Enabling Passive Measurement of Zoom Performance in Production Networks

Oliver Michel

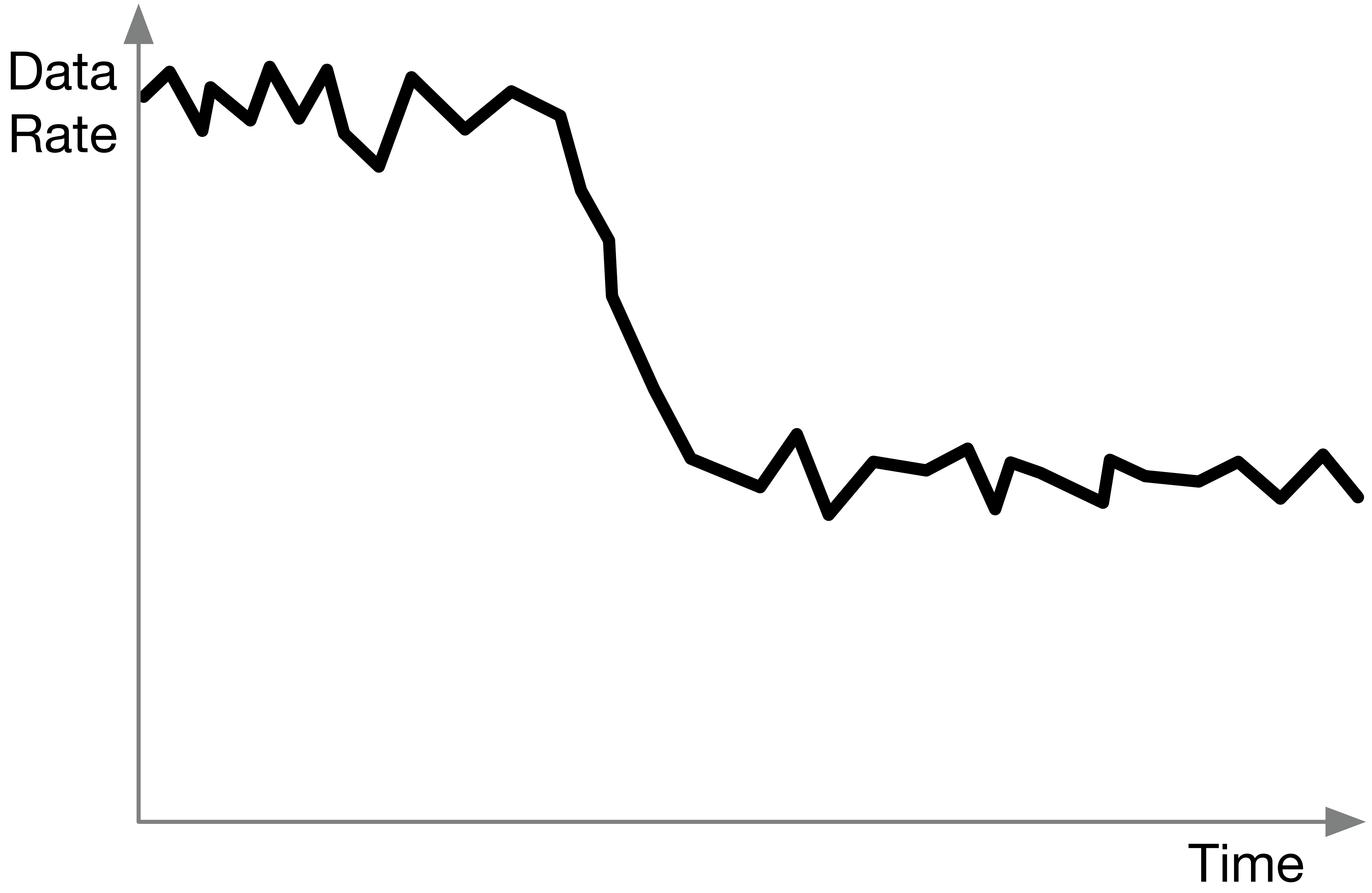
NANOG 87, Atlanta, GA, February 15th, 2023.

PRINCETON UNIVERSITY

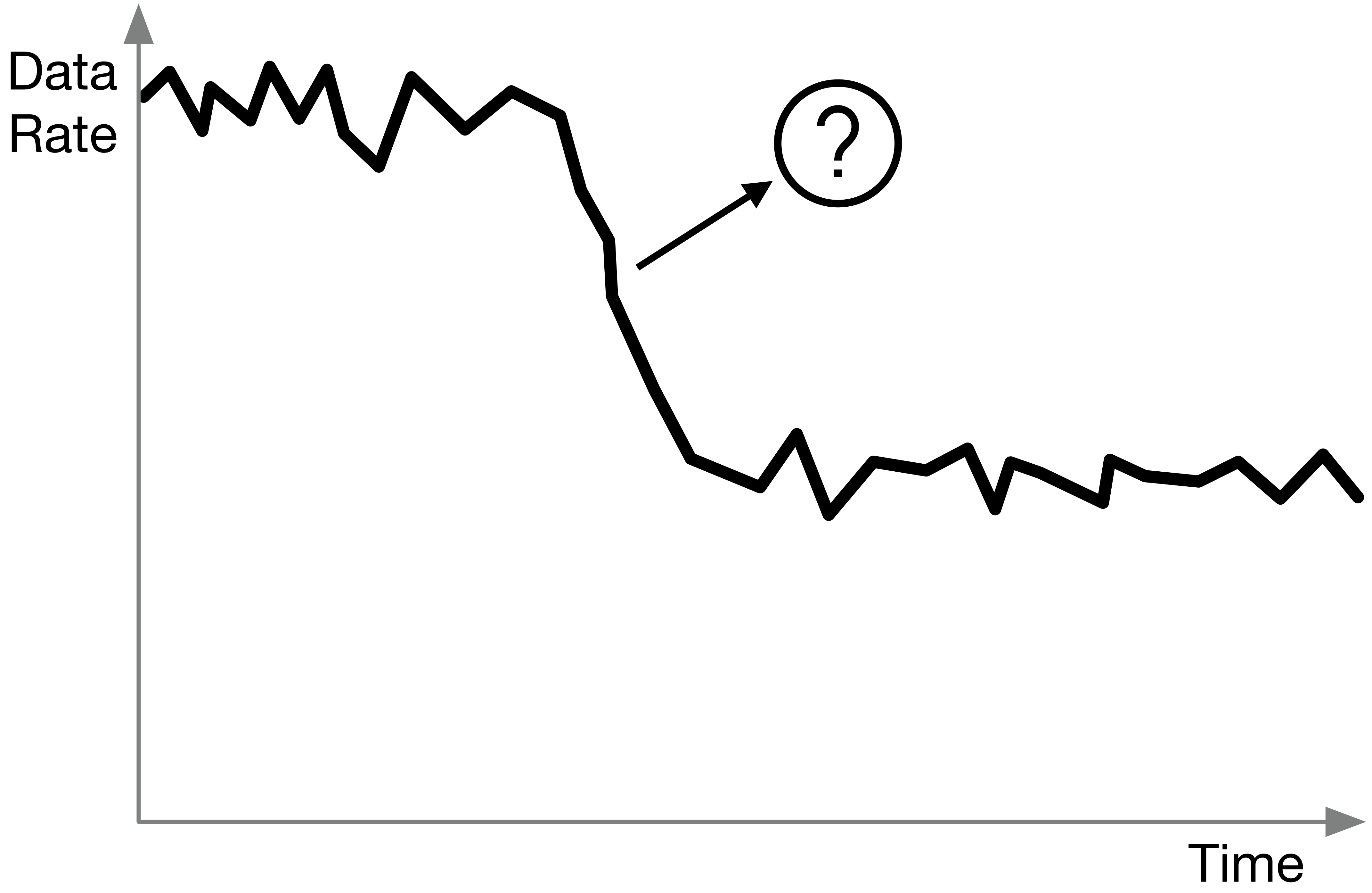
Video Conferencing

- Essential application across most industries
- Relevant to research community
- Relevant to network operators

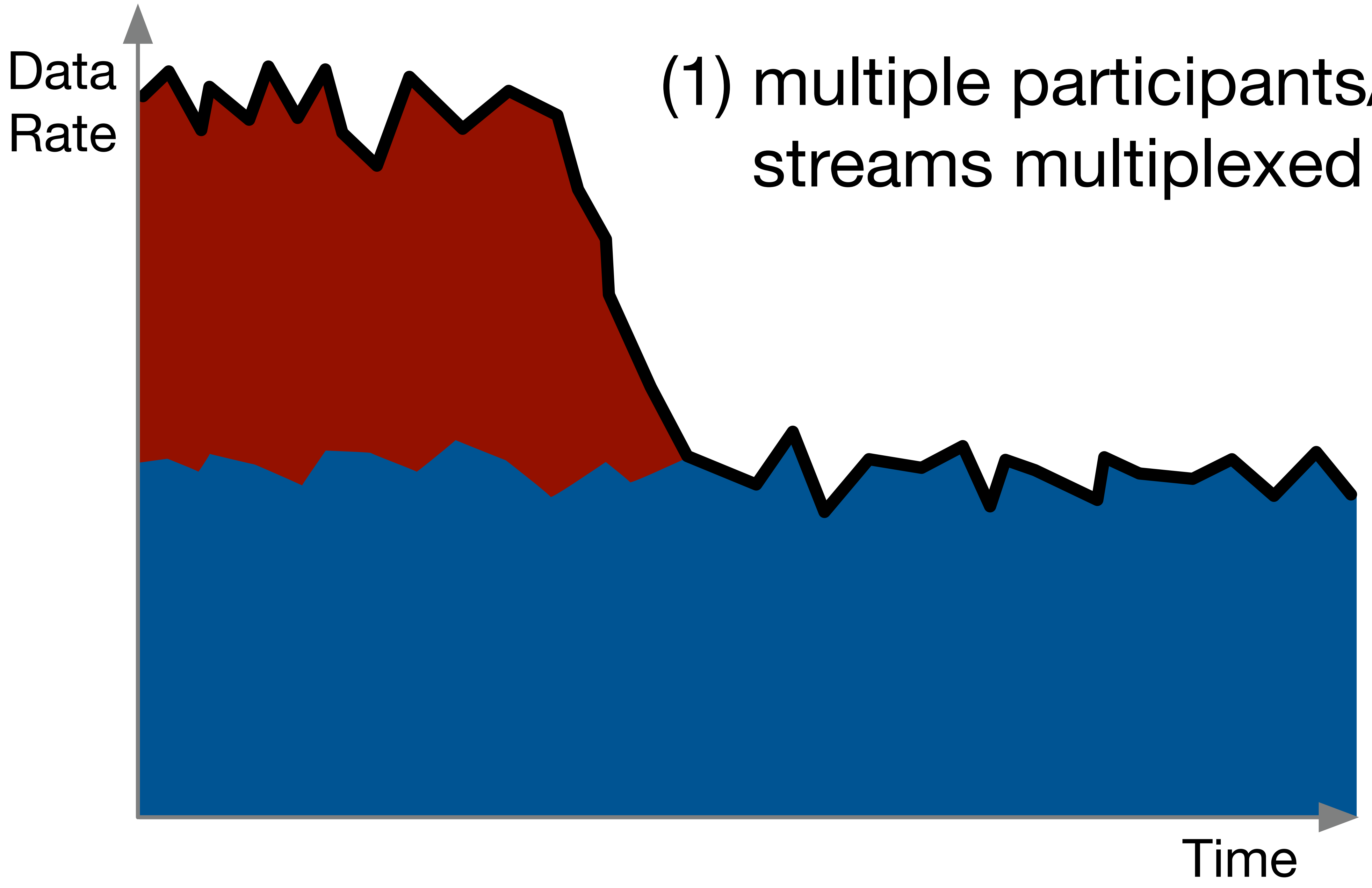
Measuring Video-Conferencing Applications is Hard



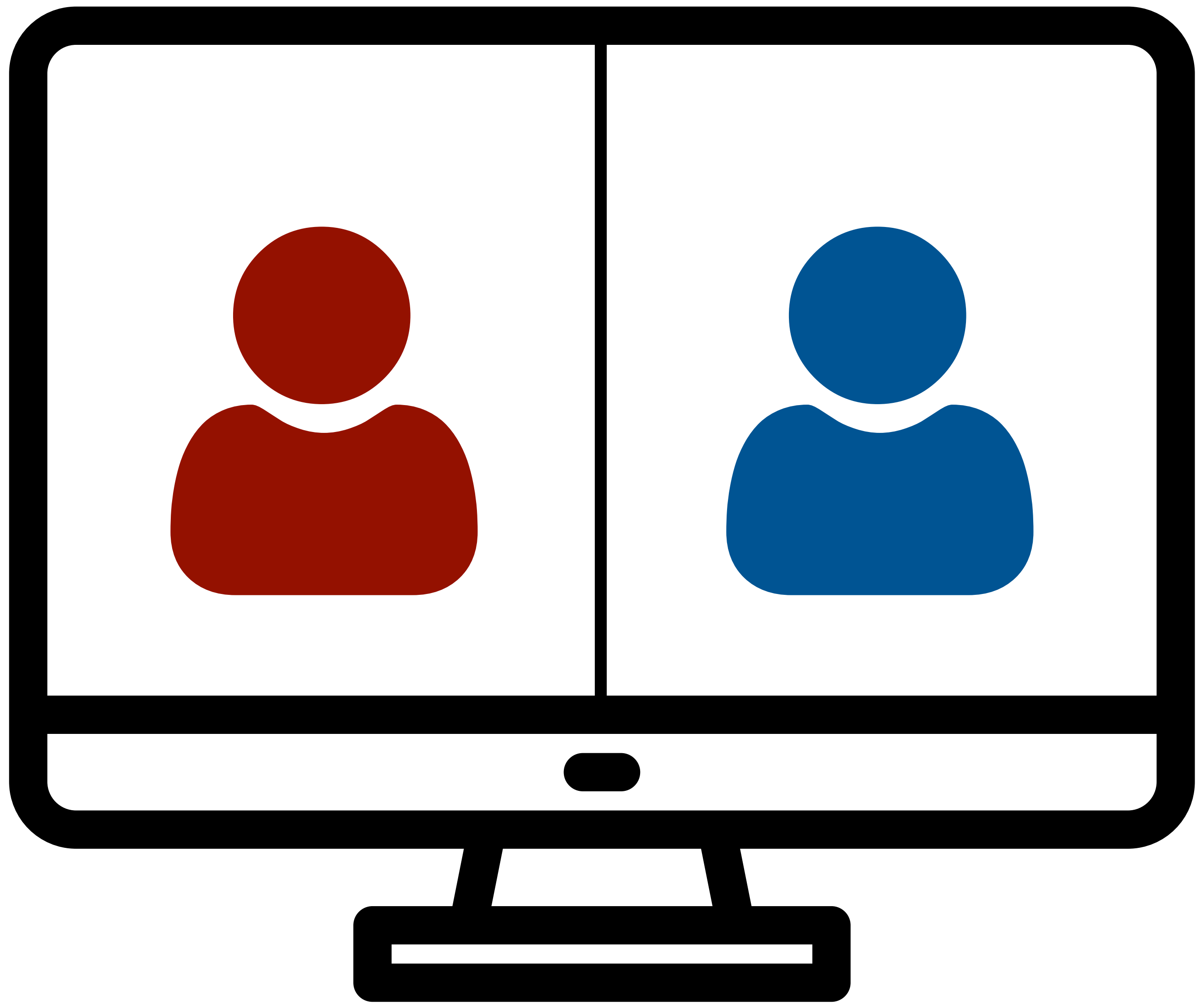
Measuring Video-Conferencing Applications is Hard



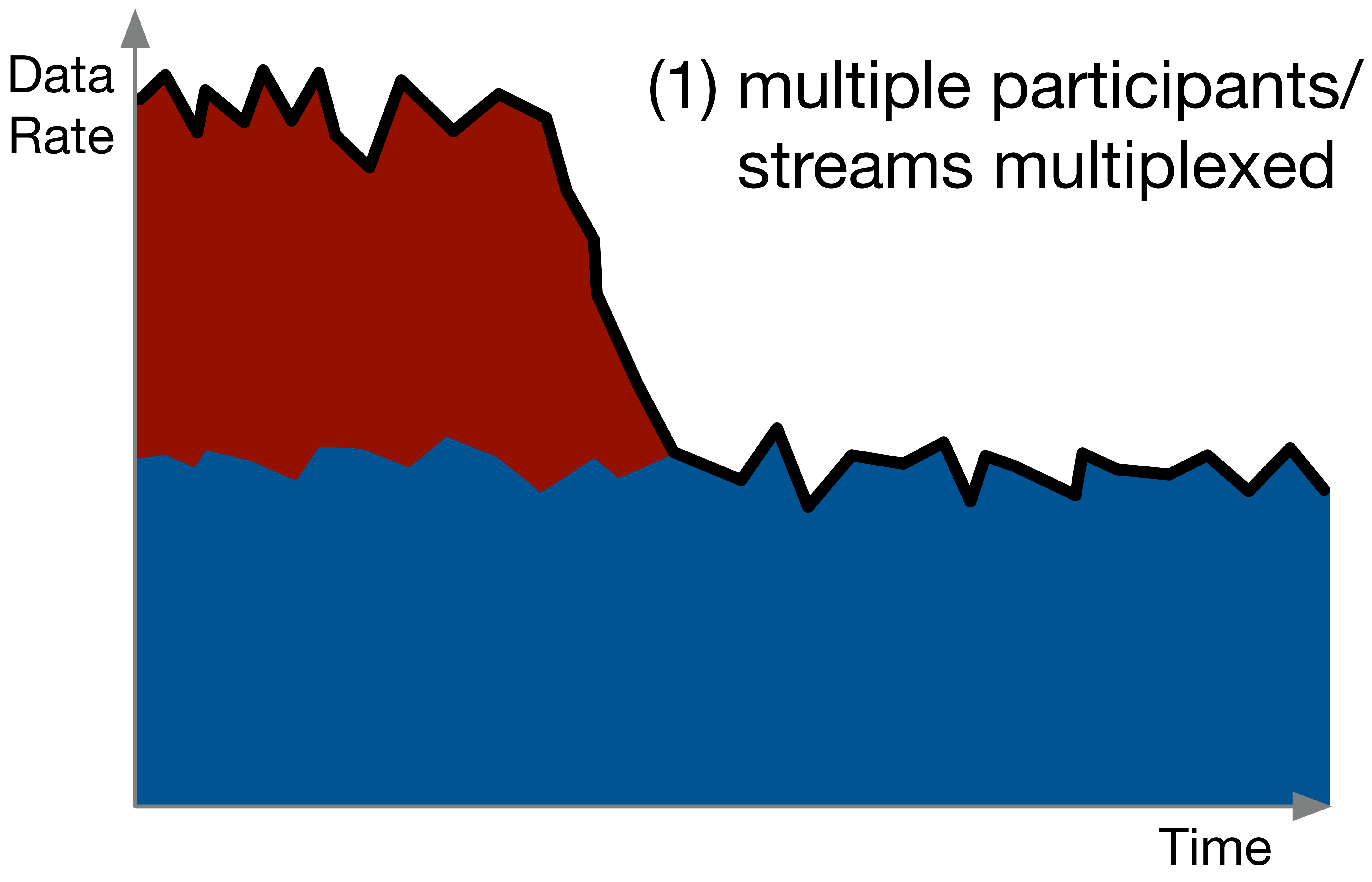
Measuring Video-Conferencing Applications is Hard



