625 ms) - total packet count

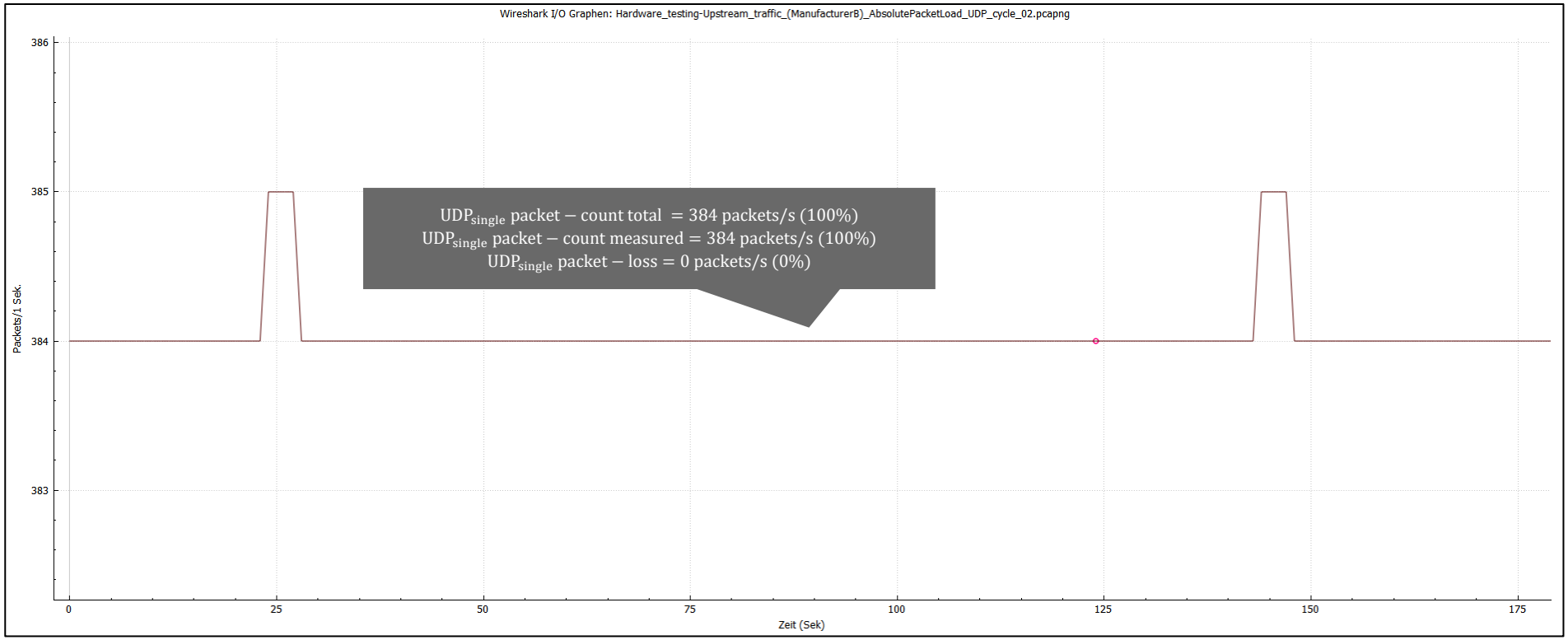

Figure 124: Upstream traffic analysis **(Alternative 2), Manufacturer B** - Measurement results (1·24 UDP packets / 15,625 ms) – single source packet count

Figure 123 shows that by doubling the total PCS ('*Packet Count per Second*') of UDP traffic from ~ 768 packets/s to ~ 1.536 packets/s, packet loss increases from 0% to 4%.

Furthermore, Figure 124 shows that the APL switch of Manufacturer B had severe stability issues, regarding packet-throughput of UDP packets, while working with a limited egress port datarate of 10 Mbit/s. Said issues also lead to the entire shutdown of APL ingress spur ports, no longer processing packets, while running packet load tests.

Hence, the packet processing of the APL switch does no longer work as intended, which leads to the conclusion, that '*10 Mbit/s –to-10 Mbit/s*' upstream connections should be avoided to prevent packet loss.

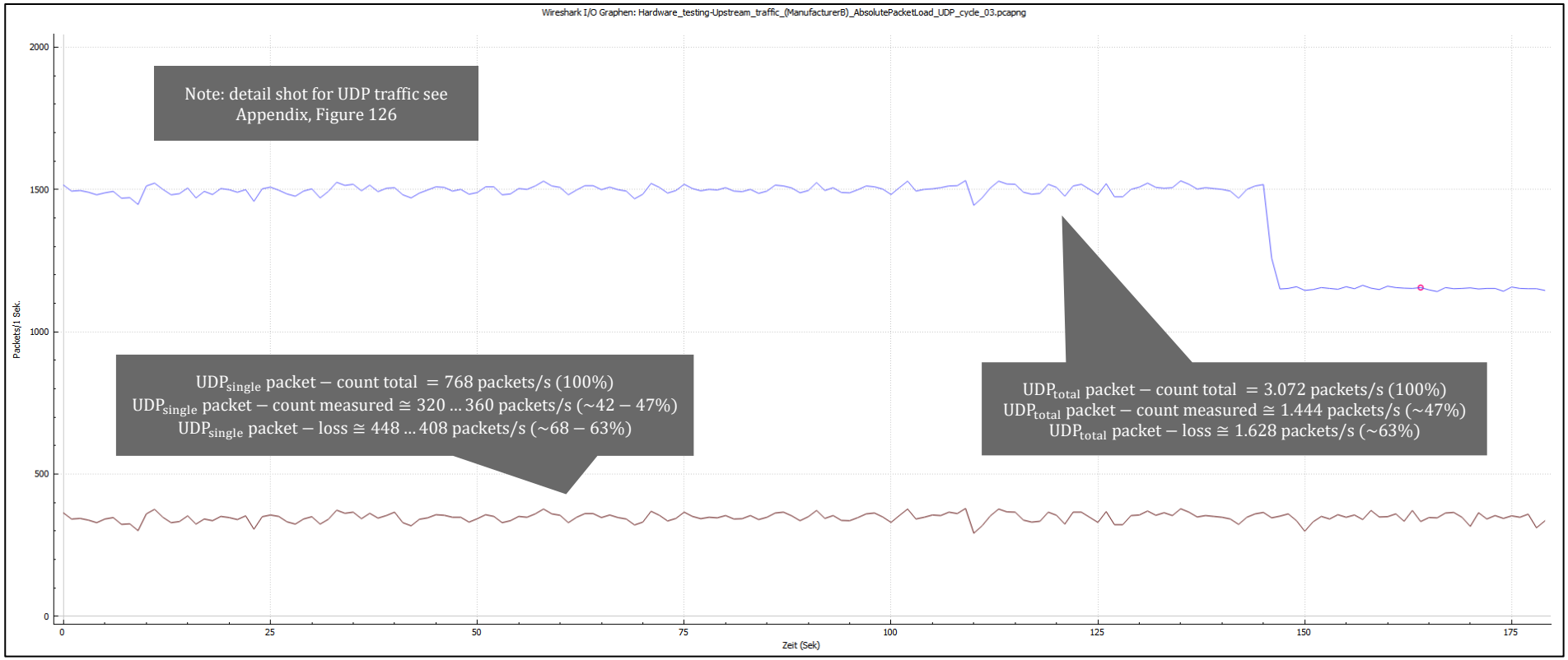

Figure 125: Upstream traffic analysis **(Alternative 2), Manufacturer B** - Measurement results (1·24 UDP packets / 7,8125 ms) - total packet count

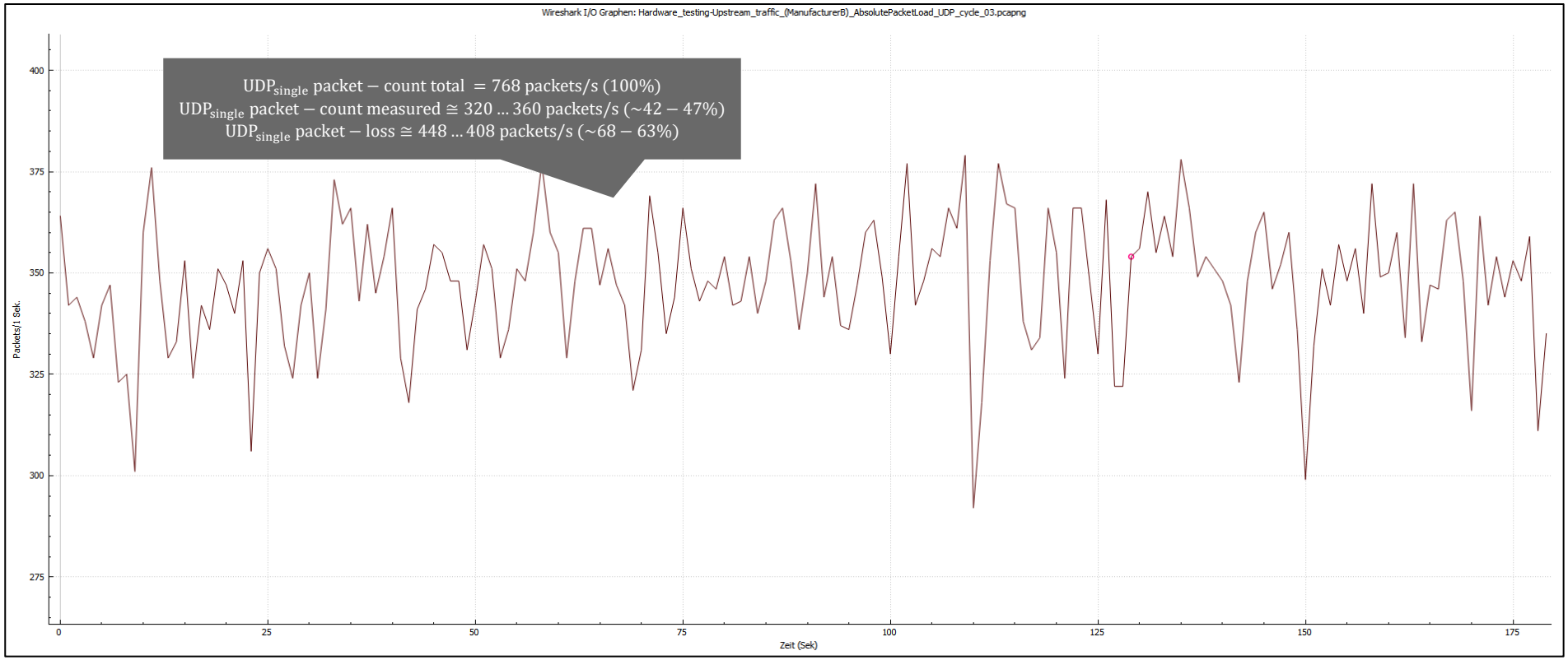

Figure 126: Upstream traffic analysis **(Alternative 2), Manufacturer B** - Measurement results (1·24 UDP packets / 7,8125 ms) – single source packet count

Figure 125 shows that by quadrupling the total PCS ('*Packet Count per Second*') of UDP traffic from ~768 packets/s to ~3.072 packets/s, packet loss increases from 4% to 63%. The same phenomenon which was referred to in the previous Figure 123 is also apparent in Figure 125, but in a more pronounced form. Hence, the packet processing of the APL switch does no longer work as intended.

Summary: (Figure 121 to Figure 126): The measurements show, that stable packet-throughput of simultaneous incoming traffic, by gradual increase of UDP traffic up to ~ 3.072 packets/s, is not always ensured.

The direct comparison between the measurements conducted in chapter A.4.2.1 and A.5.2.1, regarding the packet processing behavior of Manufacturer B, shows that the APL switch of manufacturer B struggles with packet processing at a '*10 Mbit/s –to-10 Mbit/s*' connection. By flooding the APL switch hardware with packets send via multiple ingress ports, which work at the same link speed as the single egress port responsible for forwarding packets, severe packet loss as well as shutdown of ingress ports of the APL switch are possible side effects.

In conclusion, the APL switch of Manufacturer B does not fulfill all its packet-throughput requirements according to Table 19 regarding the desired packet processing behavior, regardless of working in the constraints of its respective hardware limitations. Thus, said '*10 Mbit/s –to-10 Mbit/s*' connections should be avoided if possible or limited in regard to the PPS of high-priority traffic send via the APL switch to ensure the desired packet processing behavior.

A.5.2.2 Packet processing @ increasing UDP packet count

The following tests have been conducted by increasing UDP traffic through increasing its packet count. The traffic parameters used in these tests are stated in Table 20.

Table 20: Downstream traffic analysis (Alternative 1), Manufacturers A&B – Traffic parameters, increasing UDP traffic @ varying *Packet Count per Cycle*

	UDP real-time data, *single sensor*					
user priority	6					
total packet payload **(TPP)**	46 Byte (data) + 42 Byte (framing/transmission) = 88 Byte					
packet cycle time **(PCT)**	62,5 ms					
Packet Count per Cycle **(PCC)**	**24 · 32 / 24 · 64 / 24 · 128 packets/cycle (Manufacturer A)**			**4 · 32 / 4 · 64 / 4 · 128 packets/burst (Manufacturer B, 10 Mbit/s)**		
Packet data Payload per Cycle **(PPC)**	~34,5 / ~69,0 / ~138,0 kByte/cycle (Manufacturer A)			~5,8 / ~11,5 / ~23,0 kByte/cycle (Manufacturer B)		
total frame payload per cycle **(FPC)**	~68 / ~135 / ~270 kByte/cycle (Manufacturer A)			~11 / ~23 / ~45 kByte/cycle (Manufacturer B)		
Packet Count per Second **(PCS)**	**12.288 / packets/s (Manufacturer A)**	**24.576 packets/s (Manufacturer A)**	**49.152 packets/s (Manufacturer A)**	**2.048 / packets/s (Manufacturer B)**	**4.096 / packets/s (Manufacturer B)**	**8.192 / packets/s (Manufacturer B)**
packet data payload per second **(PPS)**	12.288 p/s · 46 Byte = 565.248 Byte/s **(~552, 0 kByte/s)**	24.576 p/s · 46 Byte = 1.130.496 Byte/s **(~1, 08 MByte/s)**	49.152 p/s · 46 Byte = 2.260.992 Byte/s **(~2, 16 MByte/s)**	2.048 p/s · 46 Byte = 94.208 Byte/s **(~92, 0 kByte/s)**	4.096 p/s · 46 Byte = 188.416 Byte/s **(~184, 0 kByte/s)**	8.192 p/s · 46 Byte = 376.832 Byte/s **(~368, 0 kByte/s)**
total frame payload per second **(FPS)**	12.288 p/s · 88 Byte = 1.081.344 Byte/s **(~1, 03 MByte/s)**	24.576 p/s · 88 Byte = 2.162.688 Byte/s **(~2, 06 MByte/s)**	49.152 p/s · 88 Byte = 4.325.376 Byte/s **(~4, 13 MByte/s)**	2.048 p/s · 88 Byte = 180.224 Byte/s **(~176 kByte/s)**	4.096 p/s · 88 Byte = 360.448 Byte/s **(~352 kByte/s)**	8.192 p/s · 88 Byte = 720.896 Byte/s **(~704 kByte/s)**

The following figures show the packet-throughput behavior of the switch when using the stated in Table 20. Figure 127 to Figure 132 show the packet-throughput behavior of Manufacturer A in upstream direction. Figure 133 to Figure 138 show the packet-throughput behavior of Manufacturer B in upstream direction.