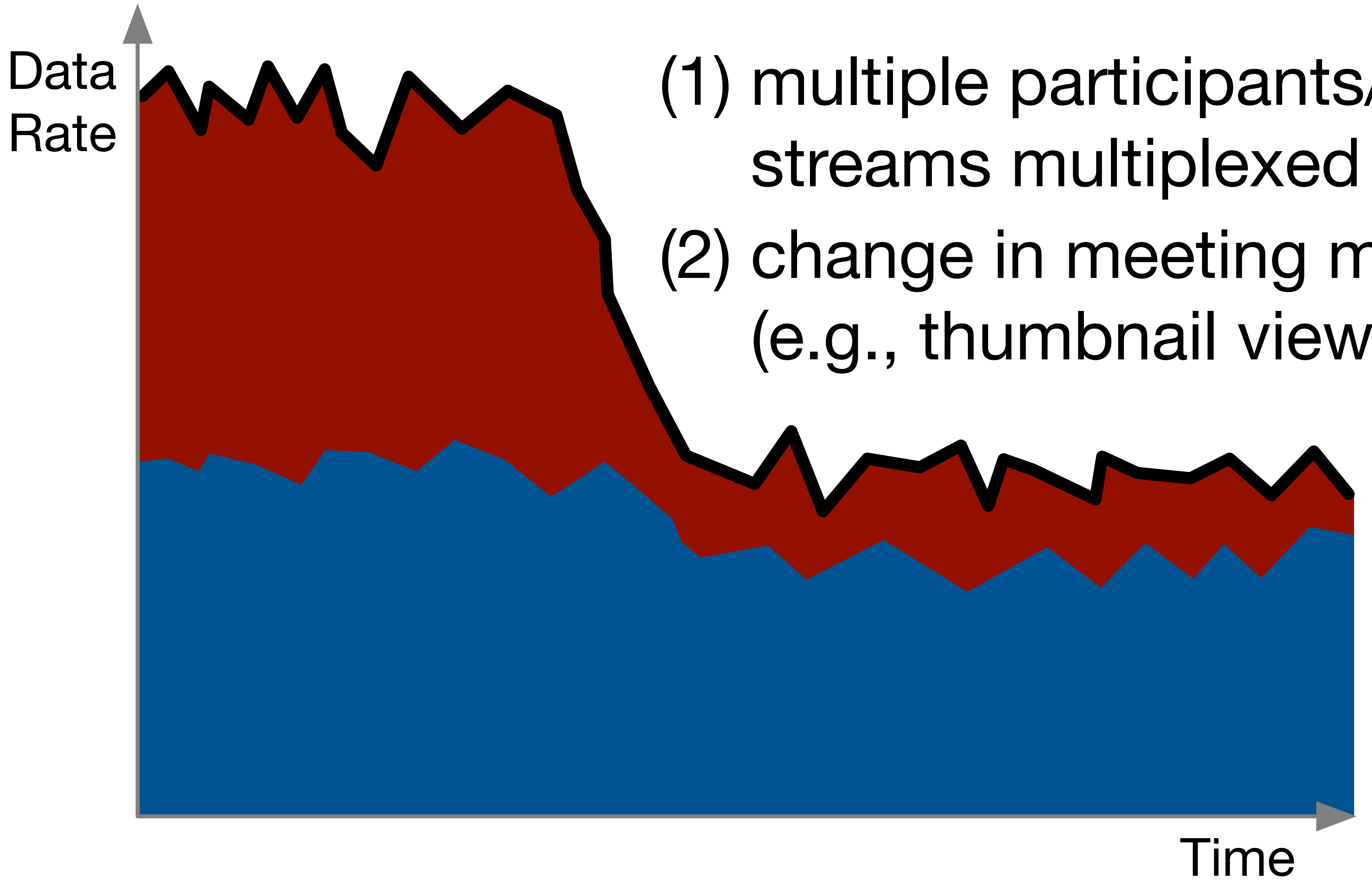
(1) multiple participants/
streams multiplexed



Measuring Video-Conferencing Applications is Hard

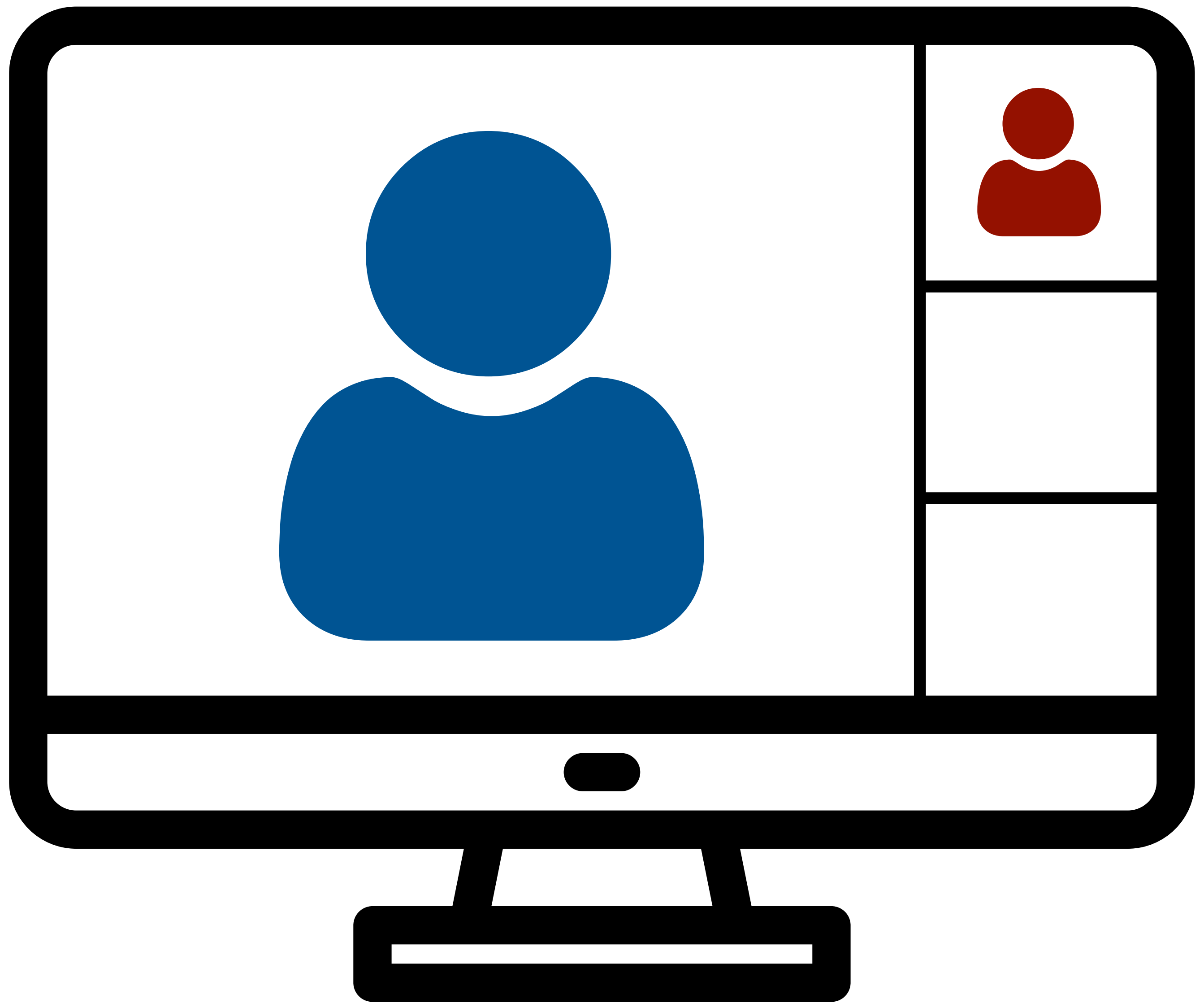


Measuring Video-Conferencing Applications is Hard

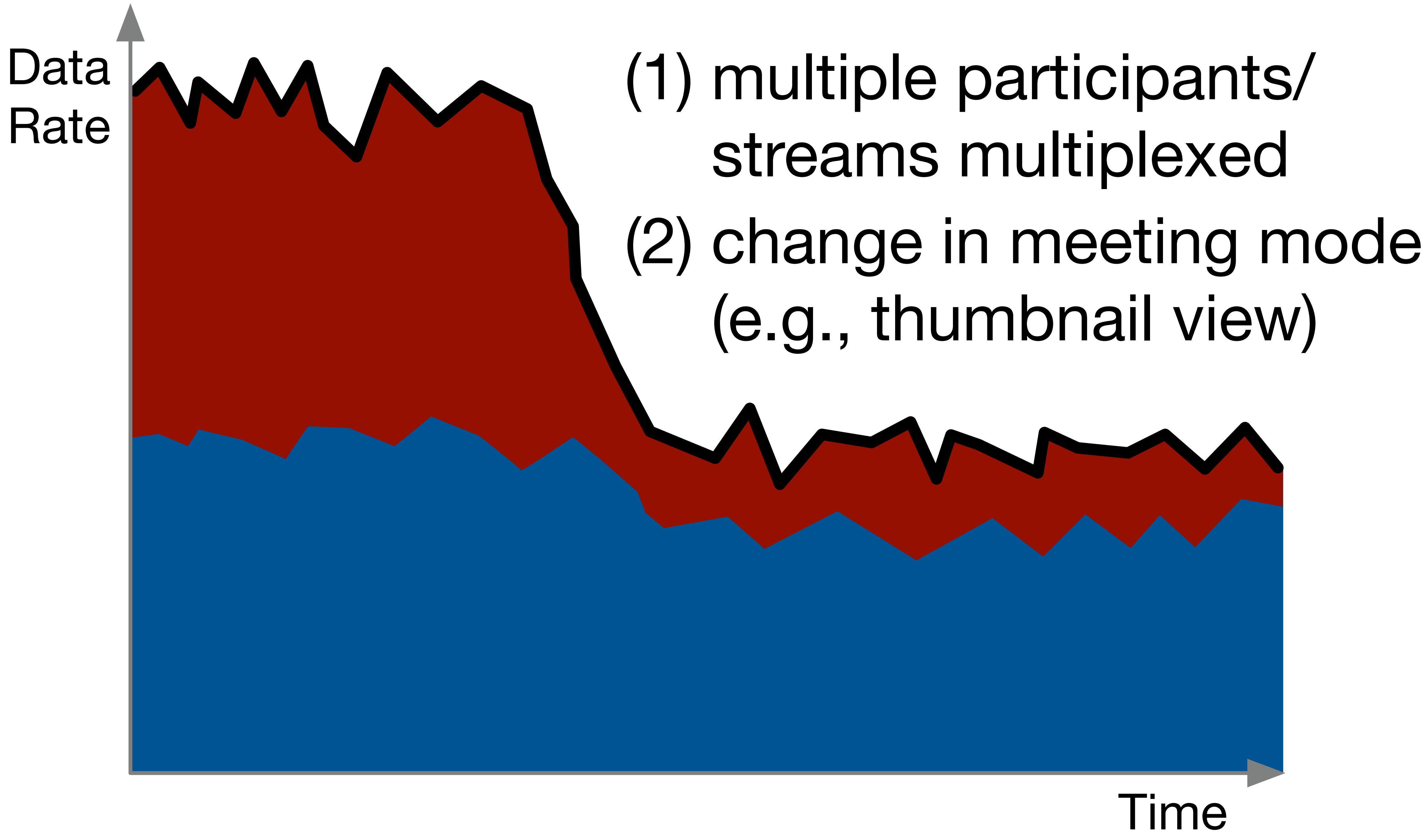


(1) multiple participants/ streams multiplexed

(2) change in meeting mode (e.g., thumbnail view)



Measuring Video-Conferencing Applications is Hard

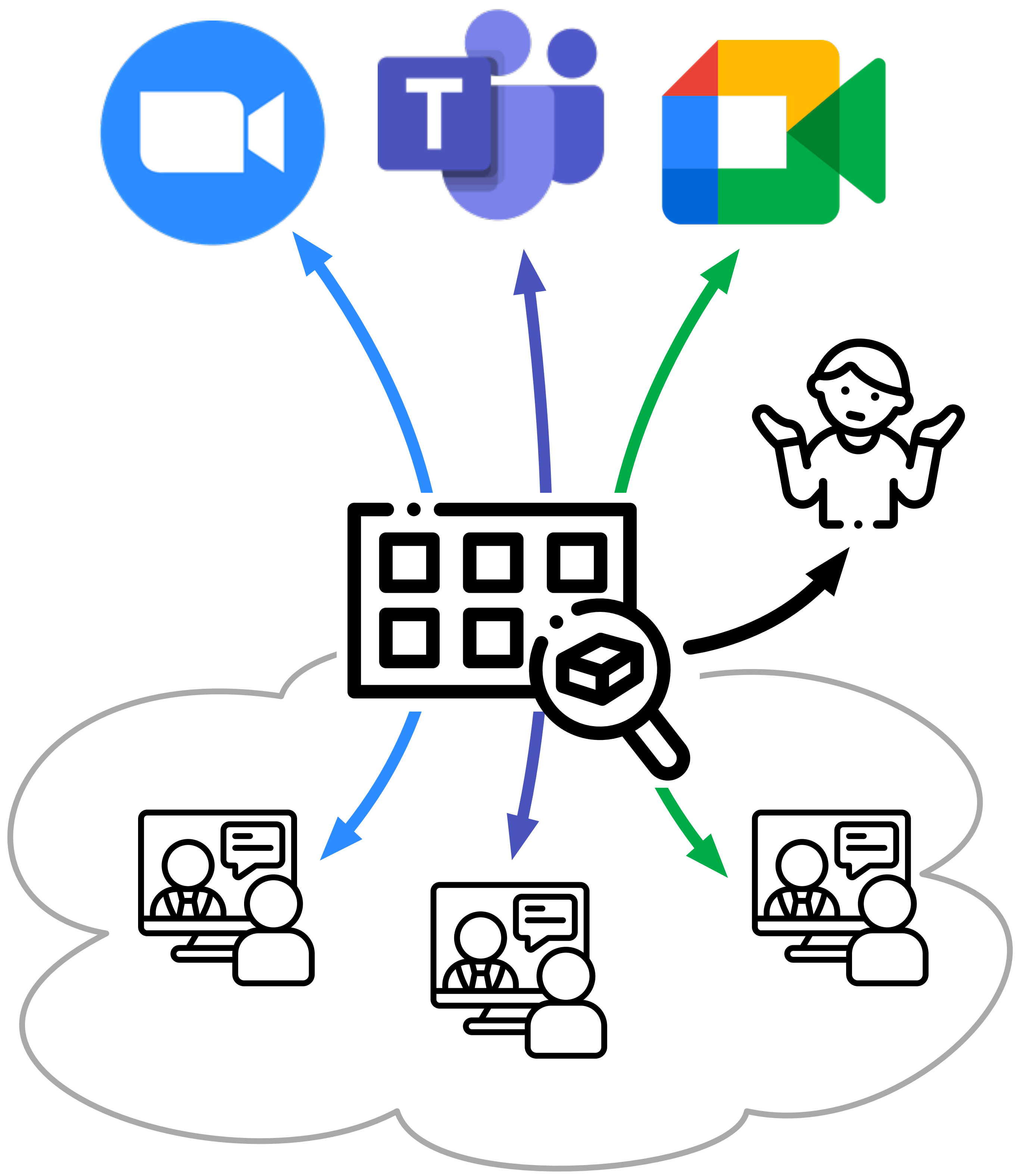


Need to dig deeper to extract meaningful metrics

Latency
Jitter
Loss & Retransmissions
Out-of-order packets
Frame rate & size
Media bit rate
Meeting composition

Measuring Video-Conferencing Applications is Hard

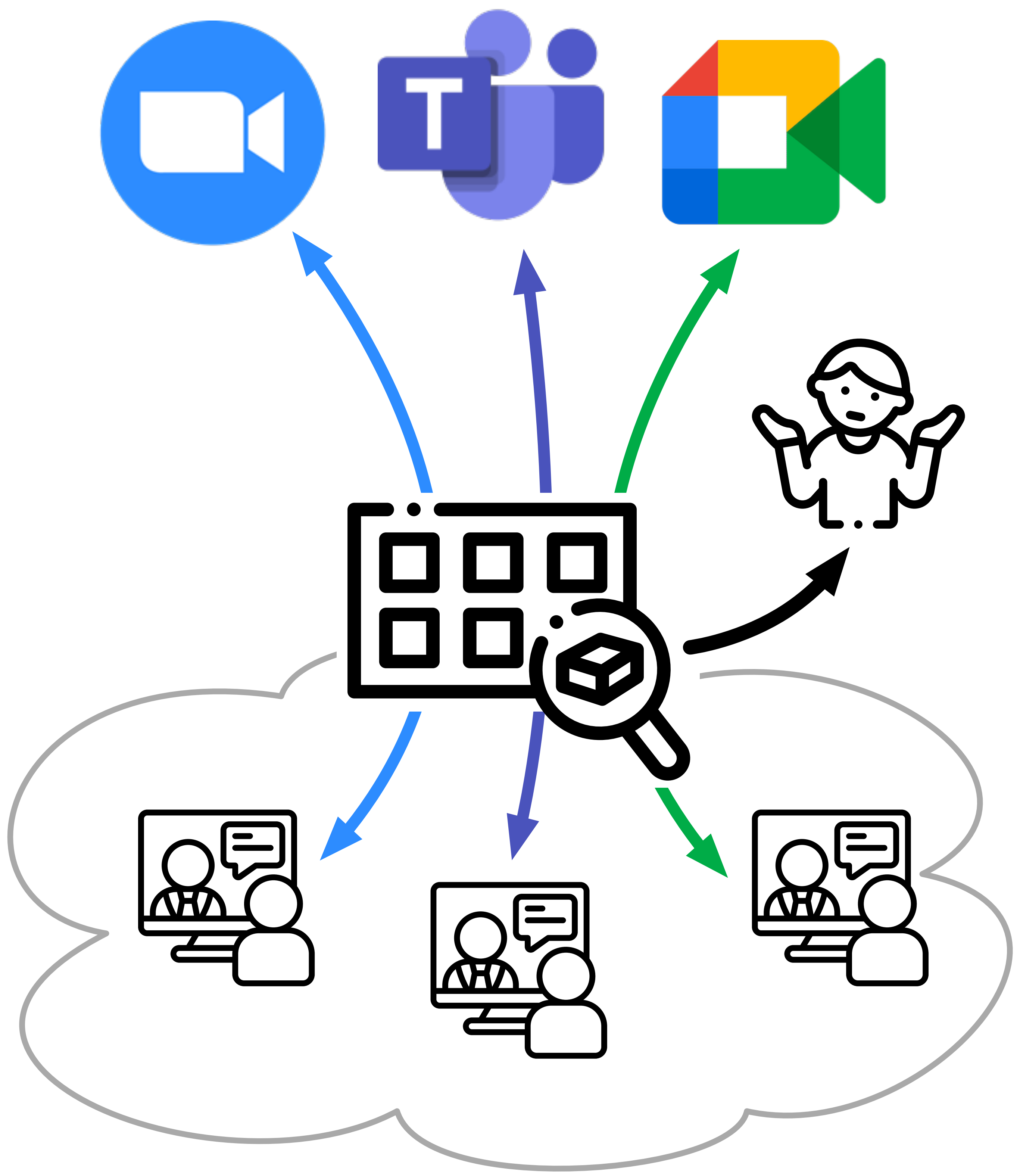
Problem: **Hard for researchers and operators to extract useful metrics from traffic**



Measuring Video-Conferencing Applications is Hard

Problem: **Hard for researchers and operators to extract useful metrics from traffic**

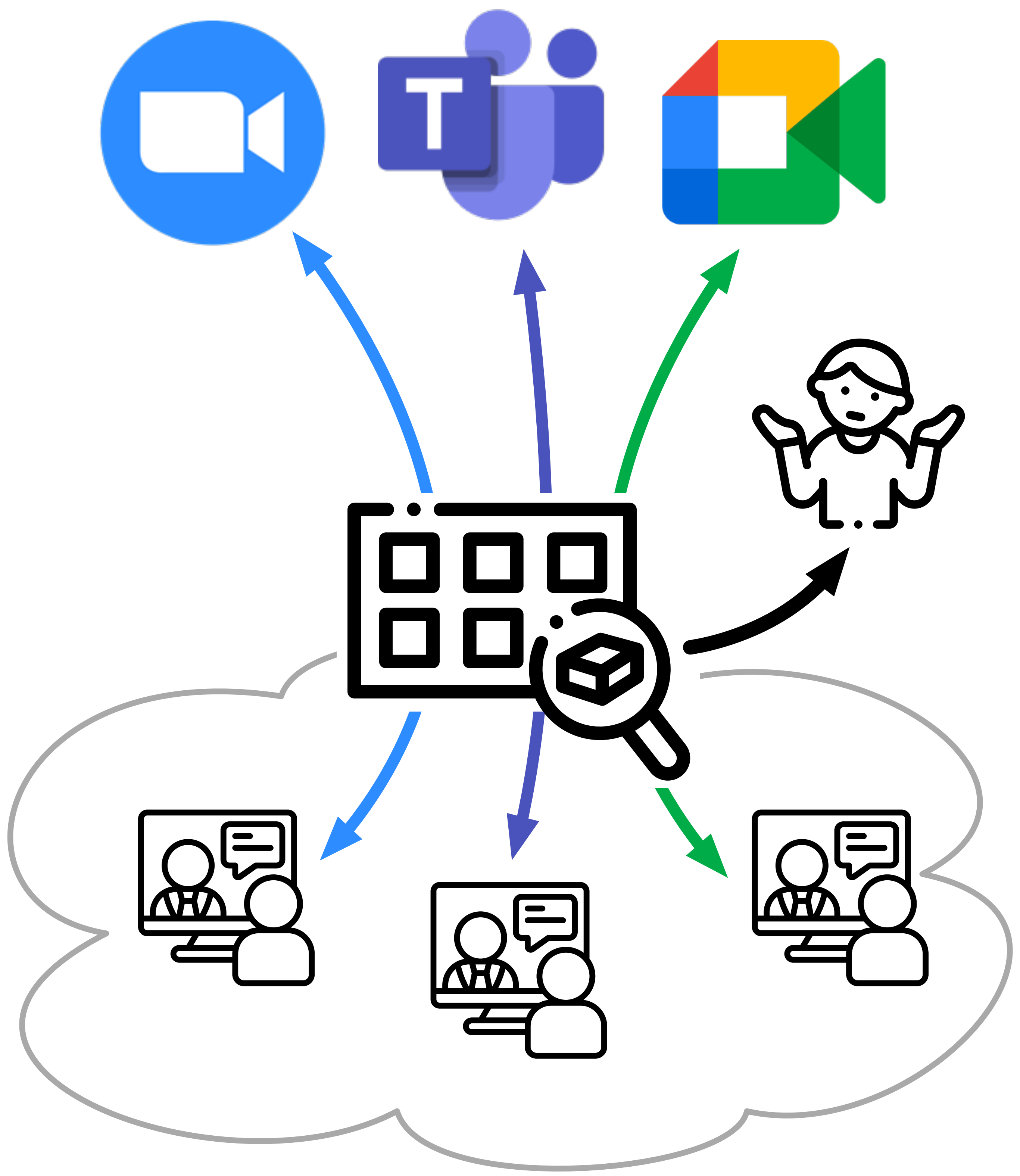
- Quality/performance metrics not directly observable from traffic
- Application-provided metrics only useful for small-scale, controlled experiment