The blue line represents the entire captured UDP traffic send by all field device emulators at ~12.288 … 49.152 packets/s for Manufacturer A and ~1.024 … 4.096 packets/s for Manufacturer B. The brown line shows the UDP traffic from one field device emulator at 512 … 2.048 packets/s for Manufacturer A and 256 … 1.024 packets/s for Manufacturer B.

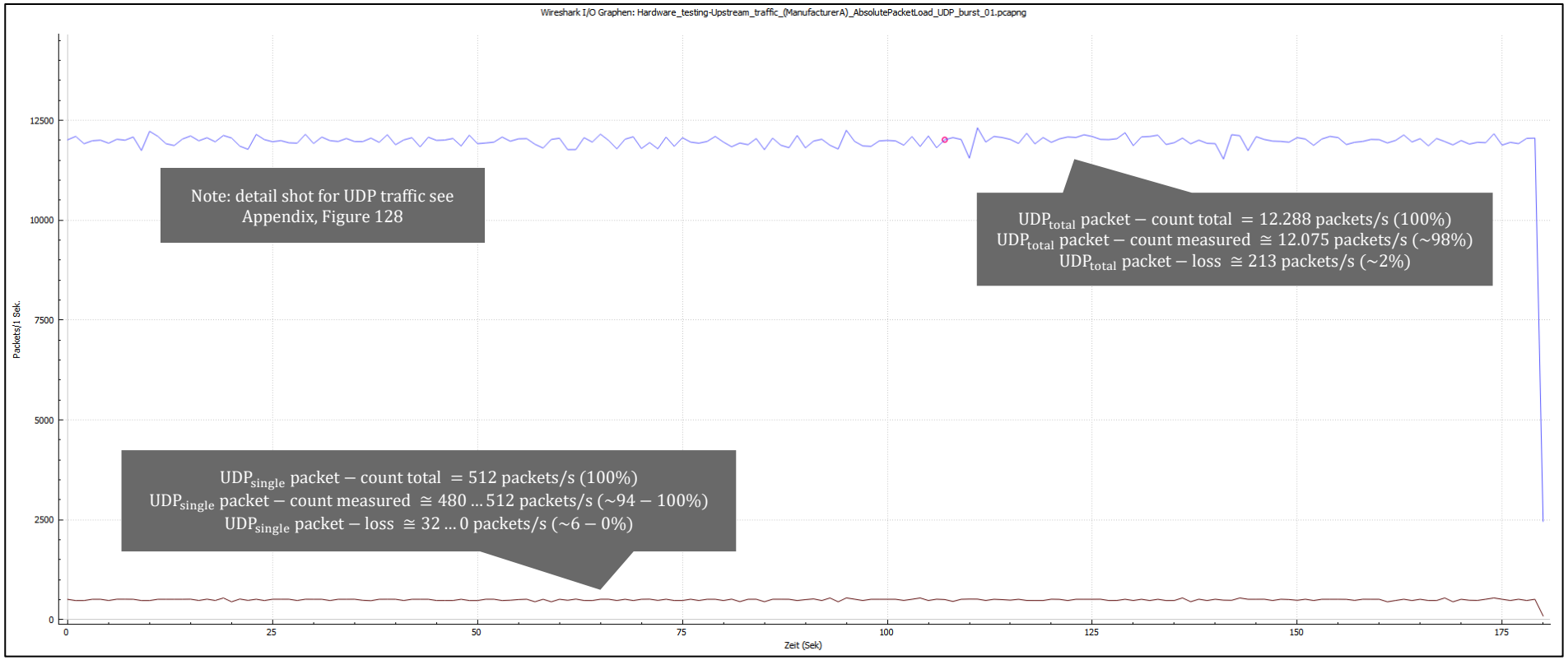

Figure 127: Upstream traffic analysis **@ increasing UDP packet count (Alternative 2), Manufacturer A** - Measurement results (24·32 UDP packets / 62,5 ms) - total packet count

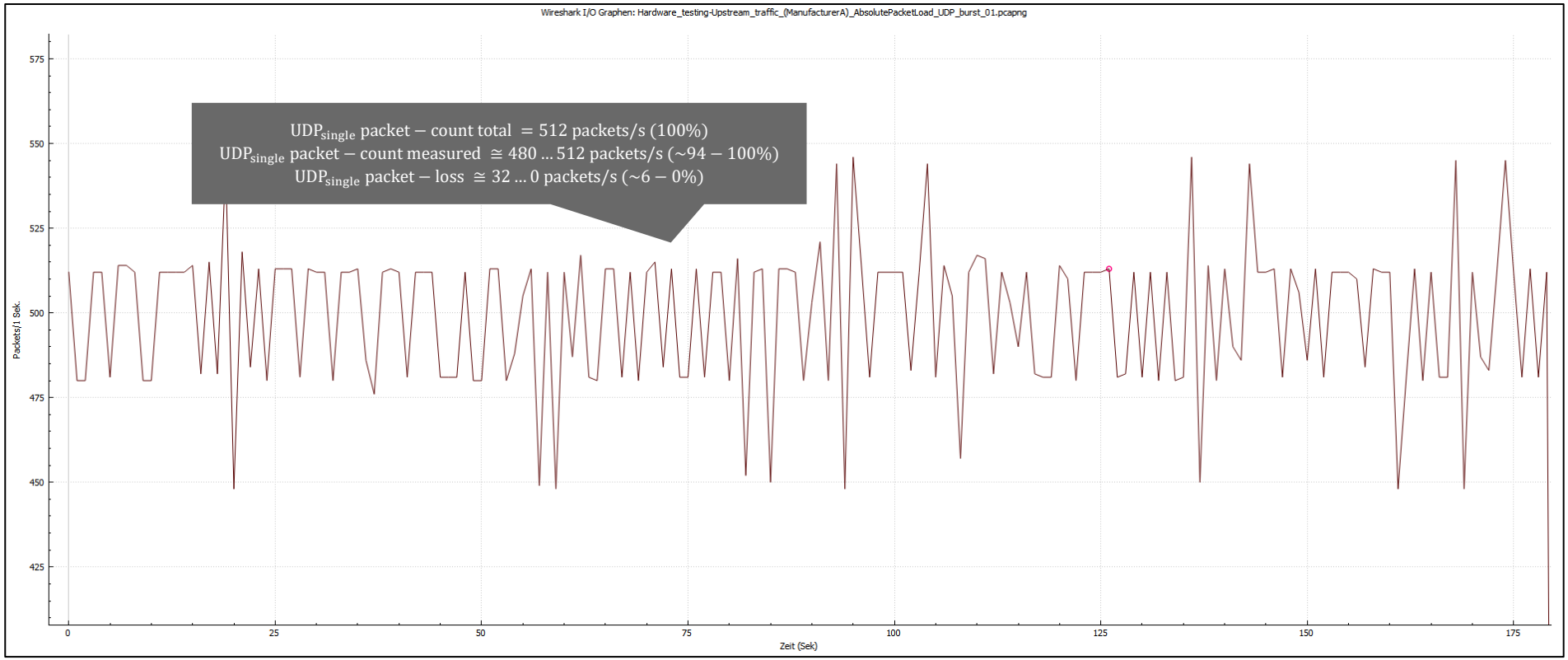

Figure 128: Upstream traffic analysis **@ increasing UDP packet count (Alternative 2), Manufacturer A** - Measurement results (24·32 UDP packets / 62,5 ms) – single source packet count

Figure 127 shows that UDP traffic has a packet loss of 2% at a total PCS (‘*Packet Count per Second*’) of $12.288\ packets/s$. Said packet loss of UDP packets happens due to the same phenomenon which was already described in the measurement summary of chapter A.4.1. Hence, the packet prioritization of the APL switch works correctly.

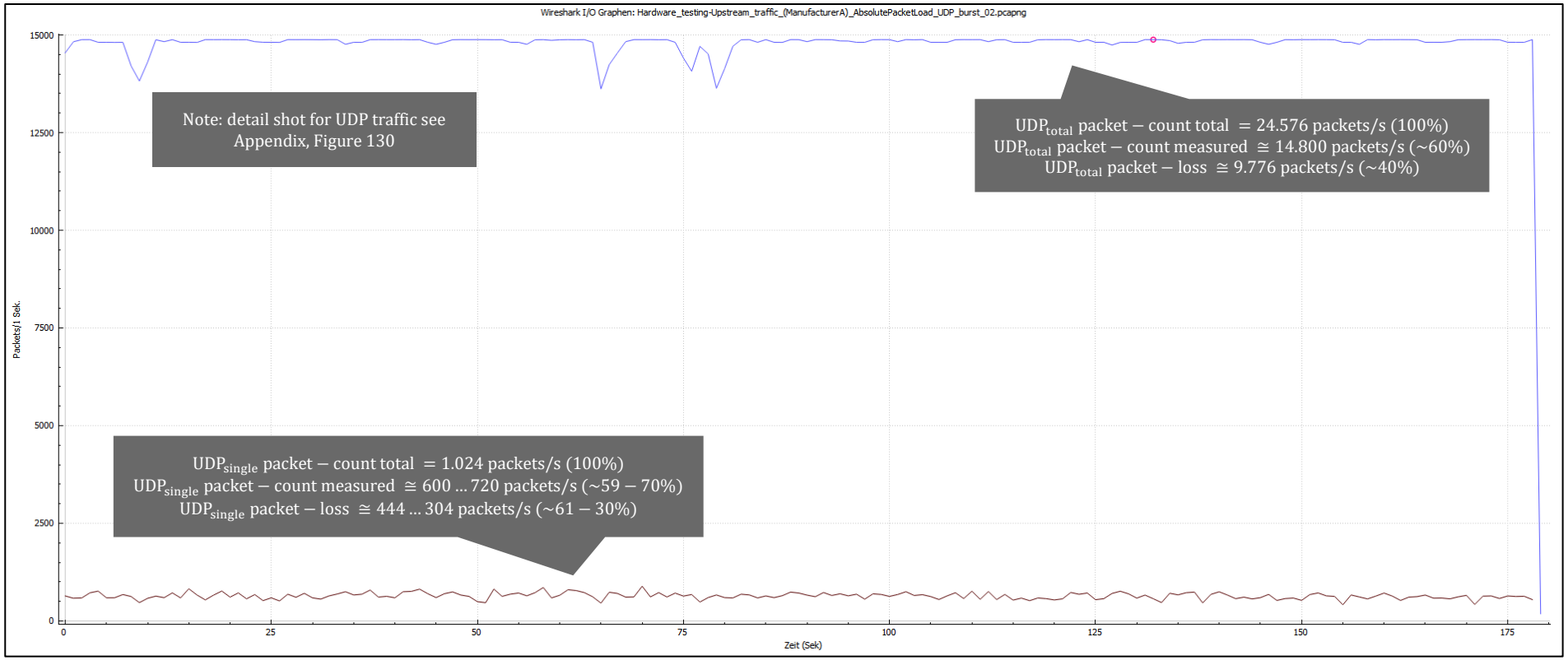

Figure 129: Upstream traffic analysis **@ increasing UDP packet count (Alternative 2), Manufacturer A** - Measurement results (24·64UDP packets / 62,5 ms) – total packet count

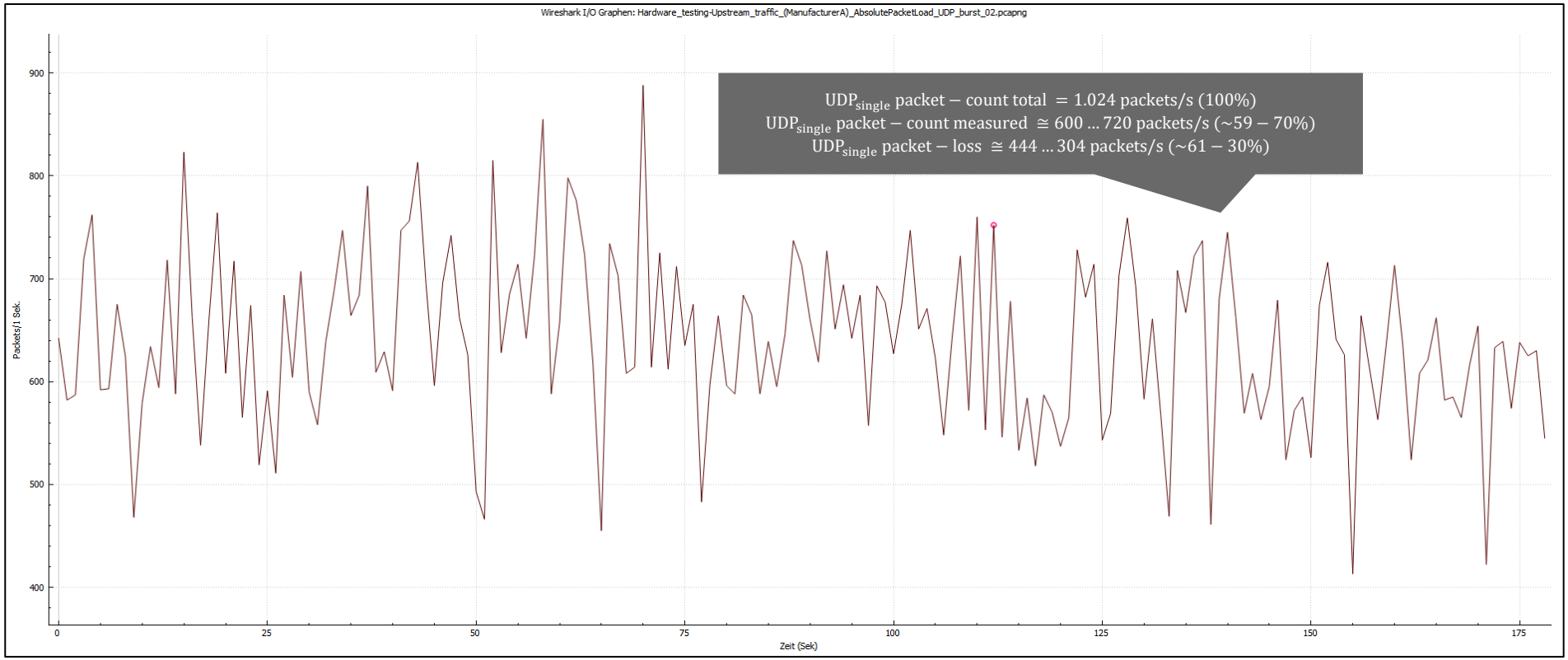

Figure 130: Upstream traffic analysis **@ increasing UDP packet count (Alternative 2), Manufacturer A** - Measurement results (24·64UDP packets / 62,5 ms) – single source packet count

Figure 129 shows that by doubling the total PCS (‘*Packet Count per Second*’) of UDP traffic from ~12.288 packets/s to ~24.576 packets/s, packet loss increases from 2% to 60%. Said packet loss happens due to exceeding the ‘*dataRate*’ limit of the APL 10 Mbit/s trunk line (~1,19 Mbyte/s), connecting the APL switch with its respective field devices reaching said limitation due to their combined packet load of UDP traffic:

$$dataRate_{\mathrm{A\&B,max}} = 10\frac{Mbit}{s} = 1.250.000\ Byte \cong 1{,}19\ MByte/s \qquad (65)$$

$$\frac{packet_{\mathrm{load}}}{s} = PPS_{\mathrm{type}} \cdot packet_{\mathrm{size,type}} \qquad (66)$$

$$\left.\frac{packet_{\text{load}}}{s}\right|_{\text{total}} = PPS_{\text{TCP}} \cdot packet_{\text{size,TCP}} + PPS_{\text{UDP}} \cdot packet_{\text{size,UDP}} \cdot x_{\text{devices}} \quad (67)$$

$$= 14.800 \frac{packets}{s} \cdot 88\ Byte = 1.302.400\ Byte \cong 1{,}24\ MByte$$

Hence, the packet prioritization of the APL switch works correctly.