Measuring Video-Conferencing Applications is Hard

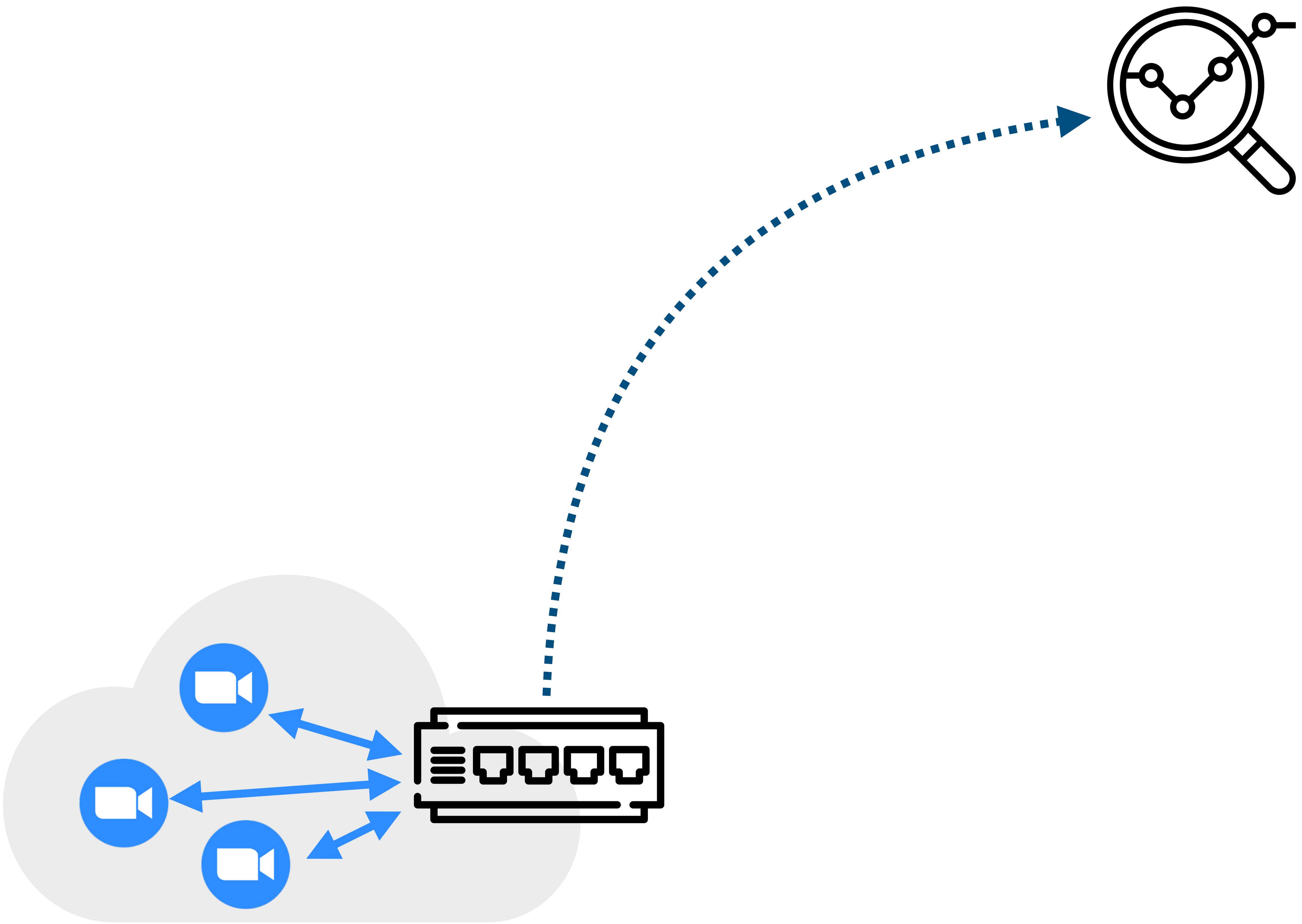
Problem: **Hard for researchers and operators to extract useful metrics from traffic**

- Quality/performance metrics not directly observable from traffic
- Application-provided metrics only useful for small-scale, controlled experiment



⟶ Operator's vantage point and capabilities less well explored

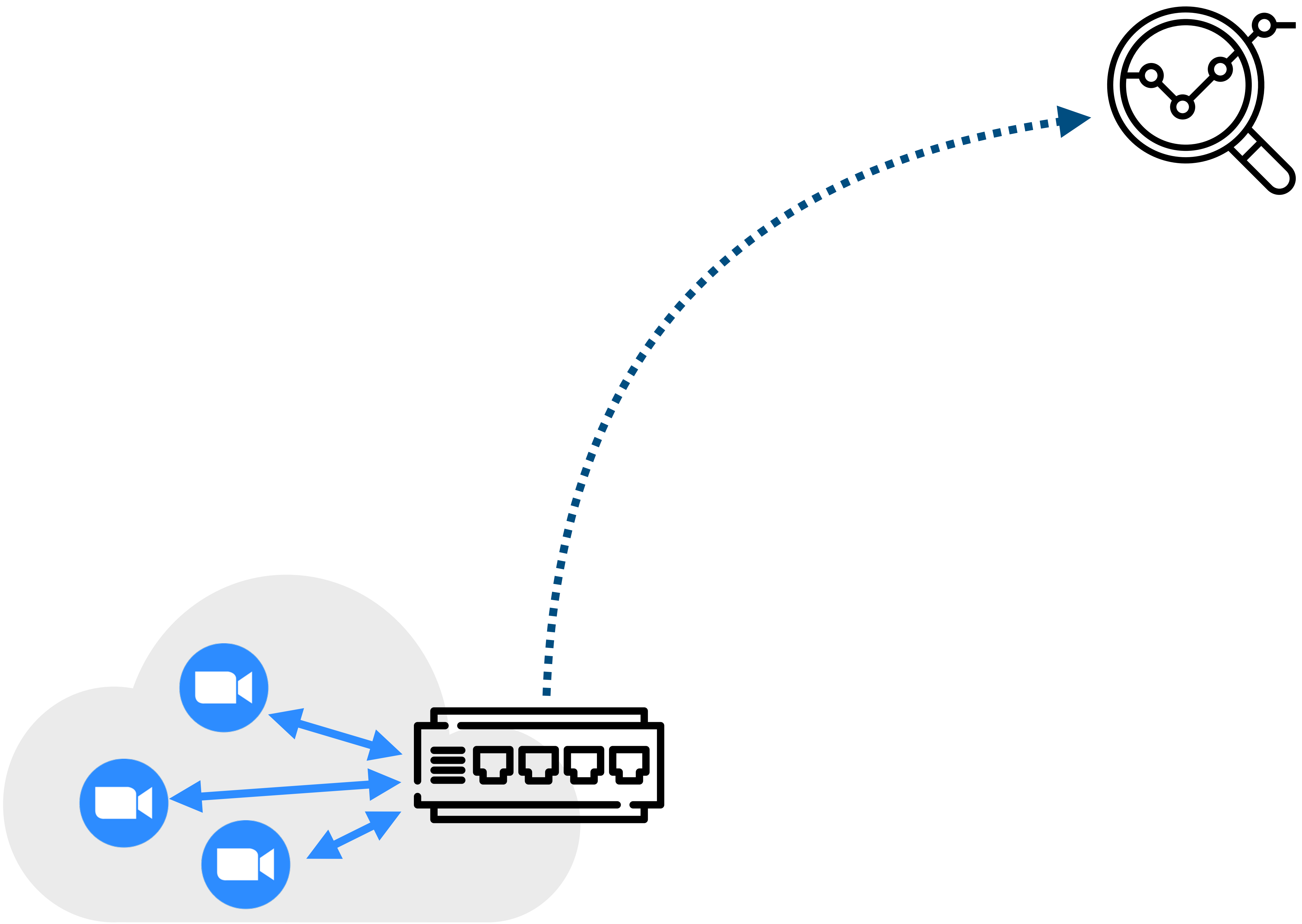
Enabling Passive Measurement



Goal:

Measure & analyze performance of video-conferencing sessions *in the wild*

Enabling Passive Measurement

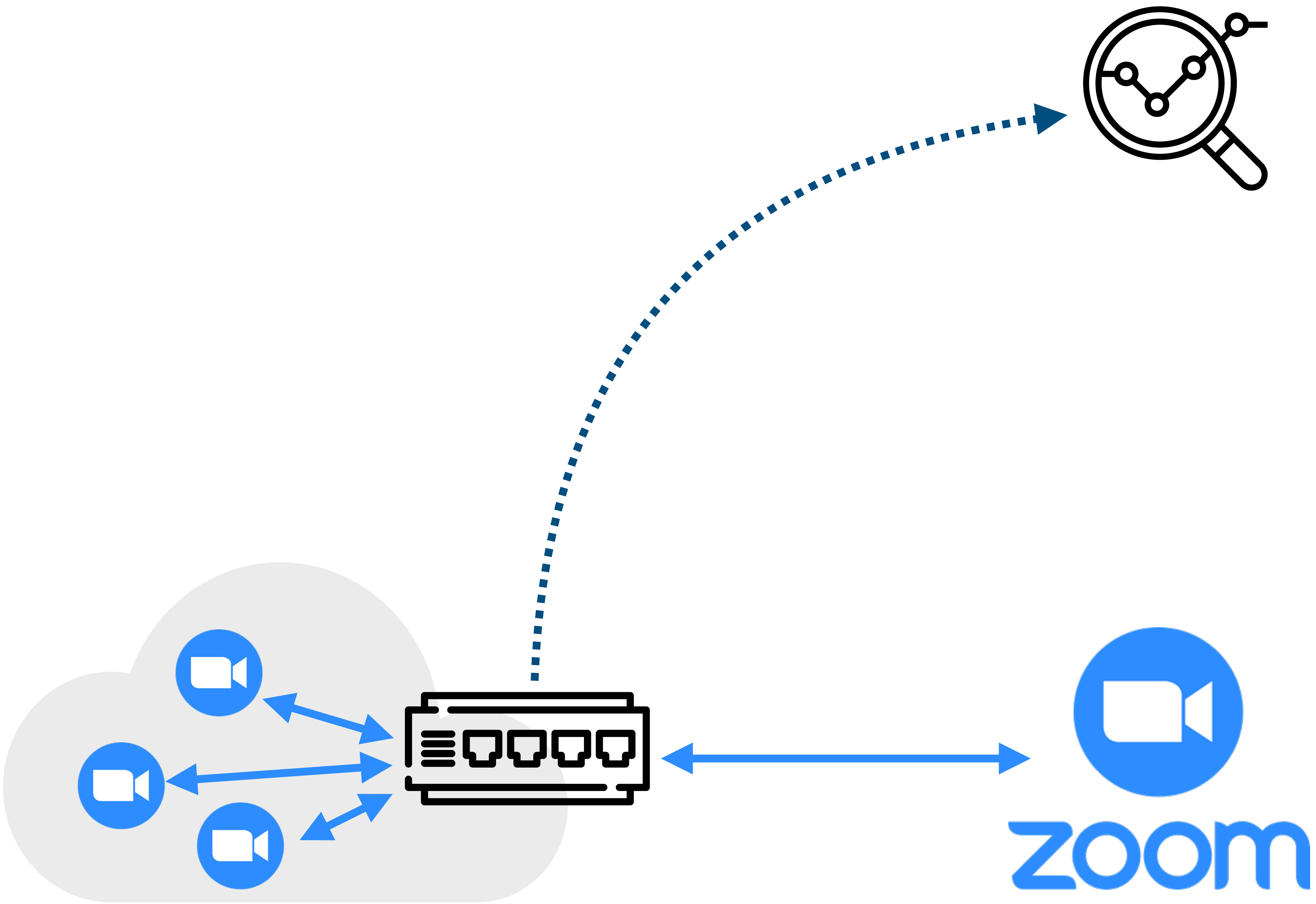


Goal:

Measure & analyze performance of video-conferencing sessions *in the wild*

- passively collected packets
- without end-host control
- in large-scale networks

Enabling Passive Measurement



Goal:

Measure & analyze performance of video-conferencing sessions *in the wild*

- passively collected packets
- without end-host control
- in large-scale networks

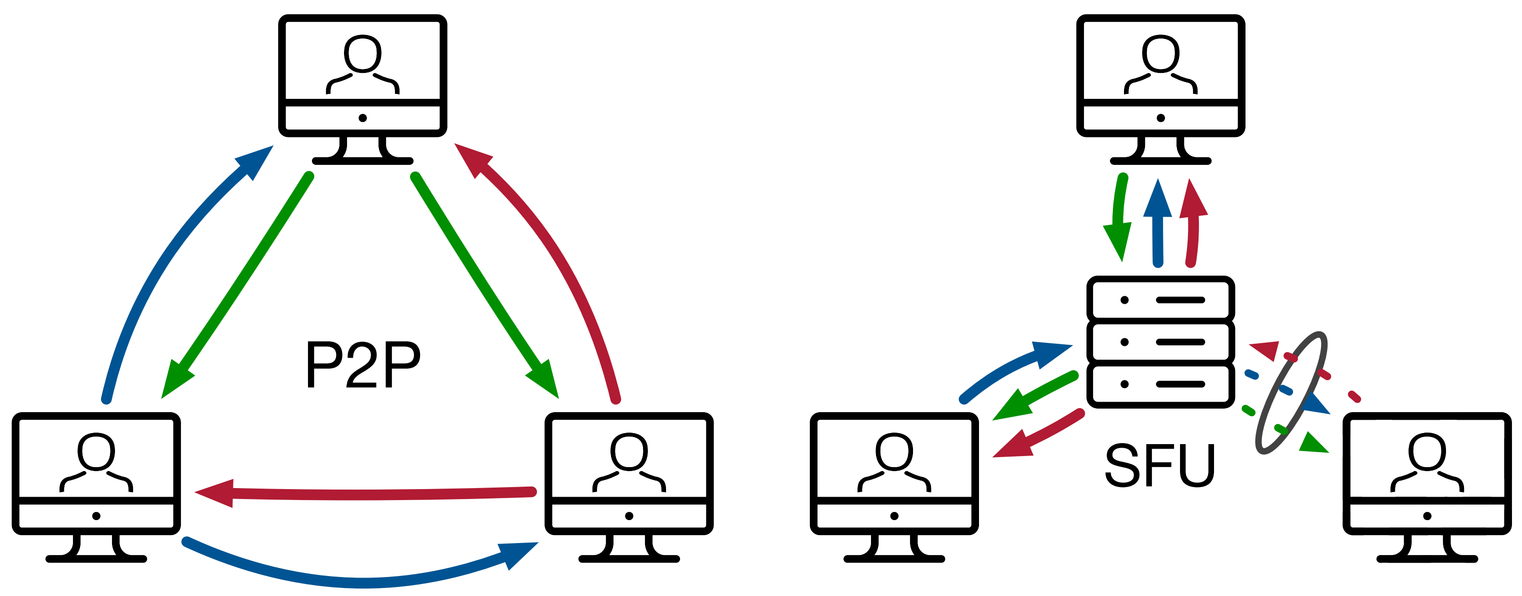
- widely used in general and at Princeton
- particularly challenging: proprietary packet format

Challenges

Video conferencing is complex.

Challenges

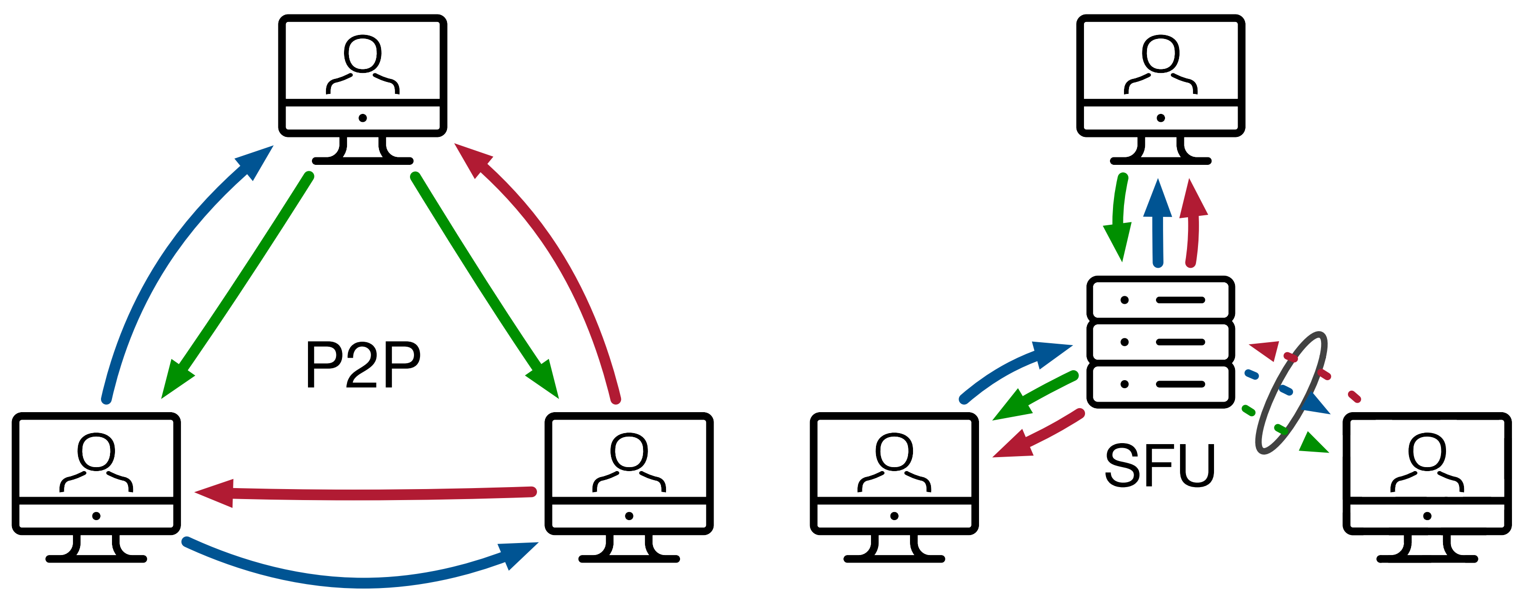
Video conferencing is complex.



Use of different conferencing architectures

Challenges

Video conferencing is complex.

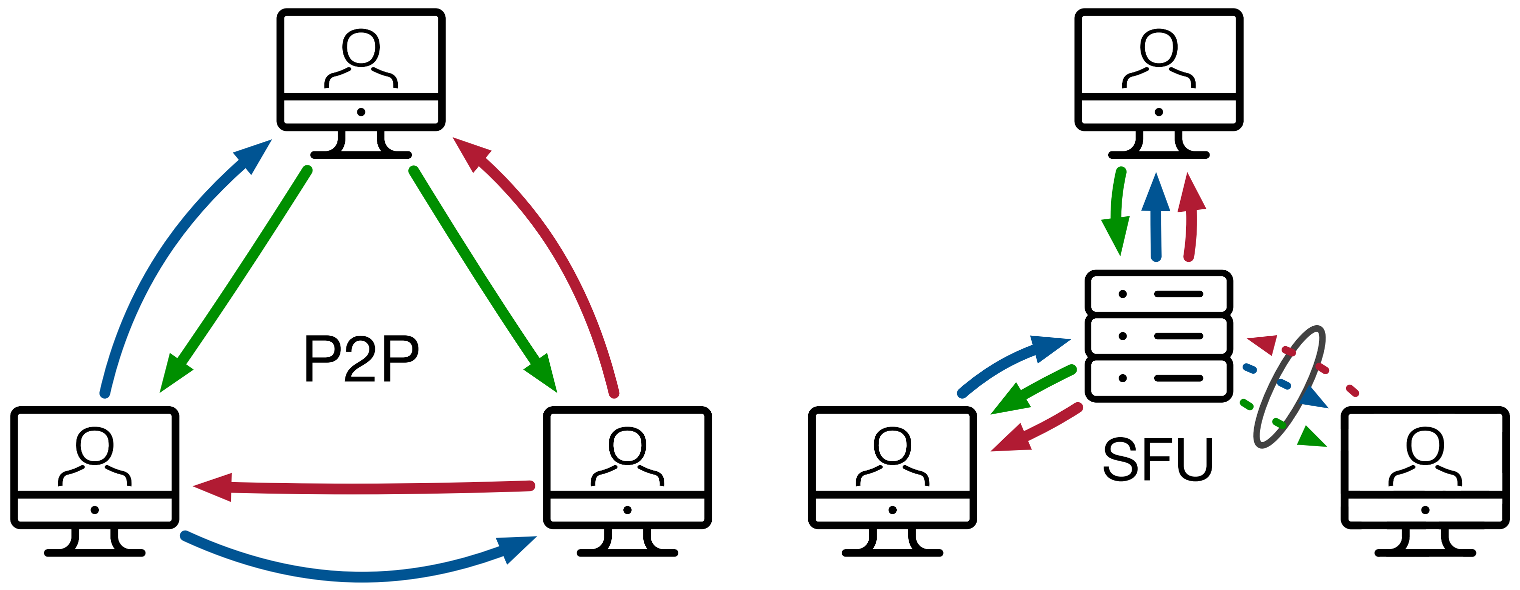


Use of different conferencing architectures

Encrypted control traffic and media

Challenges

Video conferencing is complex.