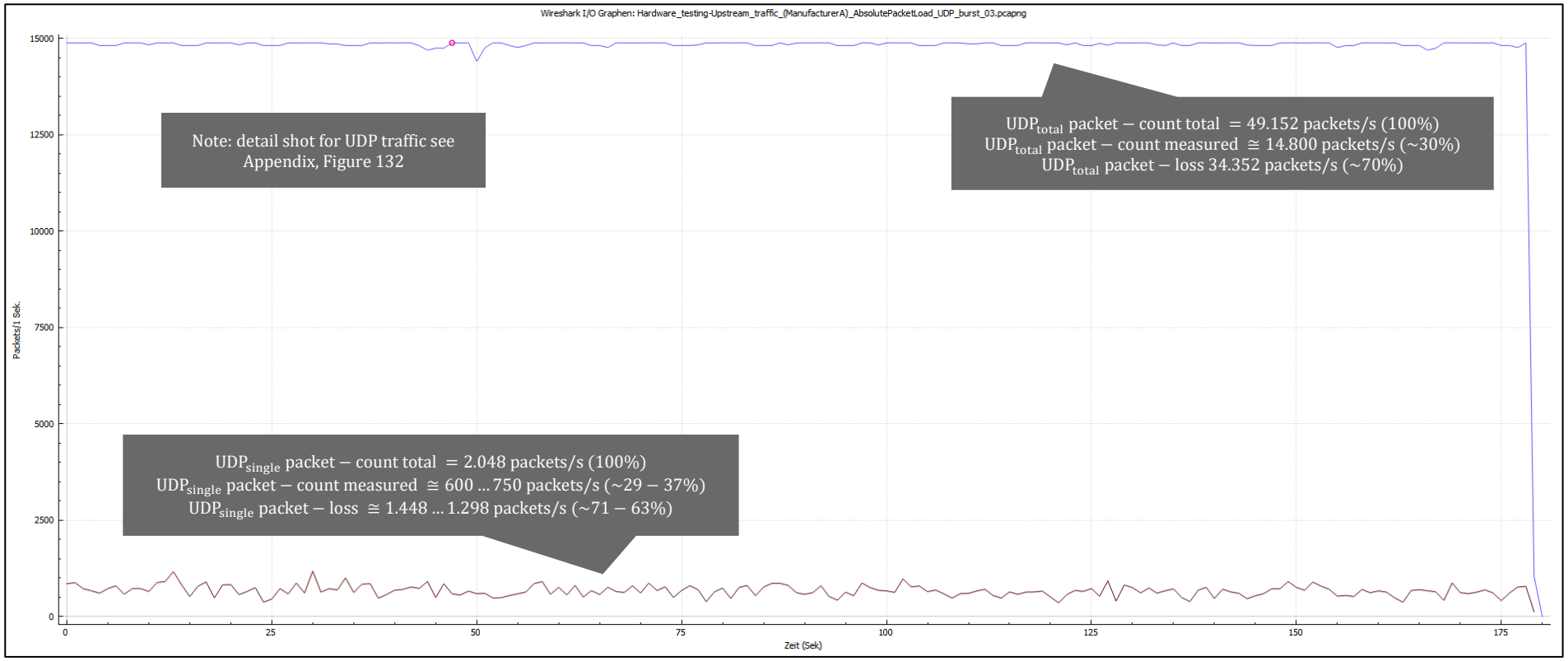

Figure 131: Upstream traffic analysis **@ increasing UDP packet count (Alternative 2), Manufacturer A** - Measurement results (24·128UDP packets / 62,5 ms) – total packet count

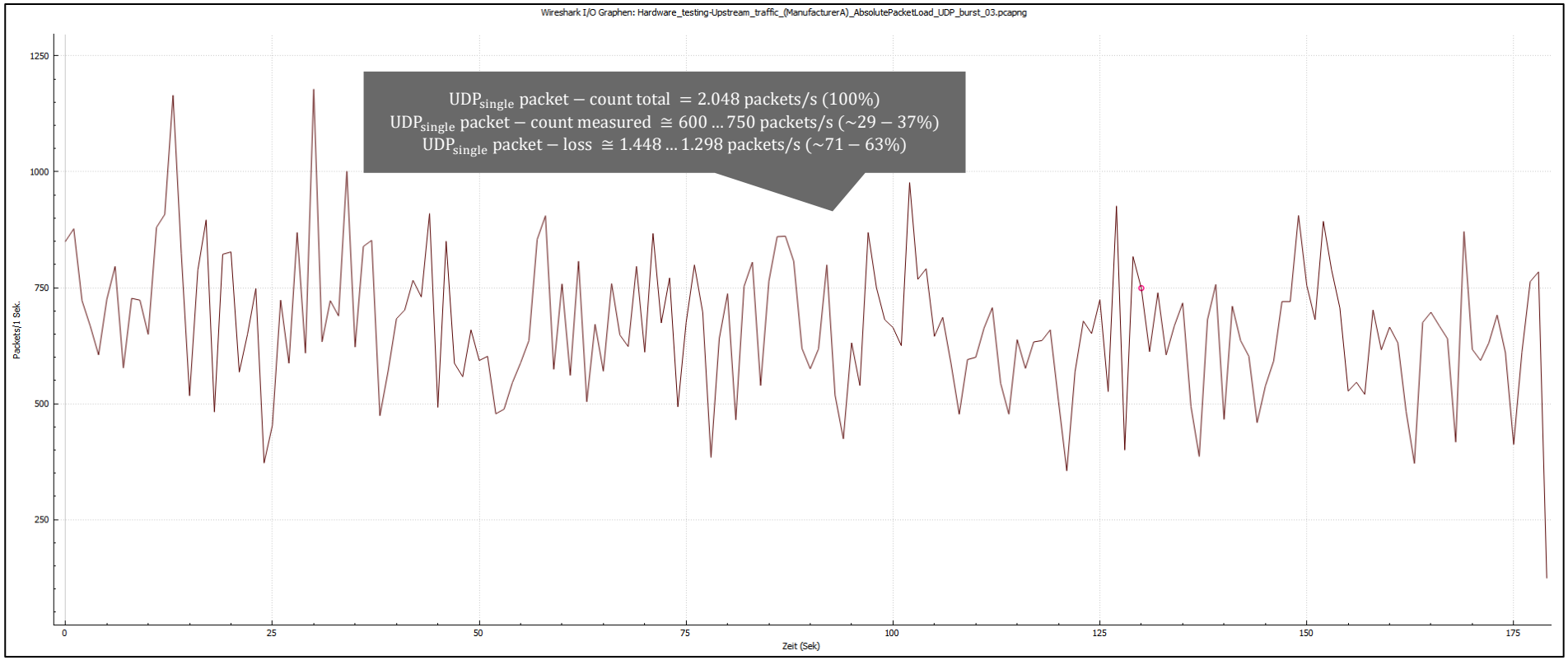

Figure 132: Upstream traffic analysis **@ increasing UDP packet count (Alternative 2), Manufacturer A** - Measurement results (24·128UDP packets / 62,5 ms) – single source packet count

Figure 131 shows that by quadrupling the total PCS (‘*Packet Count per Second*’) of UDP traffic from ~12.288 packets/s to ~49.152 packets/s, packet loss increases from 2% to 70% The same phenomenon which was referred to in the previous Figure 127 is also apparent in Figure 131, but in a more pronounced form. Hence, the packet prioritization of the APL switch works correctly.

Summary: (Figure 127 to Figure 132): The measurements show, that stable packet-throughput of simultaneous incoming traffic, by gradual increase of UDP traffic up to ~8.192 packets/s, is not always ensured.

Exceeding the '*dataRate*' limit of the APL trunk line, regarding packet load, leads to discarding of excessive packets forwarded by the egress port outside of the switch (see chapter 5.3.1). However, the PCS ('*Packet Count per Second*') of UDP traffic shows that the APL switch tries to forward as much UDP traffic as possible.

In conclusion, the APL switch of Manufacturer A fulfills its packet-throughput requirements according to Table 20 regarding the desired packet processing behavior, regardless of working outside the constraints of its respective hardware limitations.

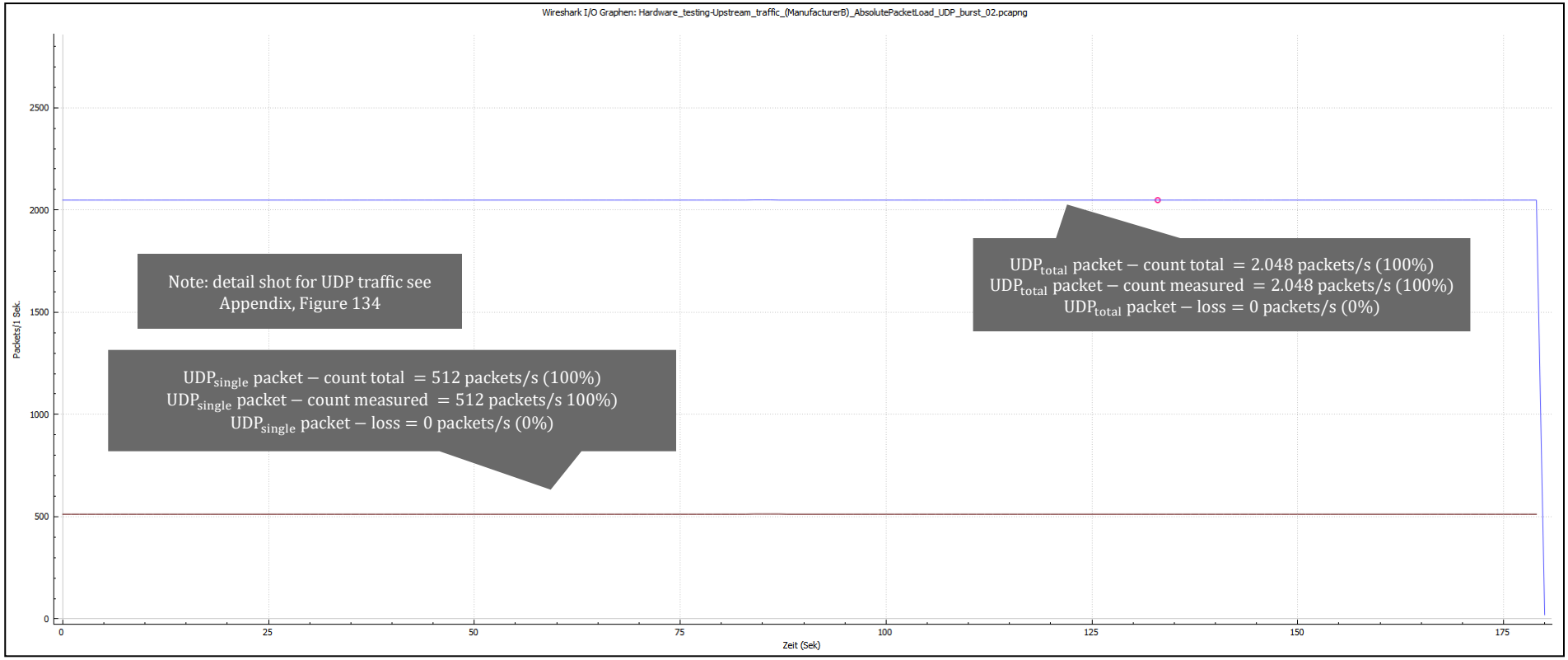

Figure 133: Upstream traffic analysis **@ increasing UDP packet count (Alternative 2), Manufacturer B** - Measurement results (4·32 UDP packets / 62,5 ms) – total packet count

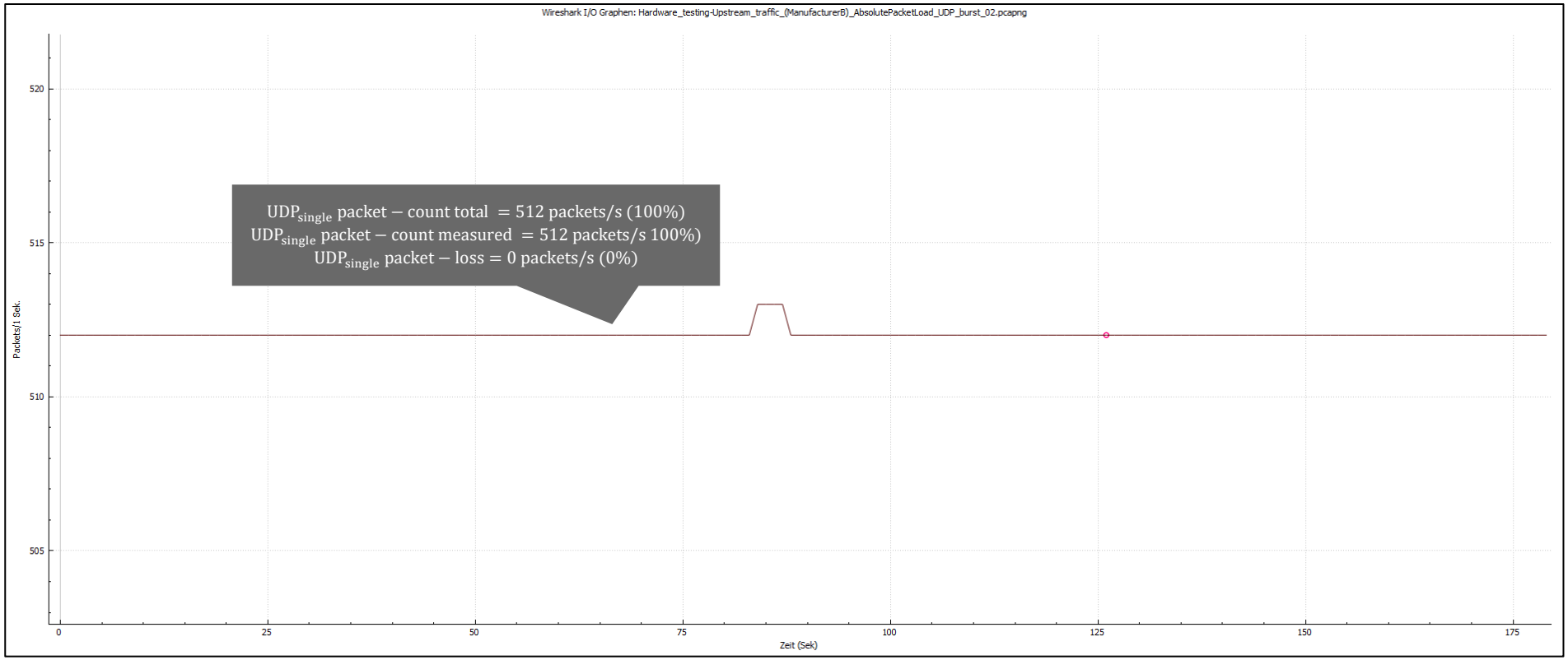

Figure 134: Upstream traffic analysis **@ increasing UDP packet count (Alternative 2), Manufacturer B** - Measurement results (4·32 UDP packets / 62,5 ms) – single source packet count

Figure 133 shows that UDP traffic does not struggle with packet loss at a total PCS ('*Packet Count per Second*') of 2.048 packets/s . Hence, all UDP packets are forwarded successfully by the APL switch.