Use of different conferencing architectures

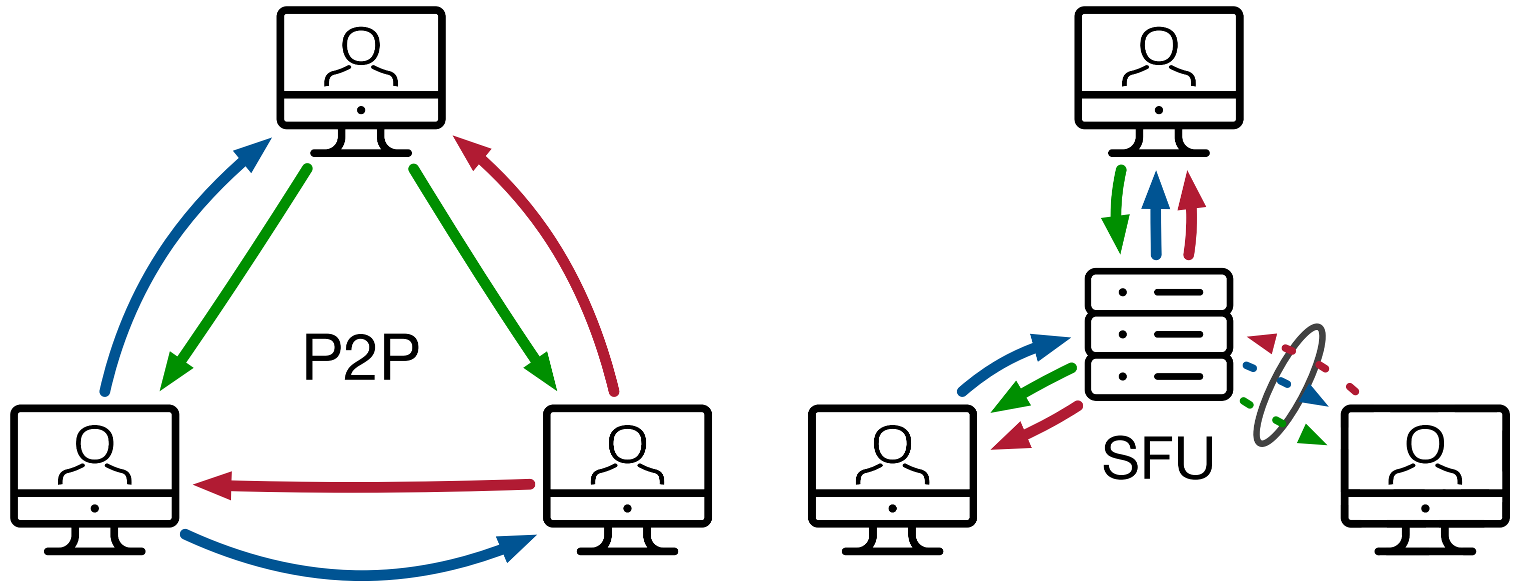
Encrypted control traffic and media

Meeting
|
Participant
|
Media Stream
|
Sub-Stream
|
Media Frame
|
Packet

Complex hierarchy within network protocols

Challenges

Video conferencing is complex.



Use of different conferencing architectures

1

How do we reliably detect all Zoom traffic?

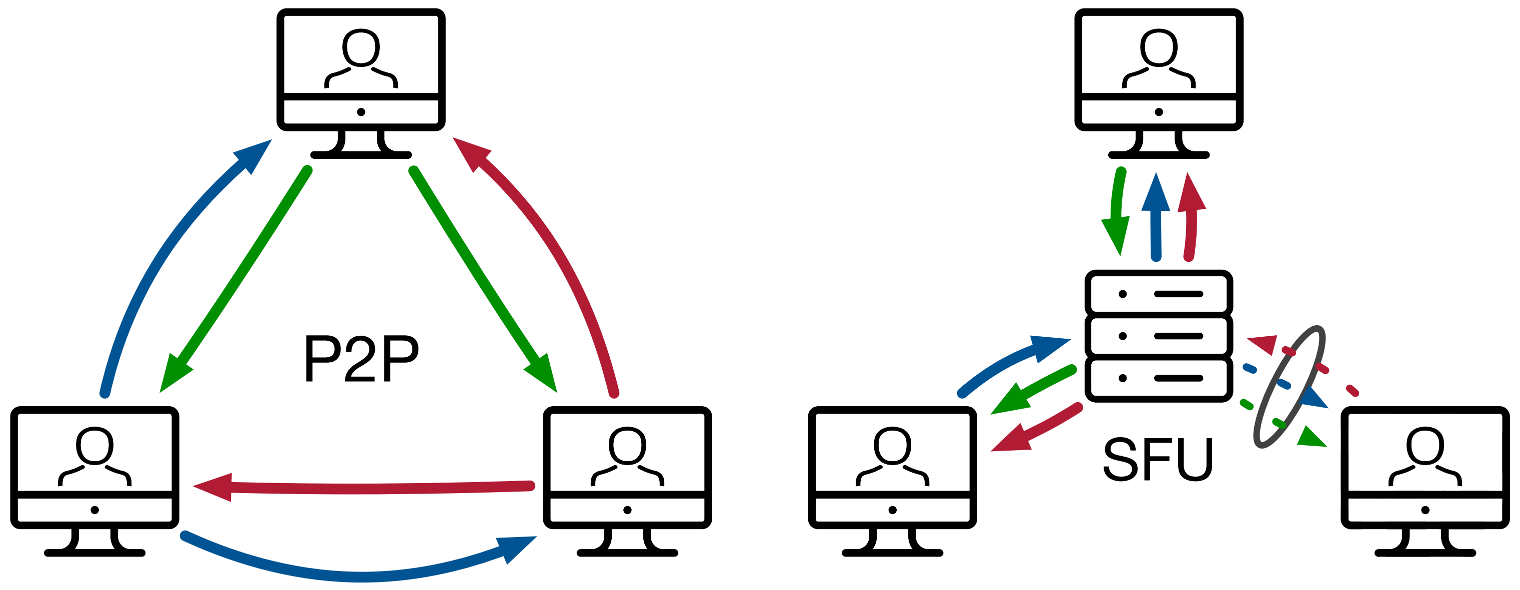
Encrypted control traffic and media

Meeting
|
Participant
|
Media Stream
|
Sub-Stream
|
Media Frame
|
Packet

Complex hierarchy within network protocols

Challenges

Video conferencing is complex.



Use of different conferencing architectures

Encrypted control traffic and media

Meeting
|
Participant
|
Media Stream
|
Sub-Stream
|
Media Frame
|
Packet

Complex hierarchy within network protocols

1

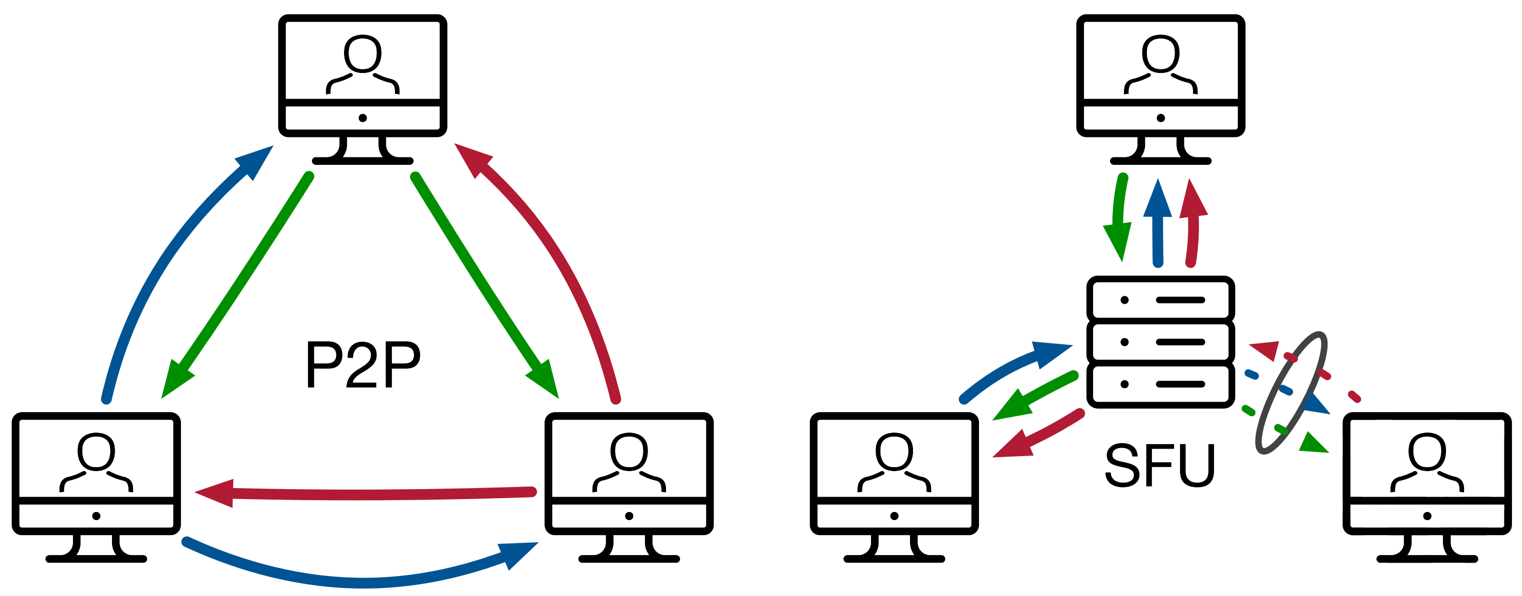
How do we reliably detect all Zoom traffic?

2

What is Zoom's packet format and what information can be extracted from packets?

Challenges

Video conferencing is complex.



Use of different conferencing architectures

1

How do we reliably detect all Zoom traffic?

Encrypted control traffic and media

2

What is Zoom's packet format and what information can be extracted from packets?

Meeting
Participant
Media Stream
Sub-Stream
Media Frame
Packet

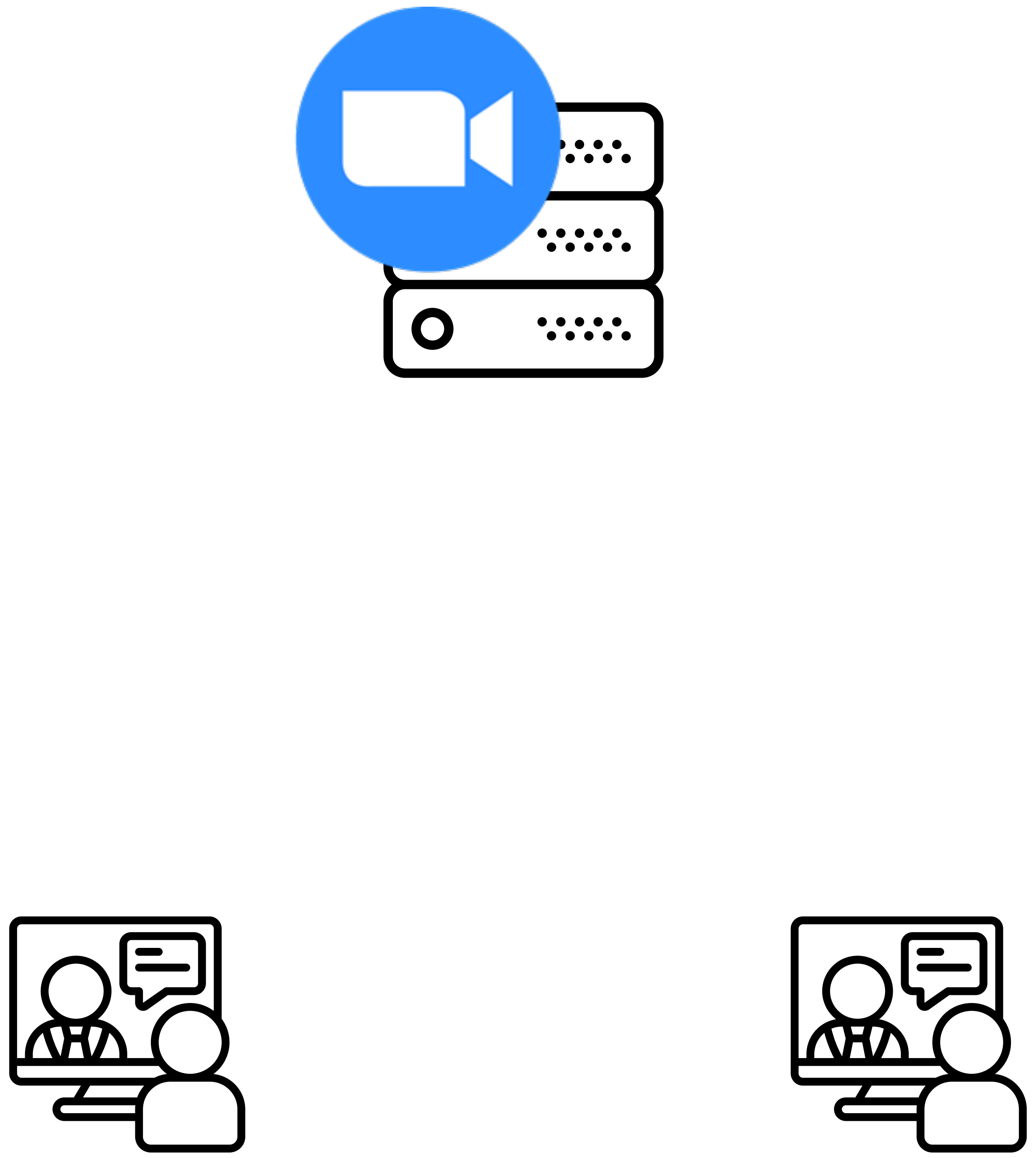
Complex hierarchy within network protocols

3

How do we group packets belonging to the same meeting together?

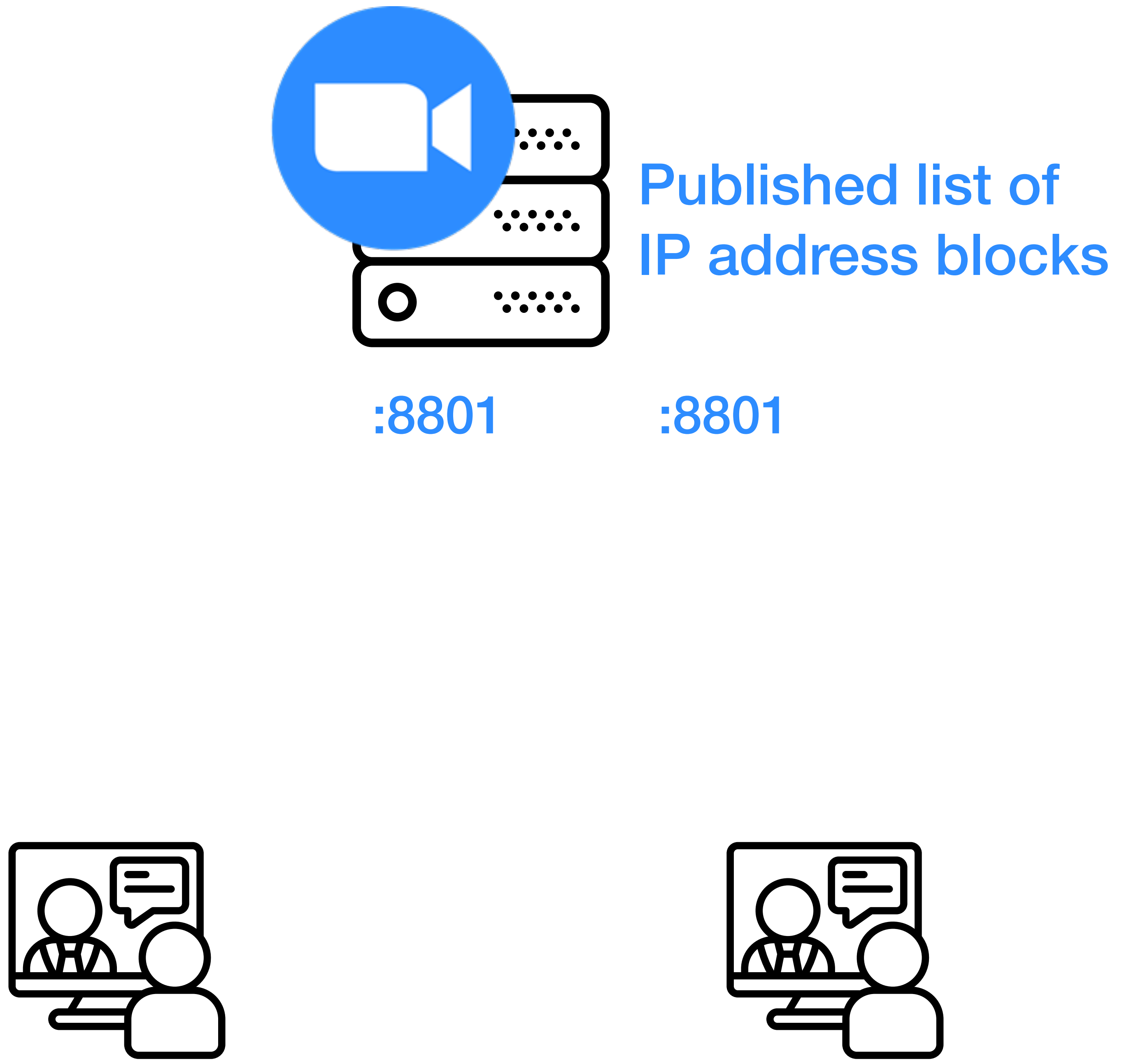
Demystifying Zoom

1 Detecting Zoom Traffic



Demystifying Zoom

1 Detecting Zoom Traffic

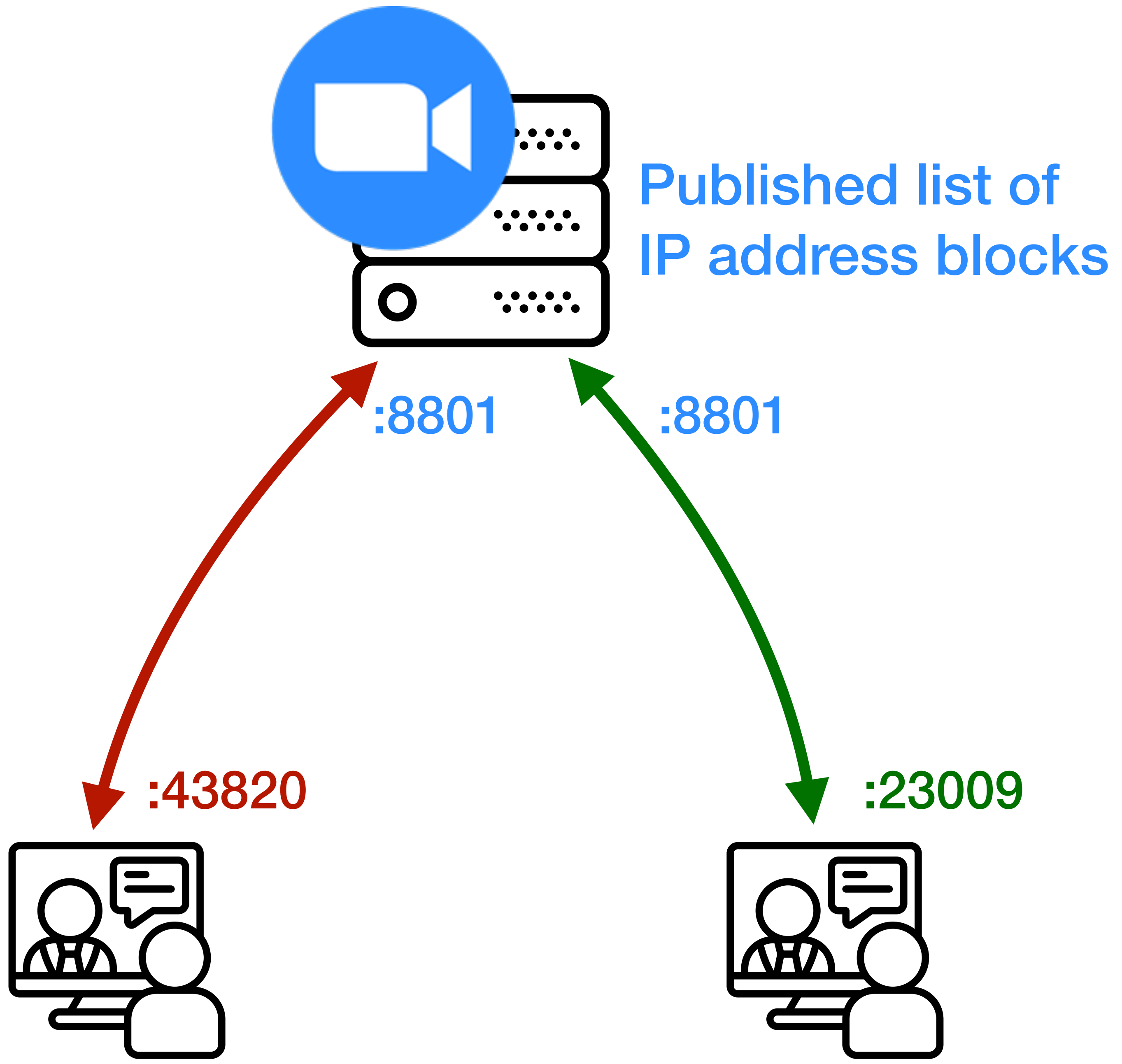


Server-Based Meetings

- Published list of IP addresses and port numbers on server side

Demystifying Zoom

1 Detecting Zoom Traffic

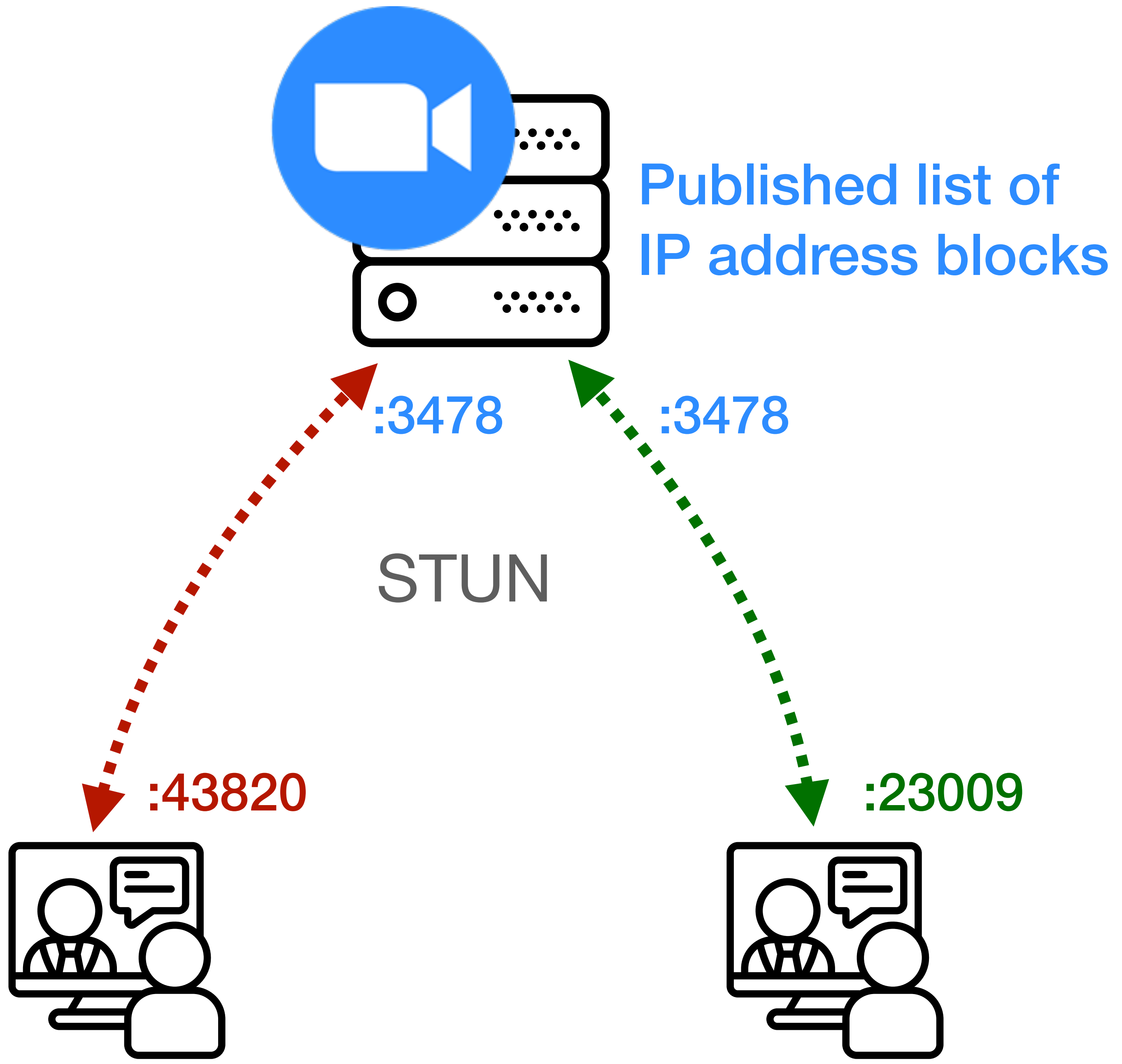


Server-Based Meetings

- Published list of IP addresses and port numbers on server side
- Ephemeral ports on client side

Demystifying Zoom

1 Detecting Zoom Traffic



Server-Based Meetings

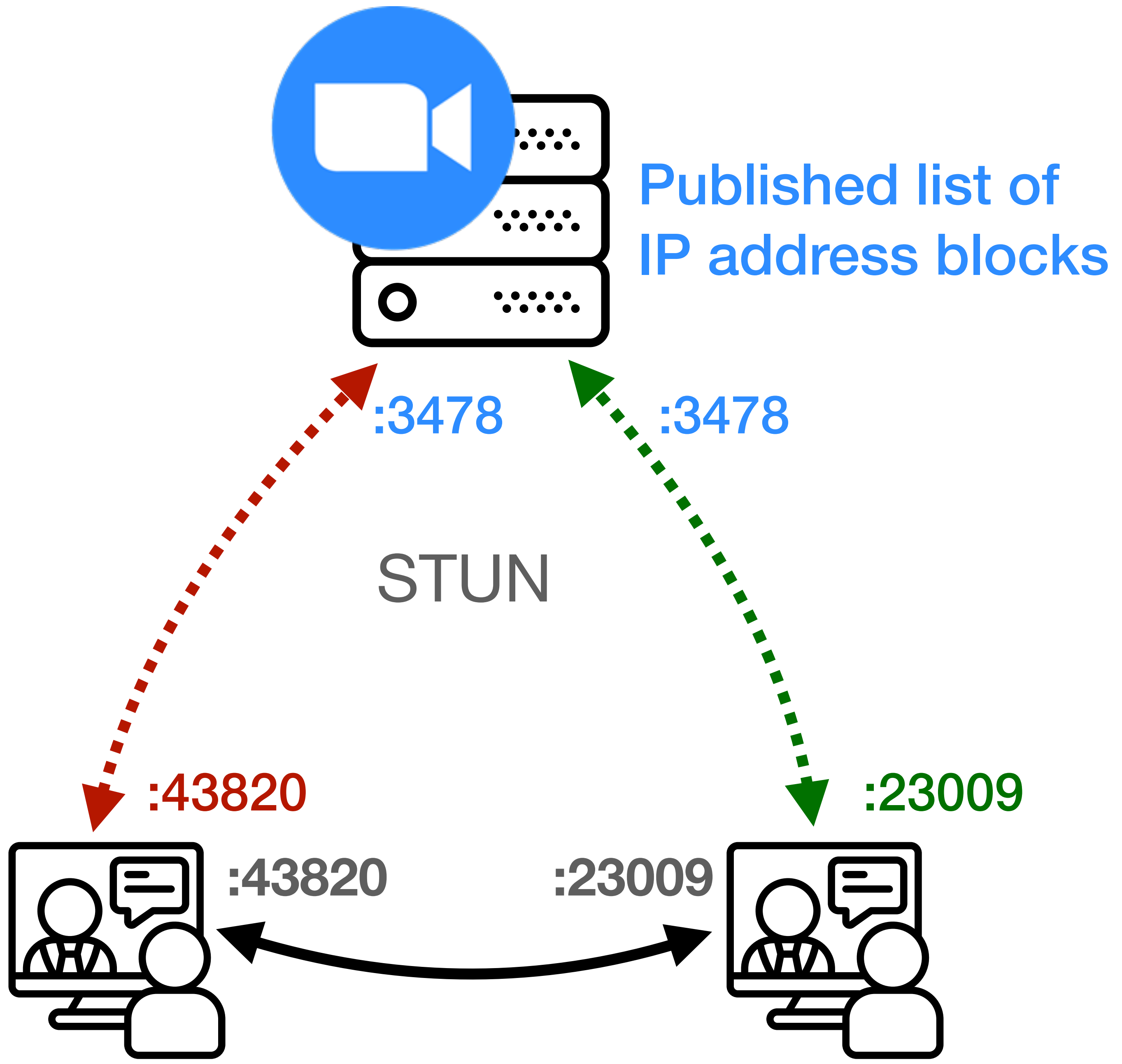
- Published list of IP addresses and port numbers on server side
- Ephemeral ports on client side

P2P Meetings

- STUN exchange before P2P establishment

Demystifying Zoom

1 Detecting Zoom Traffic



Server-Based Meetings

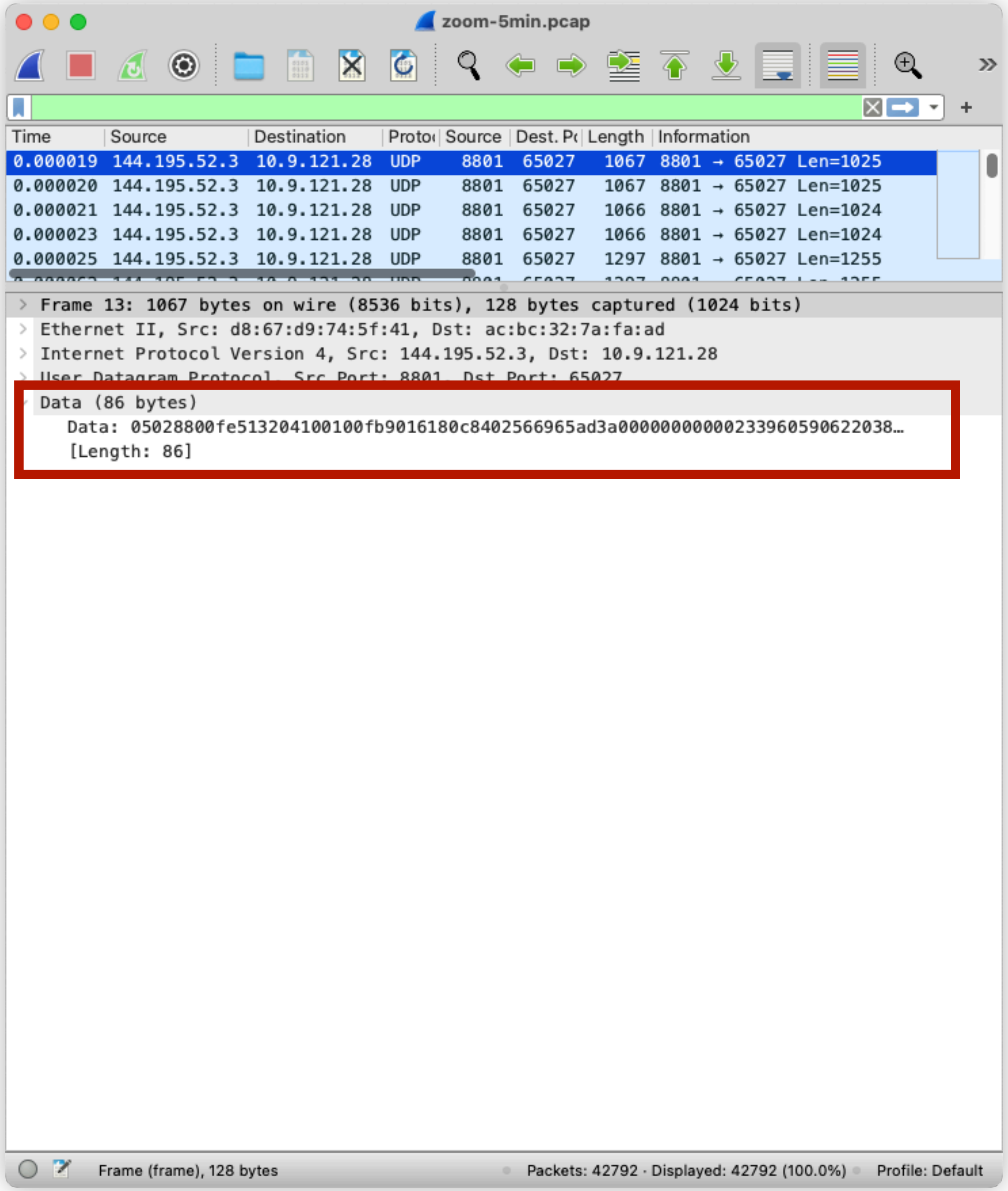
- Published list of IP addresses and port numbers on server side
- Ephemeral ports on client side

P2P Meetings

- STUN exchange before P2P establishment
- Use of client-side ports from STUN for P2P connection

Demystifying Zoom

2 Inferring Header Formats via Entropy Analysis

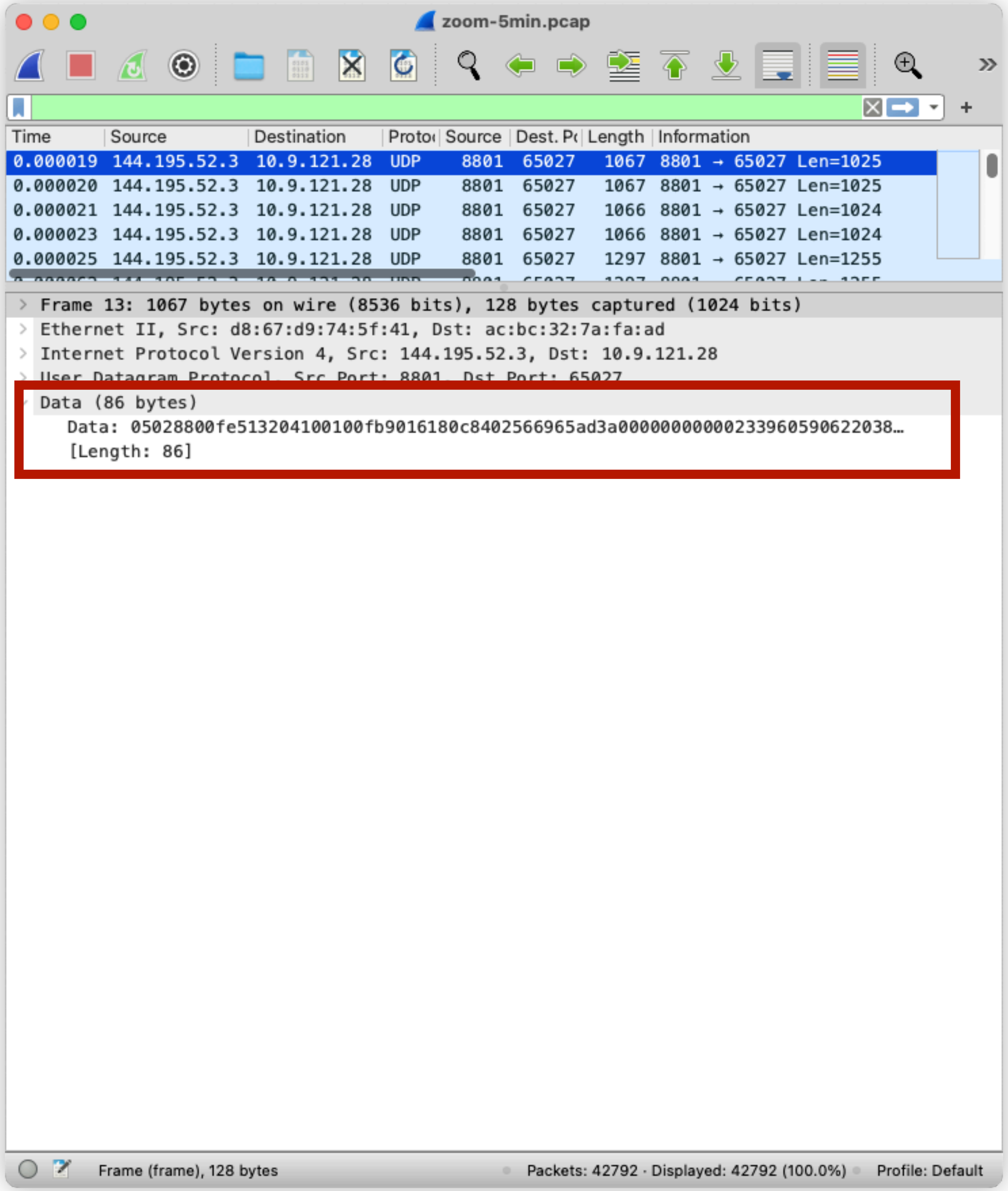


Demystifying Zoom

2 Inferring Header Formats via Entropy Analysis

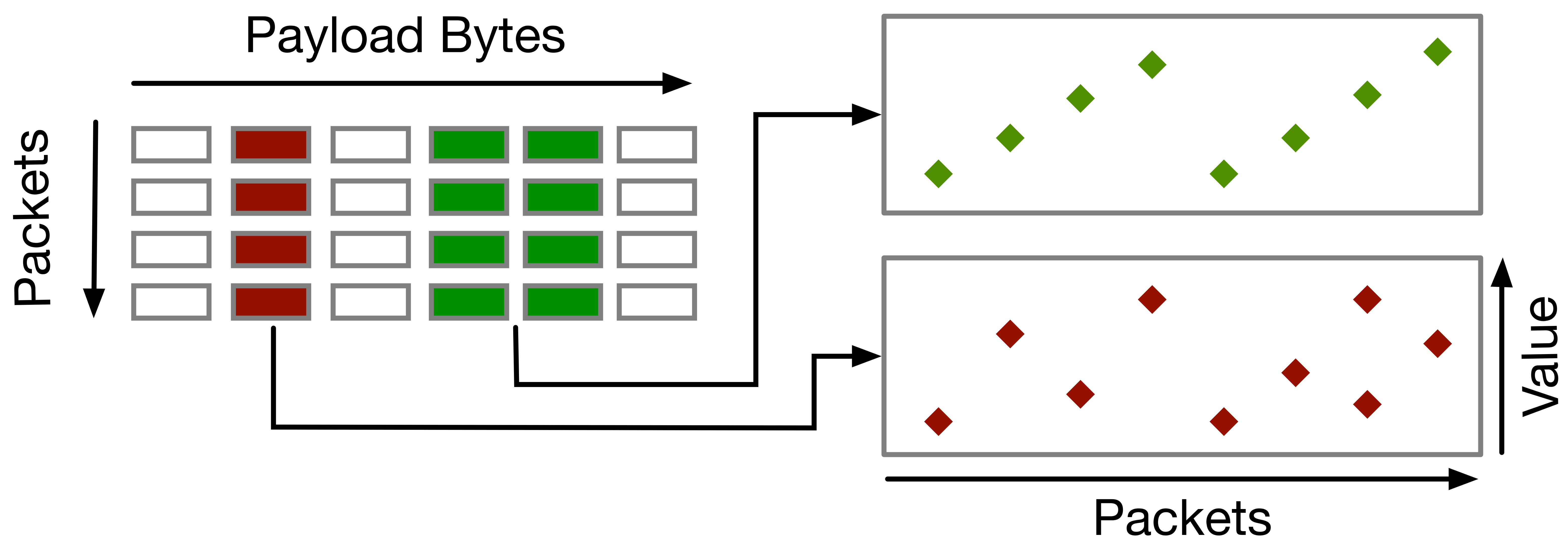
(1) Are there unencrypted parts in Zoom's media packets?

(2) If so, are there patterns that could map to header fields?