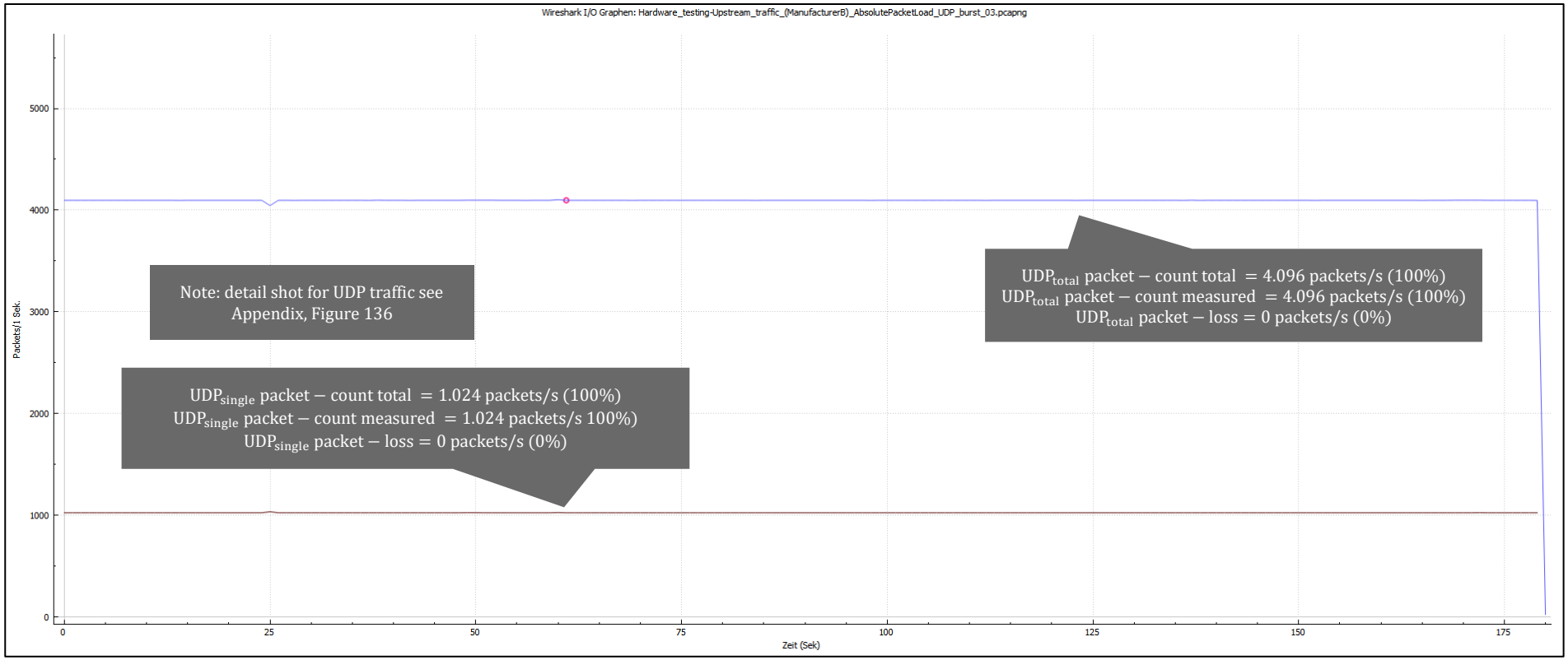

Figure 135: Upstream traffic analysis **@ increasing UDP packet count (Alternative 2), Manufacturer B** - Measurement results (4·64 UDP packets / 62,5 ms) – total packet count

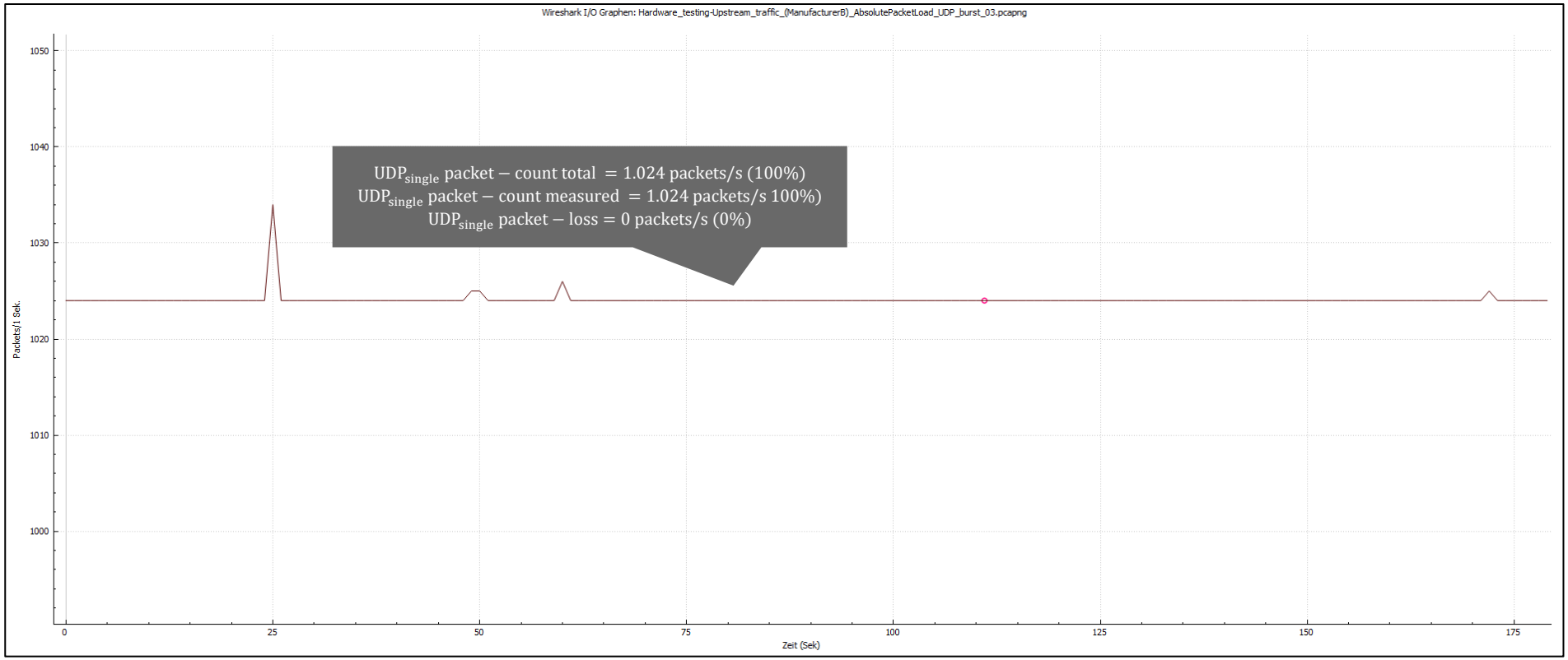

Figure 136: Upstream traffic analysis **@ increasing UDP packet count (Alternative 2), Manufacturer B** - Measurement results (4·64 UDP packets / 62,5 ms) – single source packet count

Figure 135 shows that by doubling the total PCS (‘*Packet Count per Second*’) of UDP traffic from ~ 2.048 packets/s to ~ 4.096 packets/s, packet loss still does not occur. Hence, all packets are forwarded successfully by the APL switch.

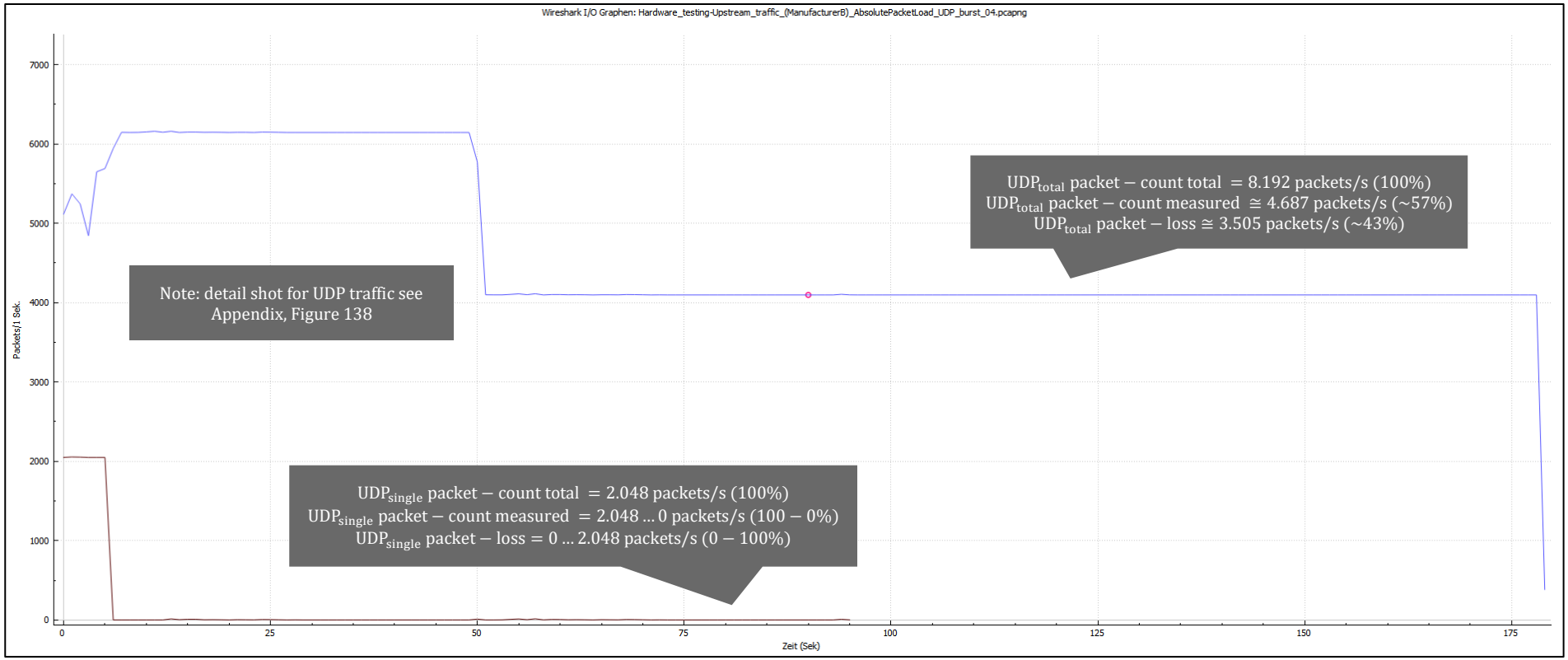

Figure 137: Upstream traffic analysis **@ increasing UDP packet count (Alternative 2), Manufacturer B** - Measurement results (4·128 UDP packets / 62,5 ms) – total packet count

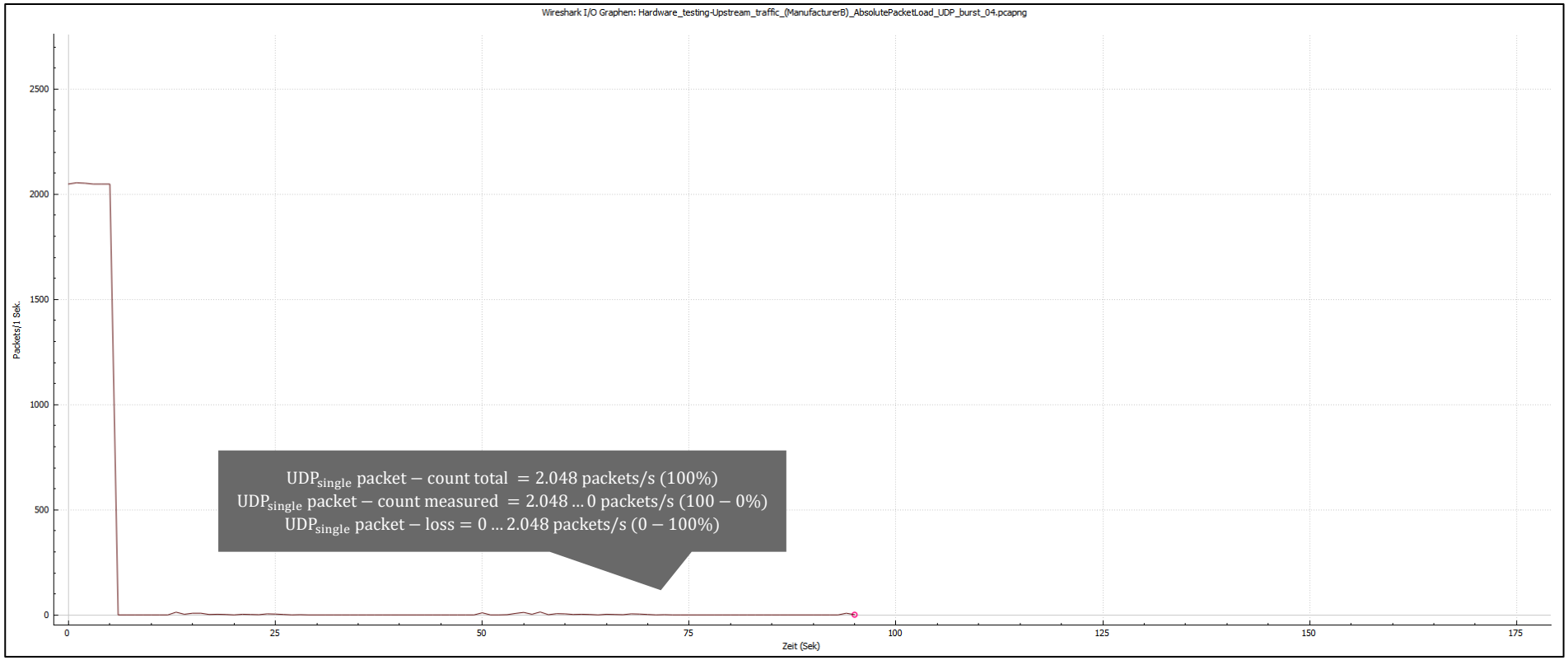

Figure 138: Upstream traffic analysis **@ increasing UDP packet count (Alternative 2), Manufacturer B** - Measurement results (4·128 UDP packets / 62,5 ms) – single source packet count

Figure 137 shows that by quadrupling the total PCS ('*Packet Count per Second*') of UDP traffic from ~2.048 packets/s to ~8.192 packets/s, packet loss increases from 0% to 43%. Said packet loss of UDP packets happens due to the same phenomenon which was already described in the measurement summary of chapter A.4.1. Hence, the packet processing of the APL switch does no longer work as intended.

Summary: (Figure 133 to Figure 138): The measurements show, that stable packet-throughput of simultaneous incoming traffic, by gradual increase of UDP traffic up to ~ 8.192 packets/s, is not always ensured.

If the packet count rises above ~ 4.096 packets/s similar side effects as described in the measurement summary of chapter A.5.2.1 are repeating, causing packet loss due to increasing packet-throughput instabilities as well as entire shutdowns of Ethernet bridge ports inside the switch hardware. In conclusion, the APL switch of Manufacturer B does not fulfill all its packet-throughput requirements according to Table 20 regarding the desired packet processing behavior, regardless of working in the constraints of its respective hardware limitations.

Thus, said '*10 Mbit/s –to-10 Mbit/s*' connections should be avoided if possible or limited in regard to the PPS of high-priority traffic send via the APL switch to ensure the desired packet processing behavior.