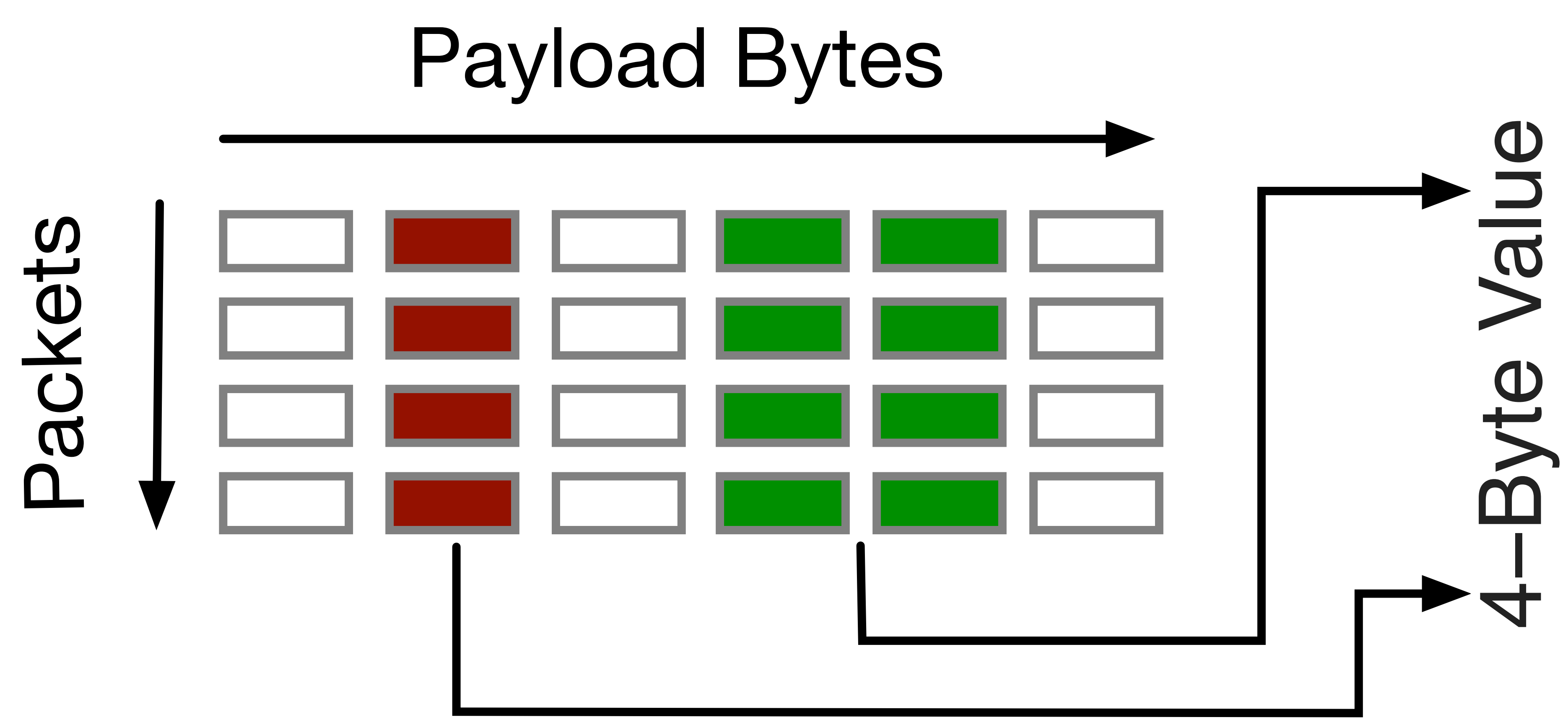
Demystifying Zoom

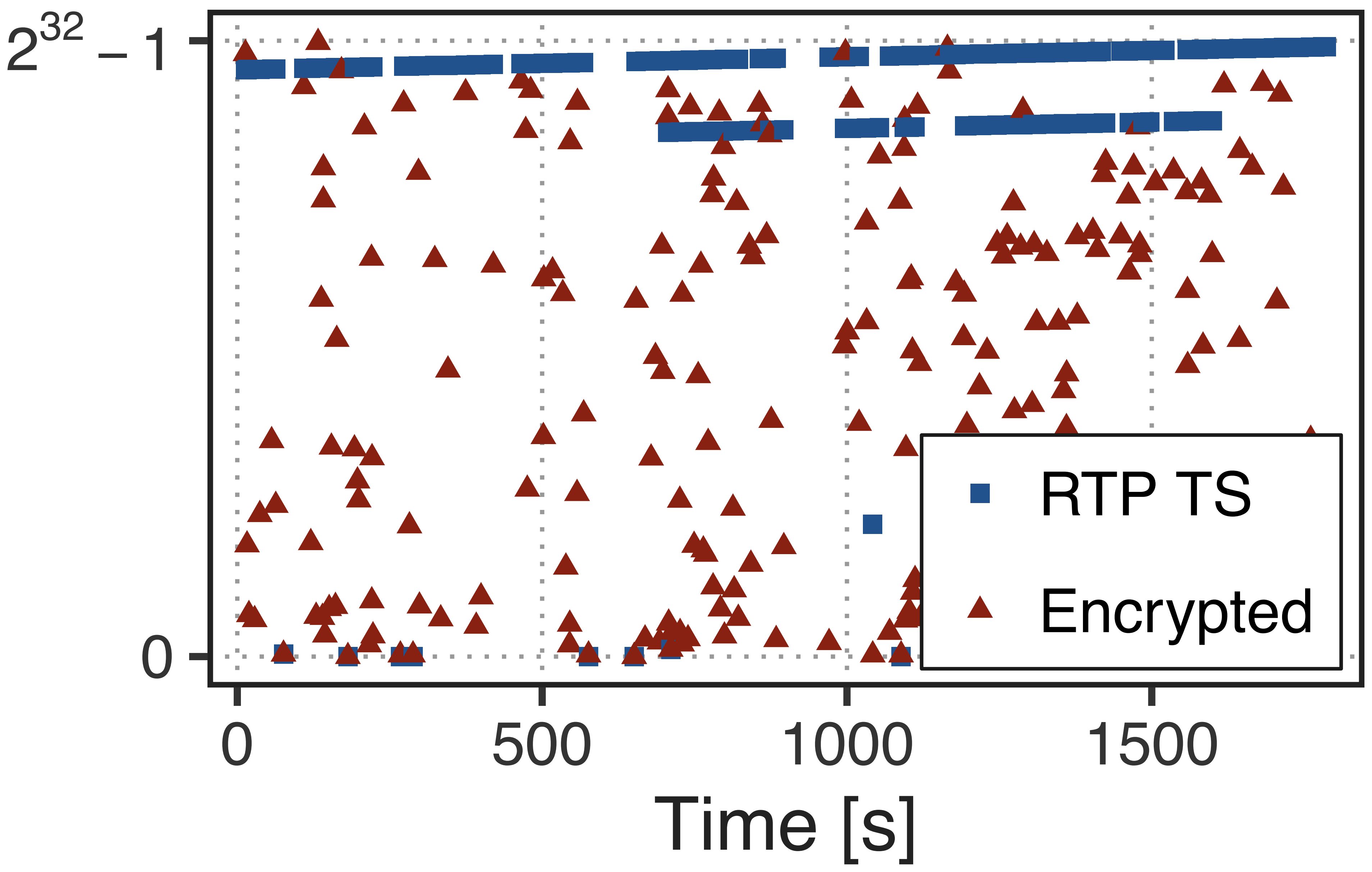
2 Inferring Header Formats via Entropy Analysis



Demystifying Zoom

2 Inferring Header Formats via Entropy Analysis



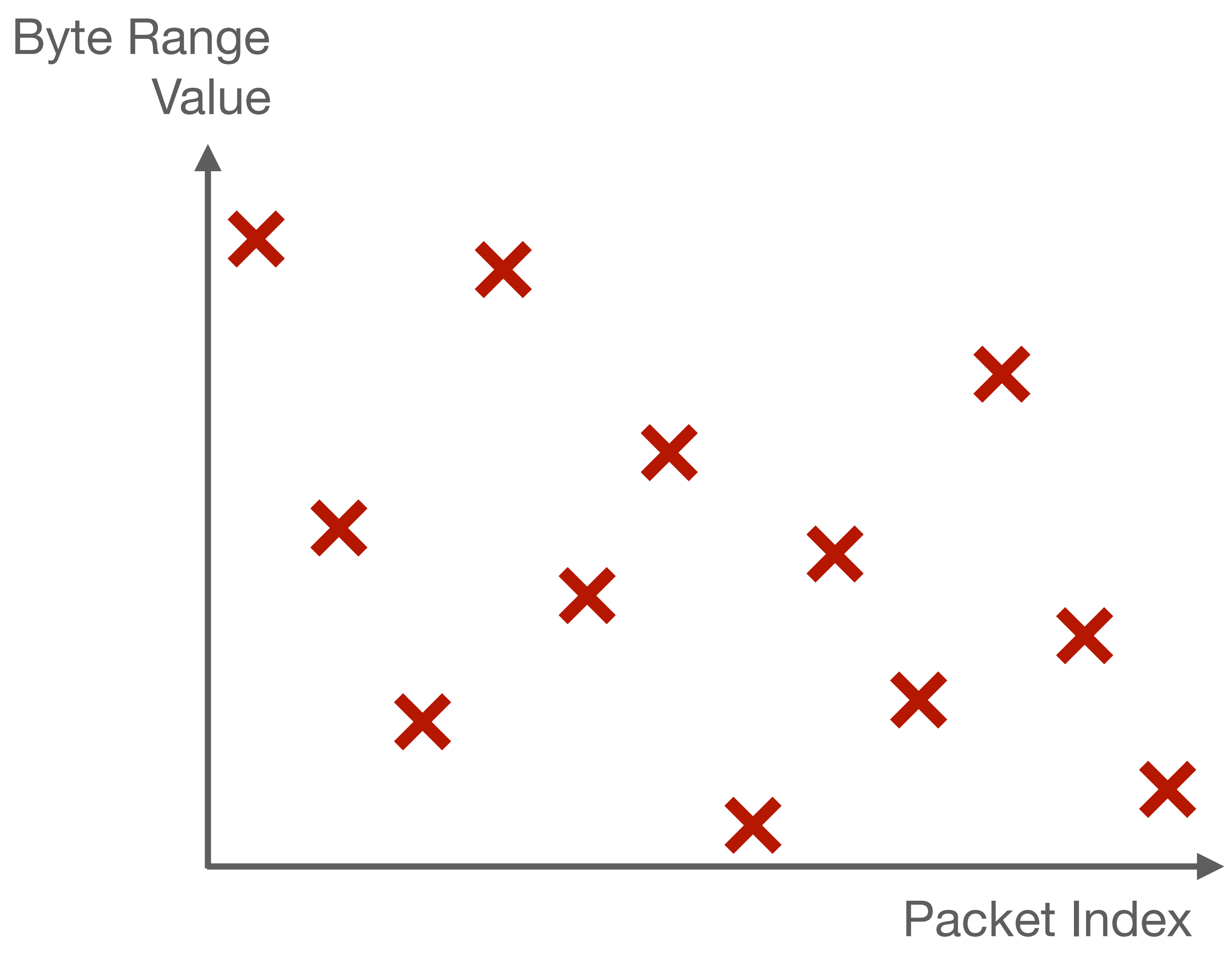


Demystifying Zoom

2 Inferring Header Formats via Entropy Analysis

Demystifying Zoom

2 Inferring Header Formats via Entropy Analysis



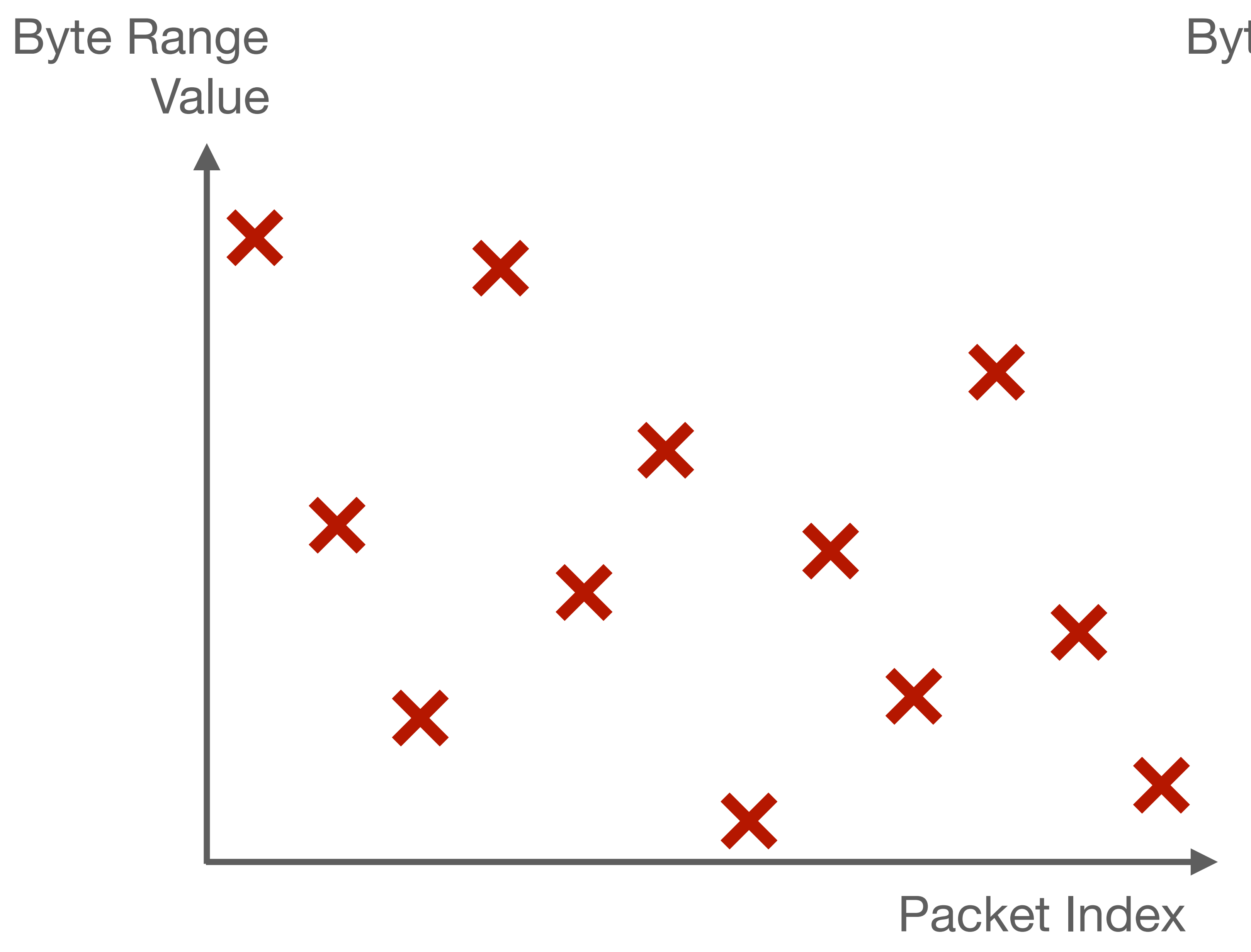
(1) random / max. entropy

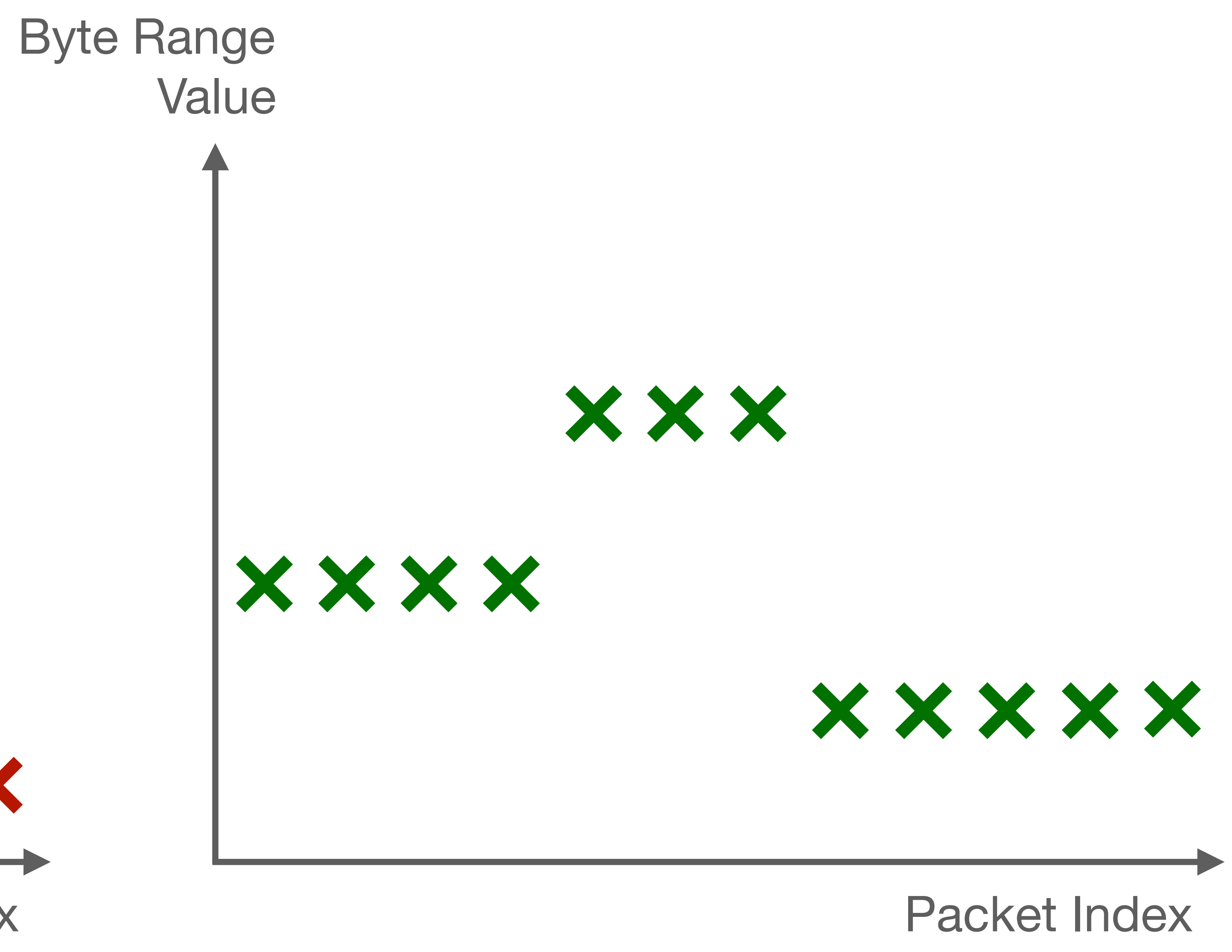
↓

encrypted

Demystifying Zoom

2 Inferring Header Formats via Entropy Analysis



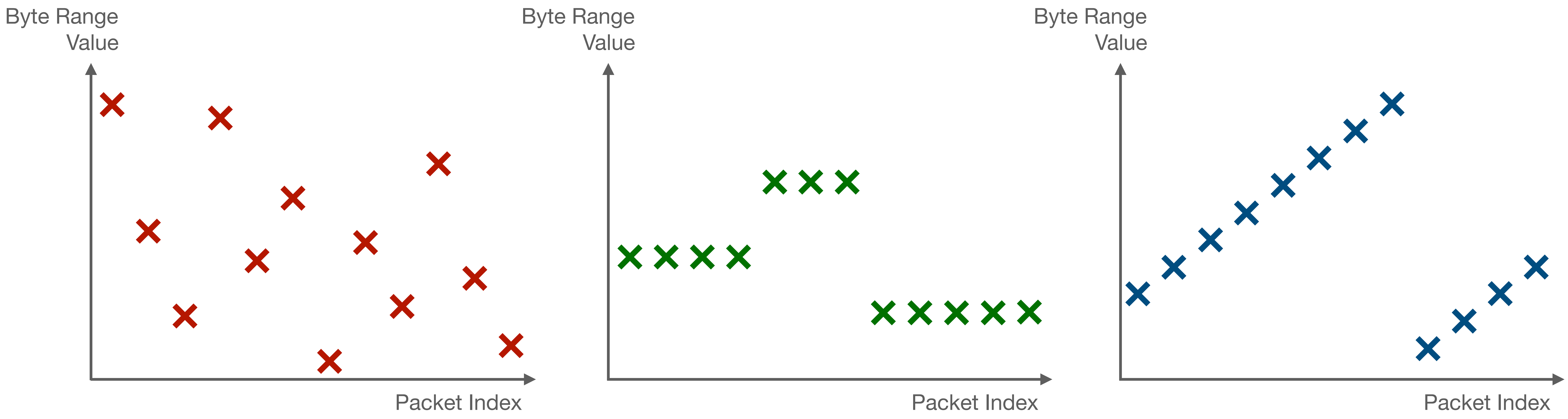


(1) random / max. entropy
↓
encrypted

(2) horizontal lines
↓
identifiers, bitfields

Demystifying Zoom

2 Inferring Header Formats via Entropy Analysis



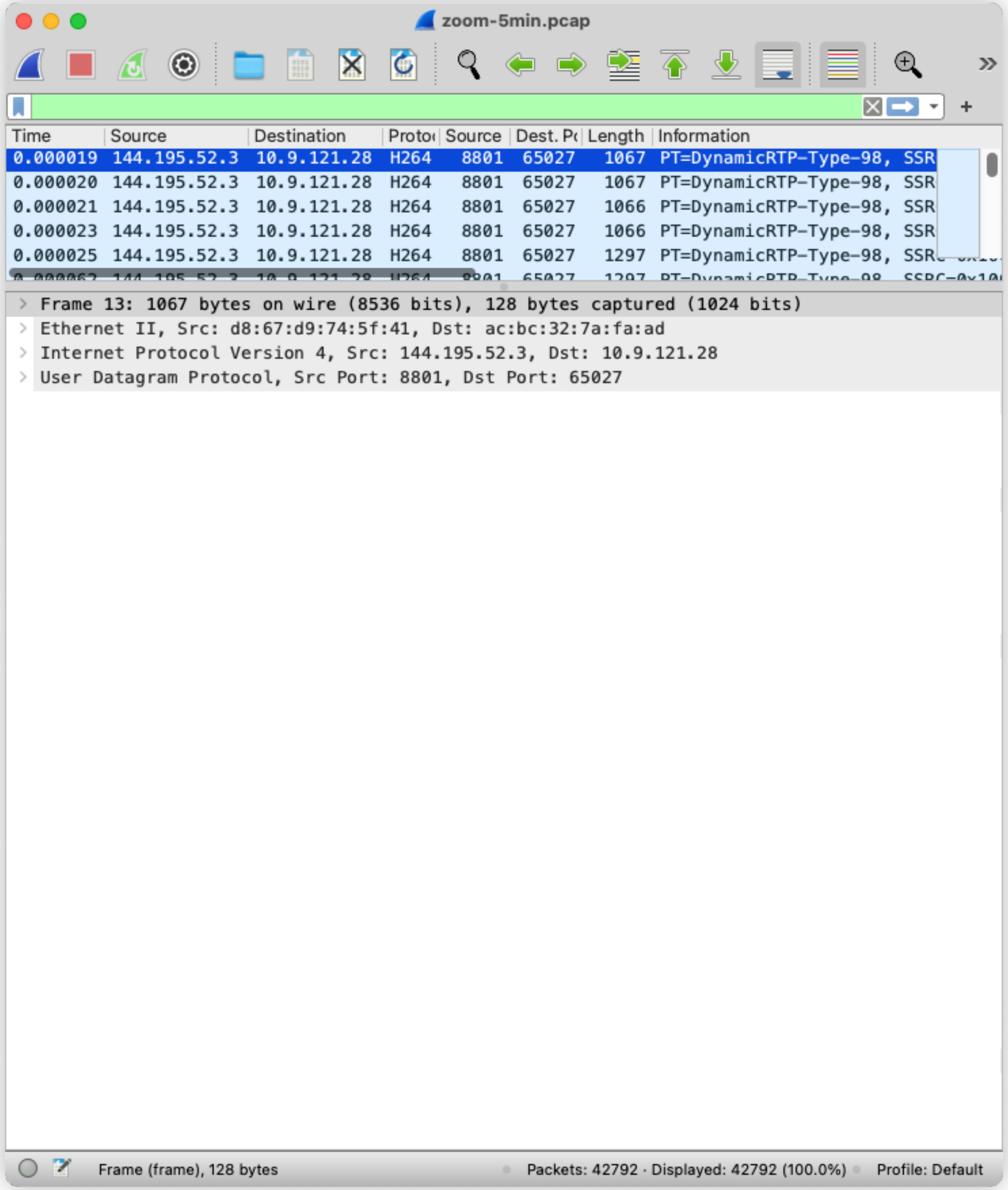
(1) random / max. entropy → encrypted

(2) horizontal lines → identifiers, bitfields

(3) diagonal lines → seq. numbers, timestamps

Demystifying Zoom

2 Inferring Header Formats via Entropy Analysis

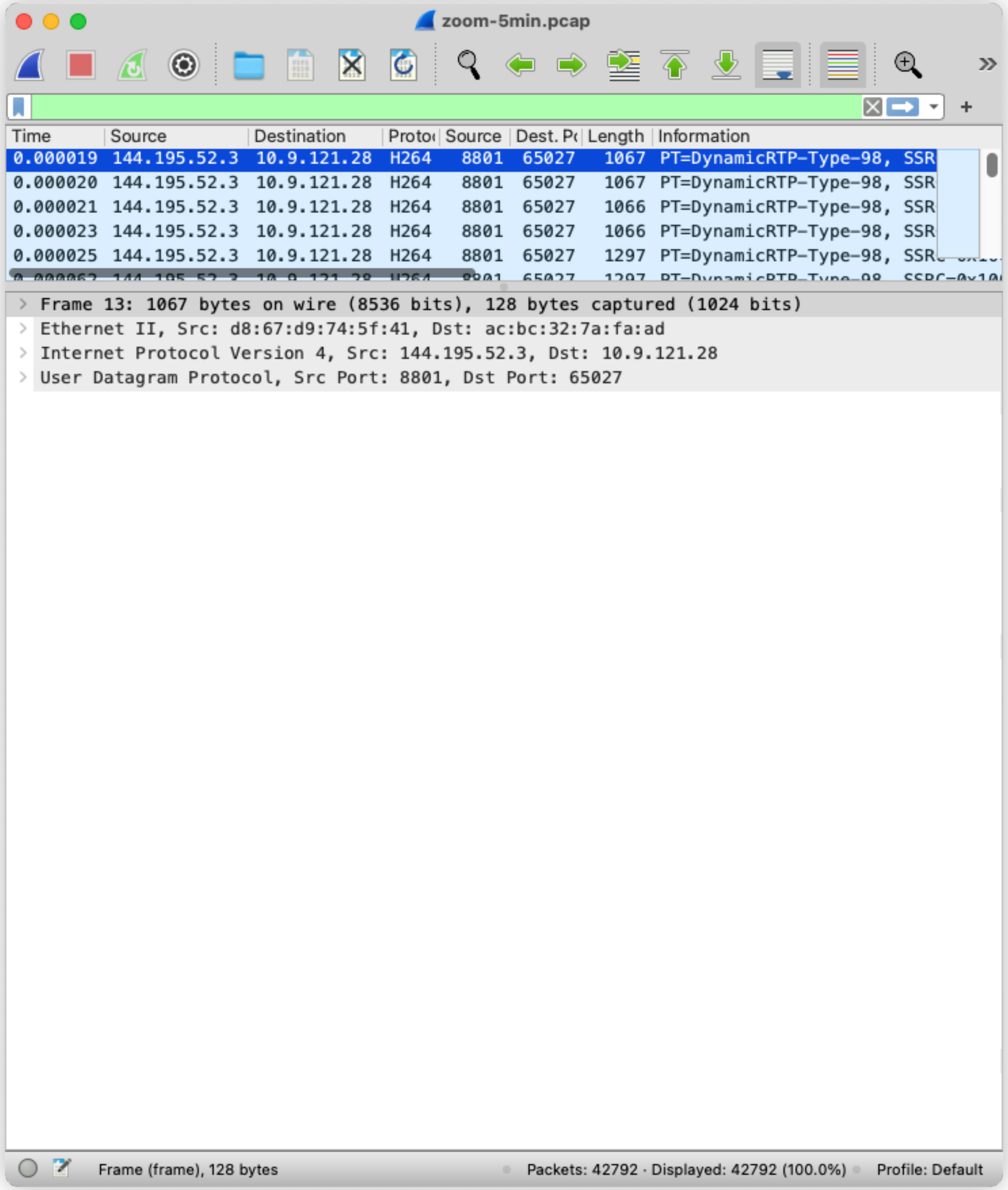


Demystifying Zoom

2 Inferring Header Formats via Entropy Analysis

…

UDP



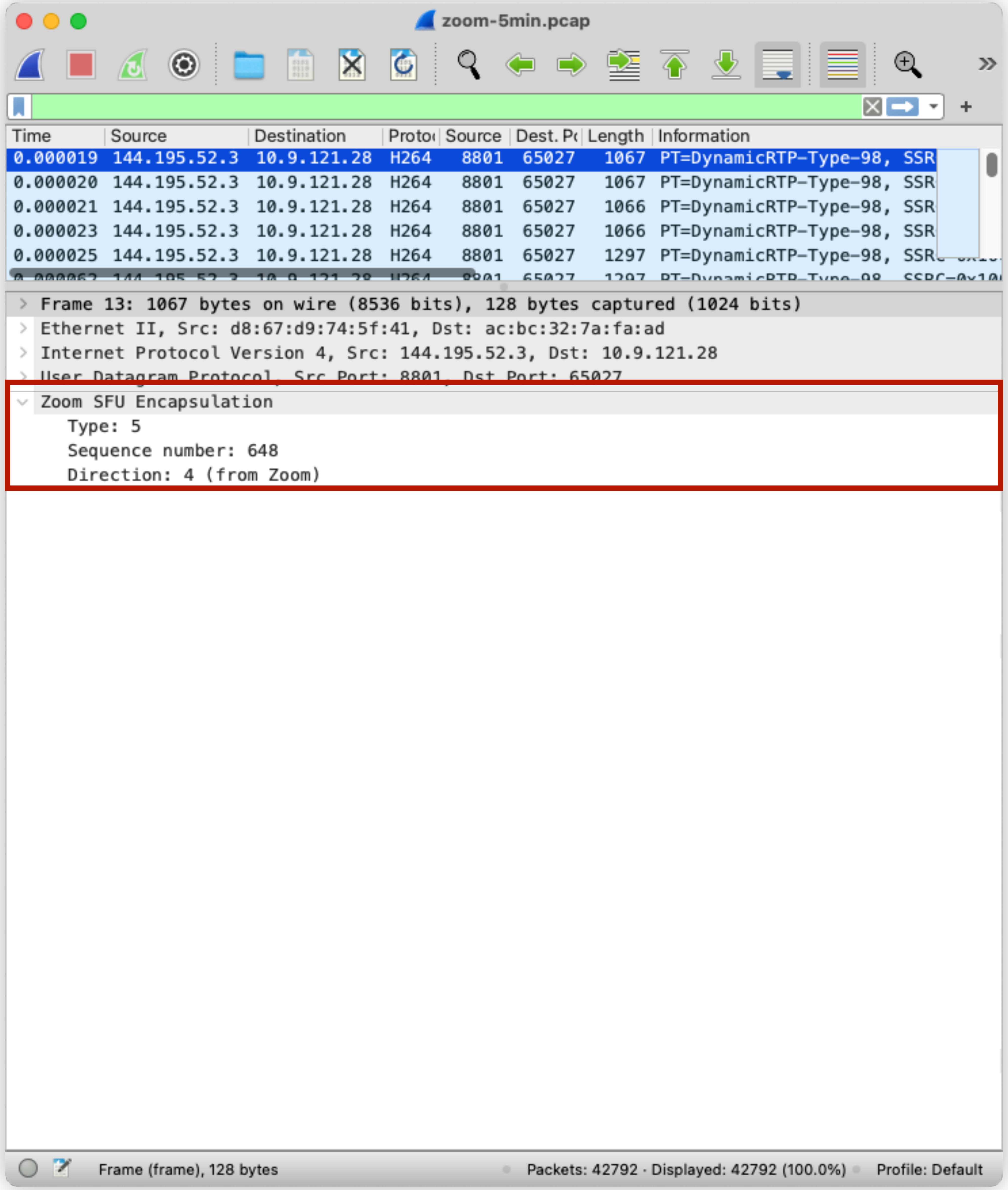
Demystifying Zoom

2 Inferring Header Formats via Entropy Analysis

…

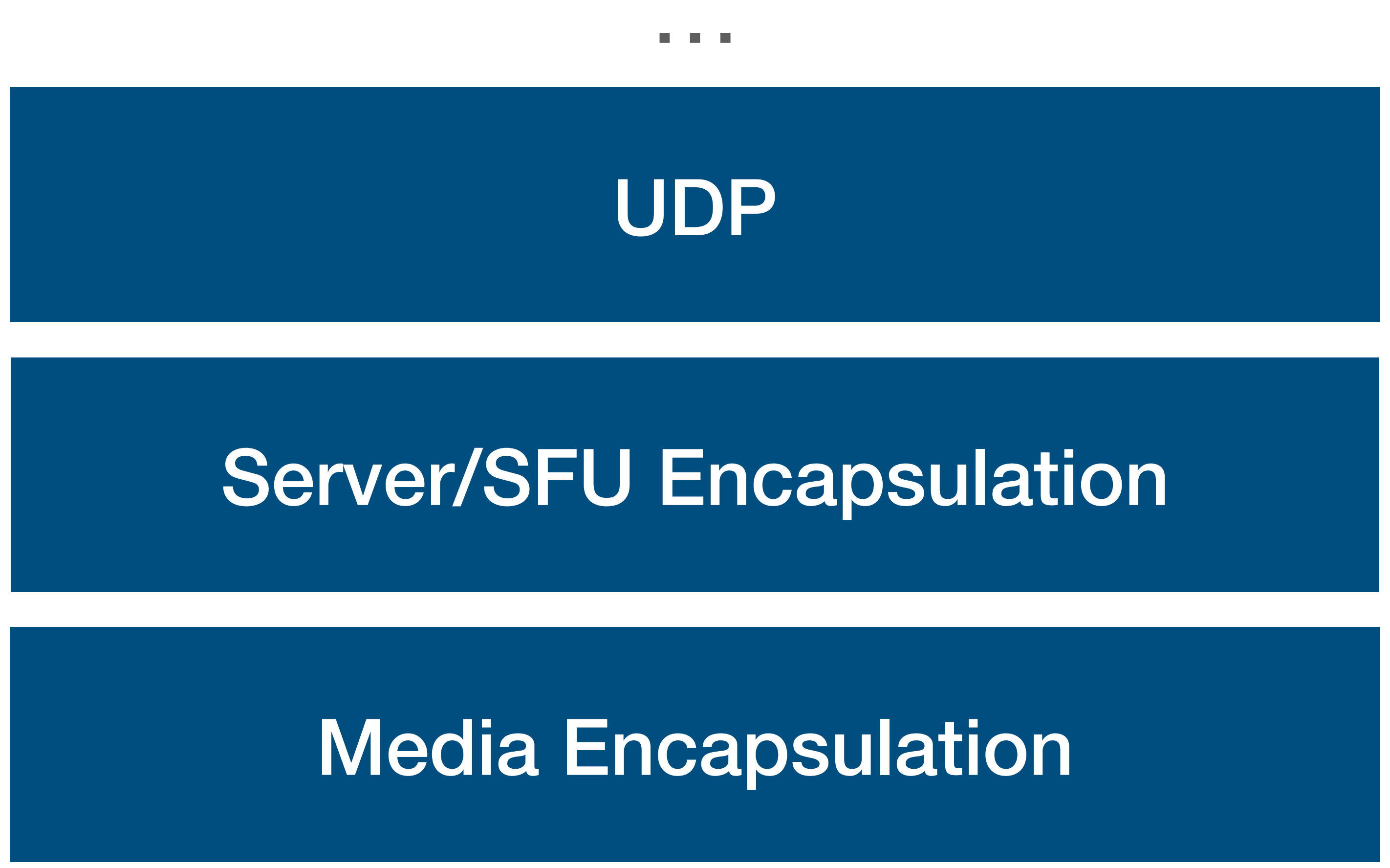
UDP

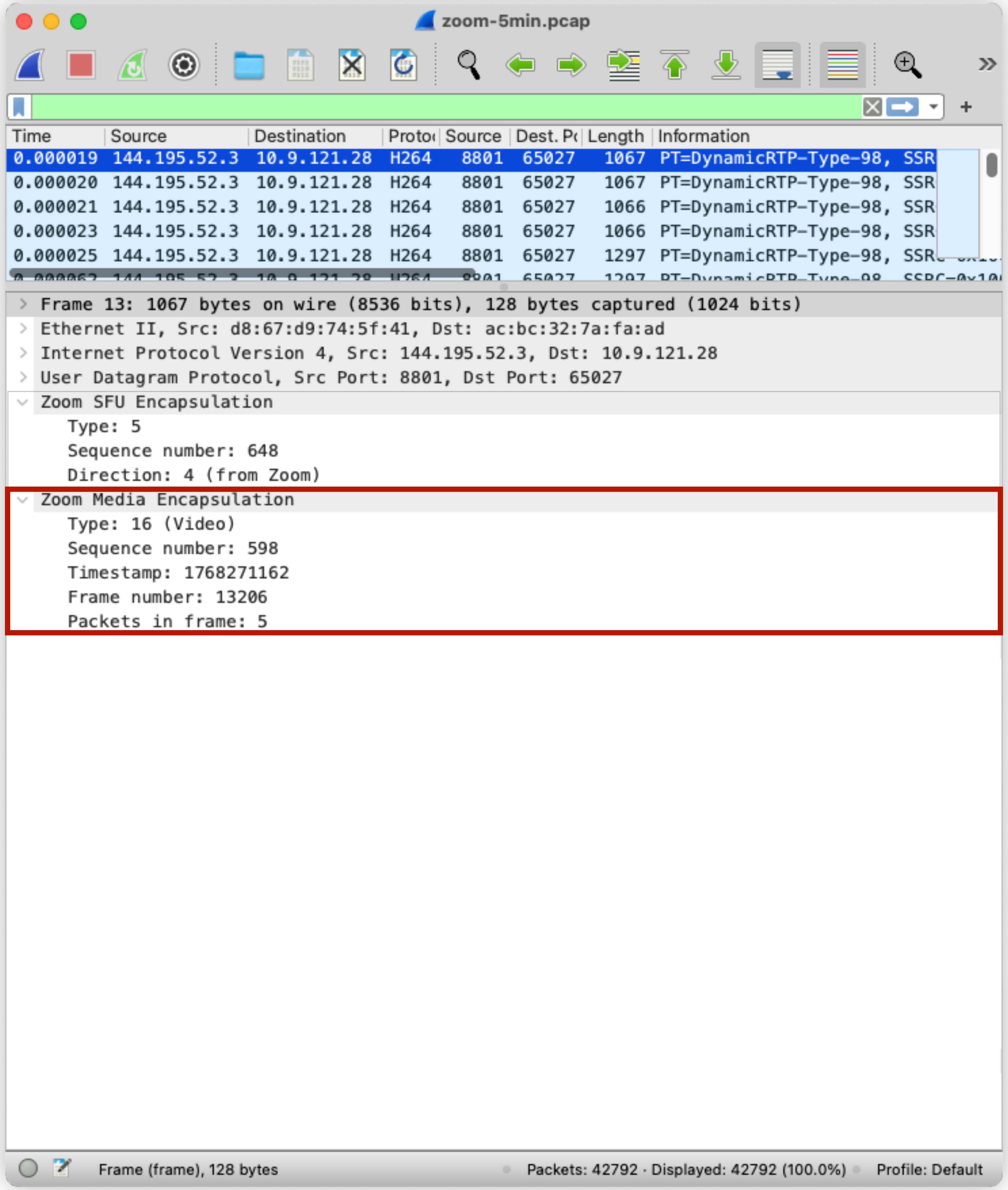
Server/SFU Encapsulation



Demystifying Zoom

❷ Inferring Header Formats via Entropy Analysis





Demystifying Zoom

② Inferring Header Formats via Entropy Analysis

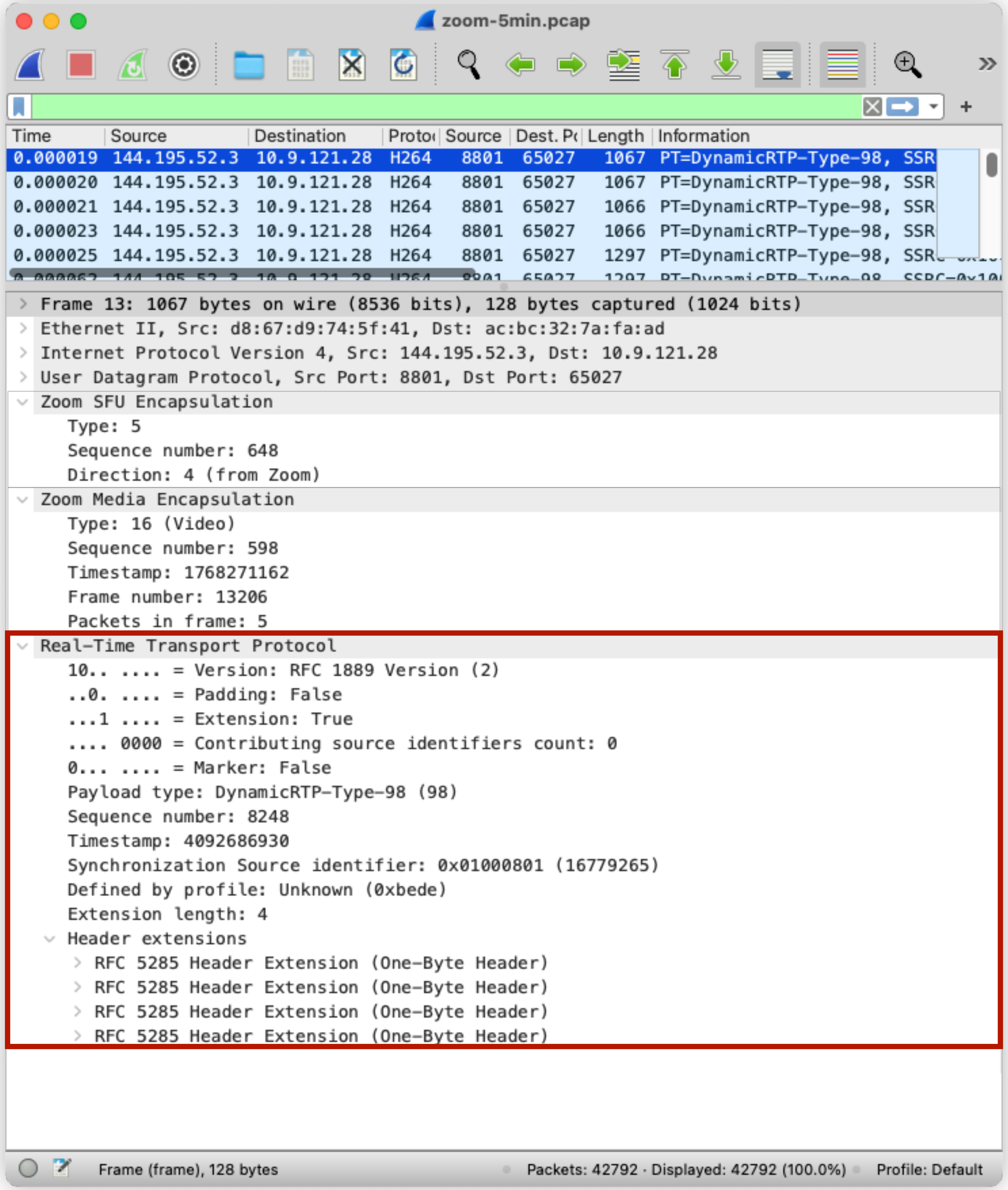
...

UDP

Server/SFU Encapsulation

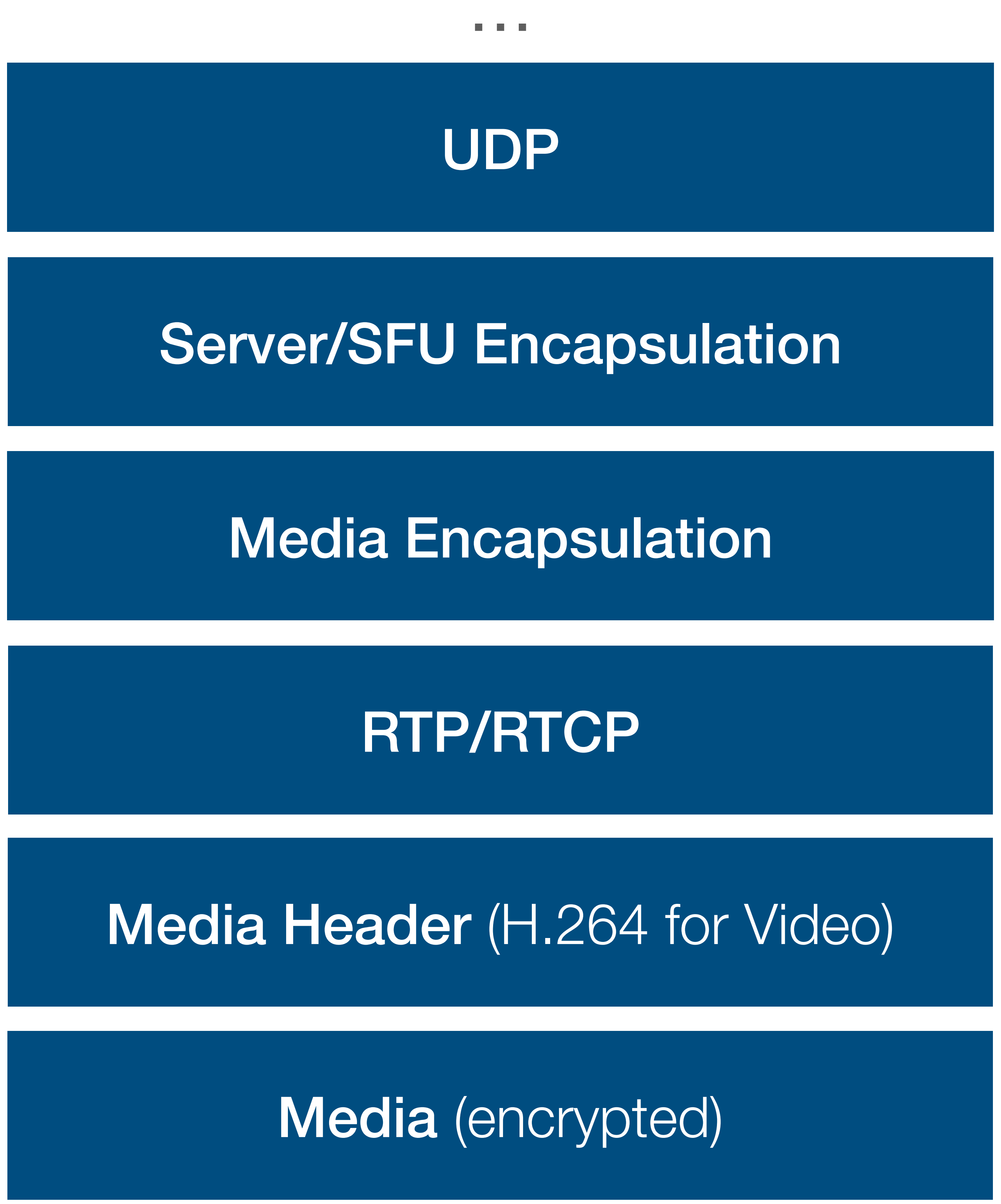
Media Encapsulation

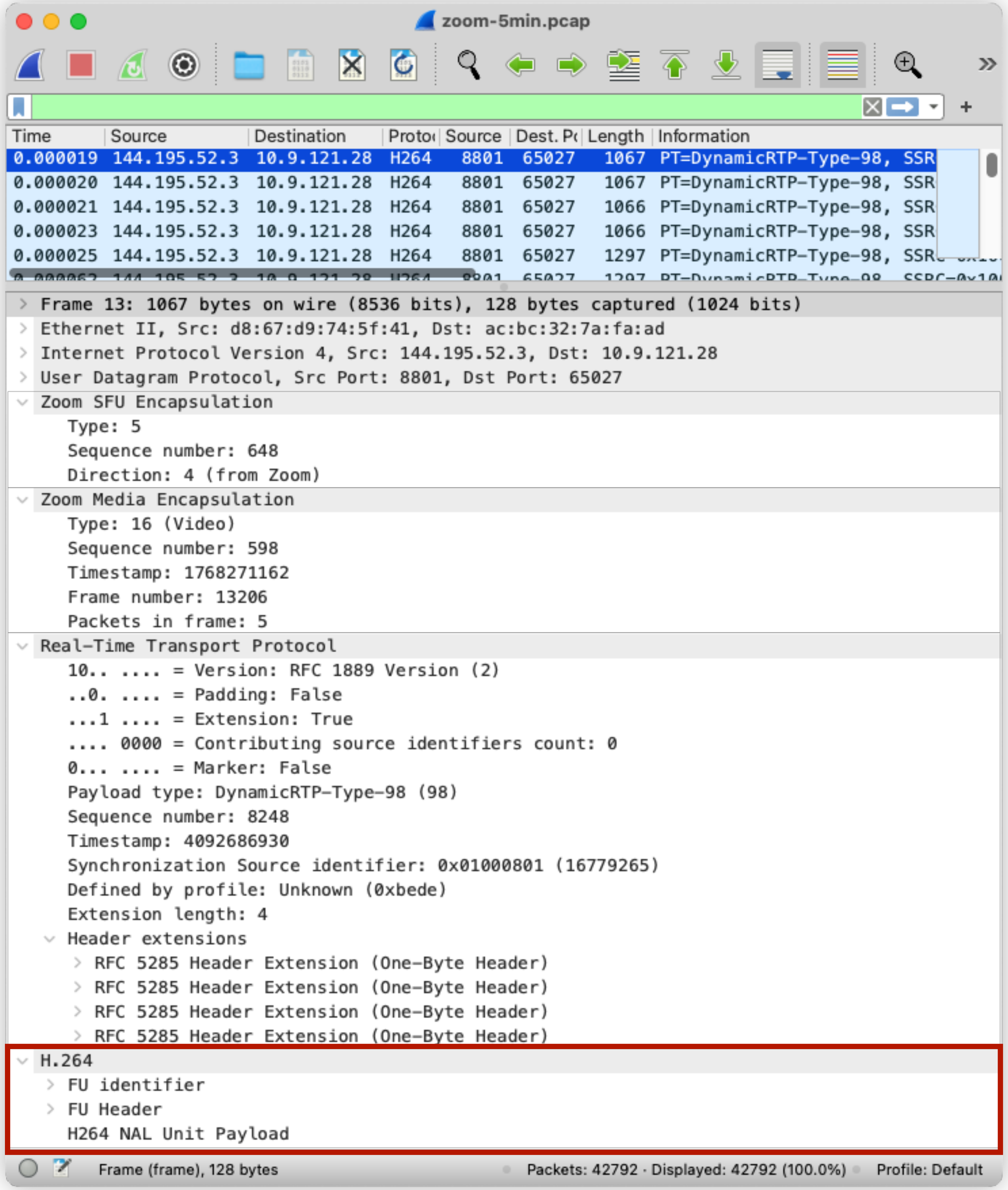
RTP/RTCP



Demystifying Zoom

2 Inferring Header Formats via Entropy Analysis





Demystifying Zoom

2 Inferring Header Formats via Entropy Analysis

...

UDP

Server/SFU Encapsulation

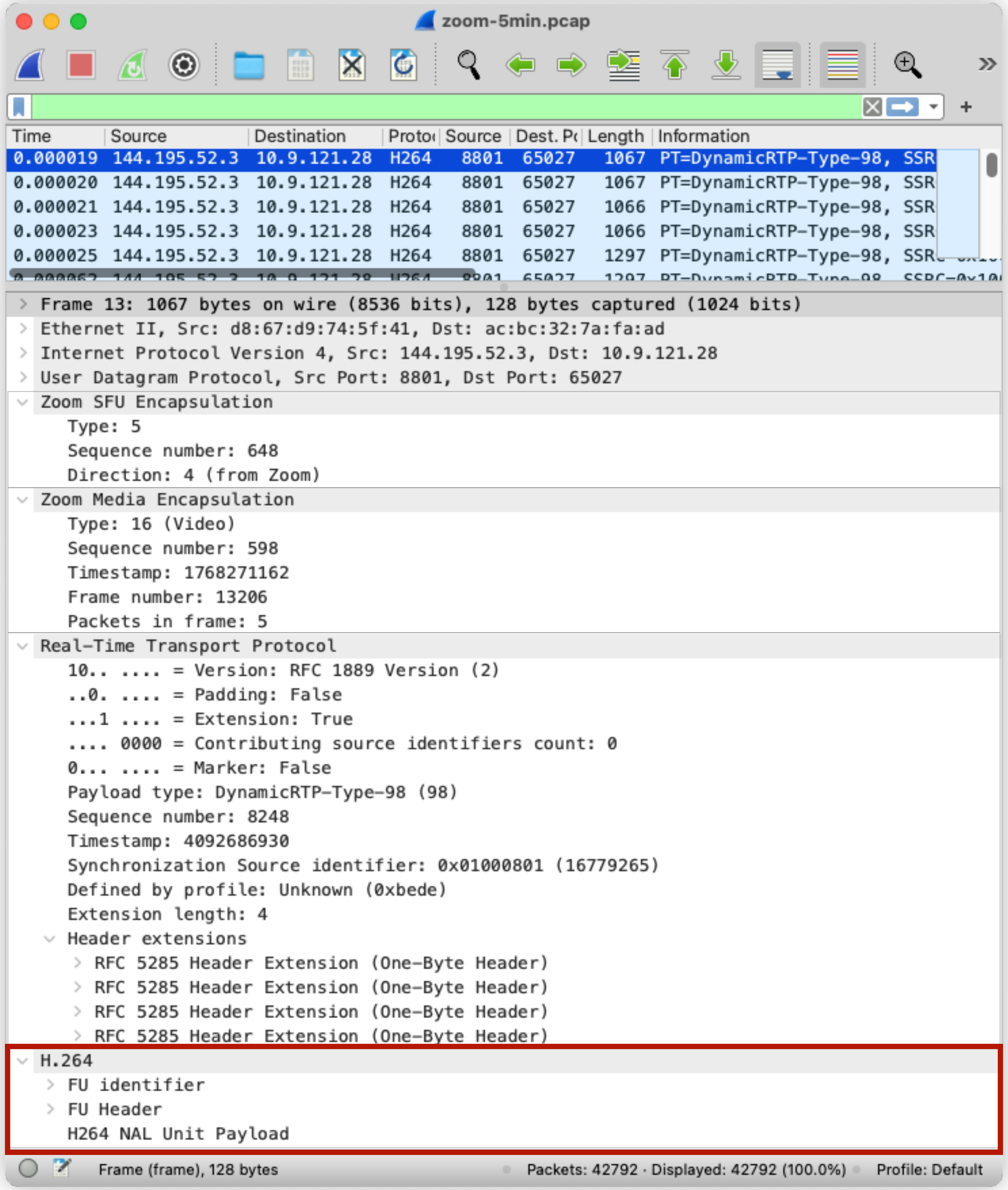
Media Encapsulation

RTP/RTCP

Media Header (H.264 for Video)

Media (encrypted)

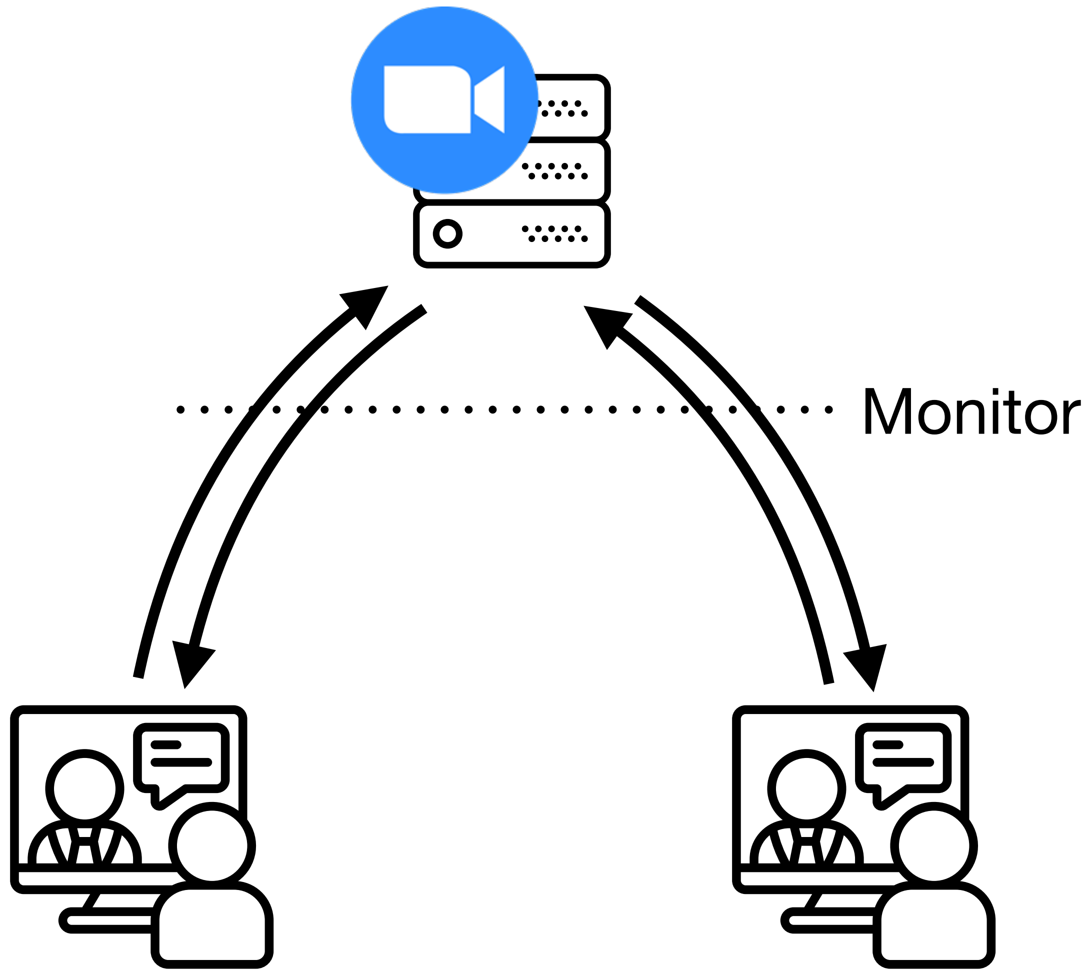
Wireshark Plugin for Zoom
(part of paper artifacts)



Demystifying Zoom

3 Grouping Packets by Meeting

Knowledge about individual streams not sufficient, e.g. for latency measurement

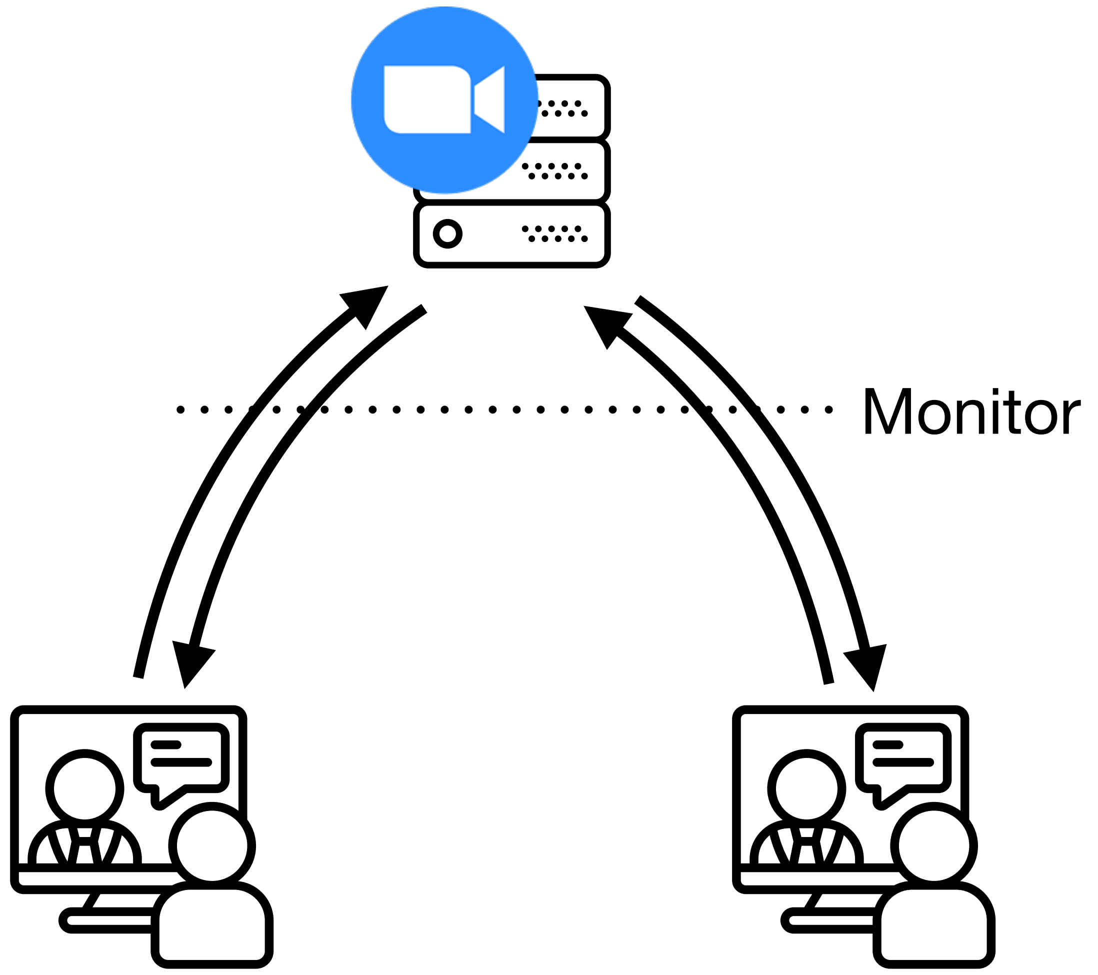


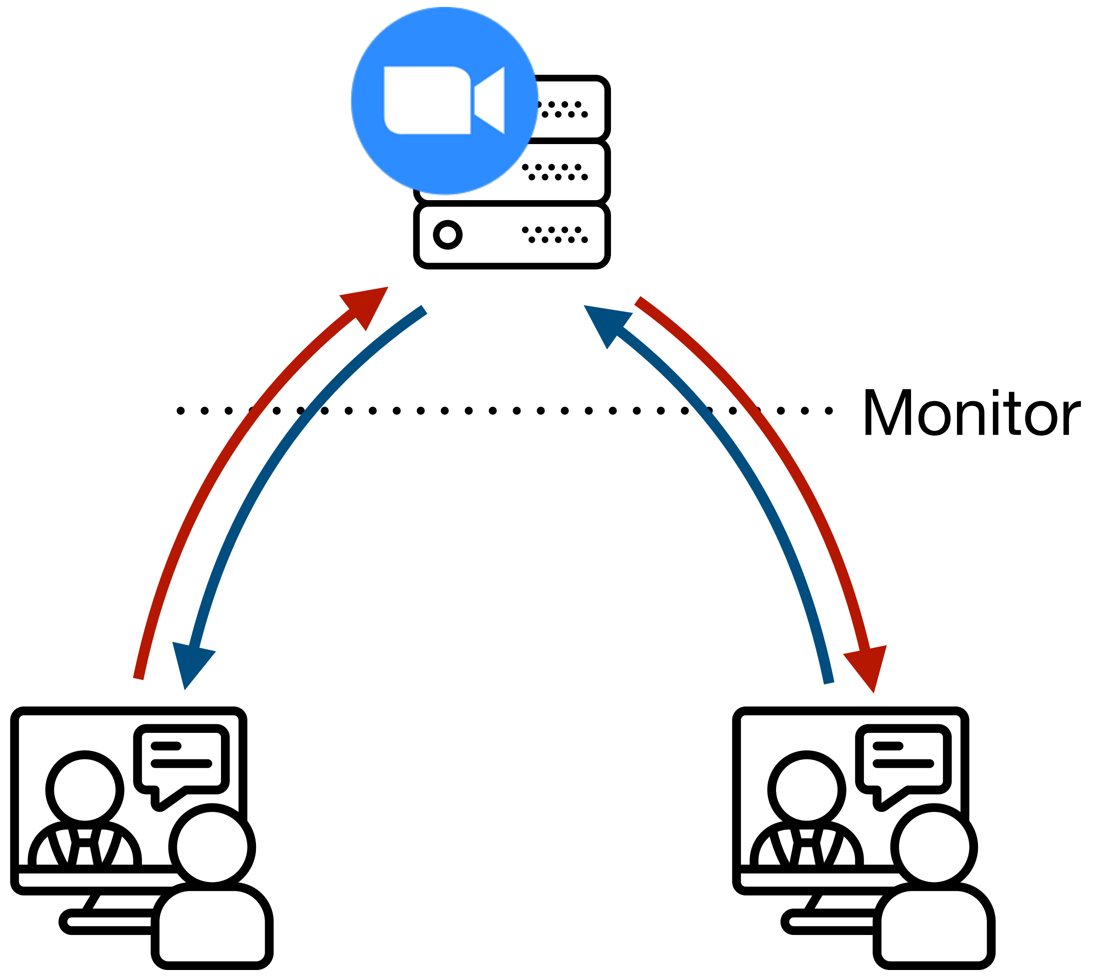
Meeting Topology

Demystifying Zoom

3 Grouping Packets by Meeting

Knowledge about individual streams not sufficient, e.g. for latency measurement



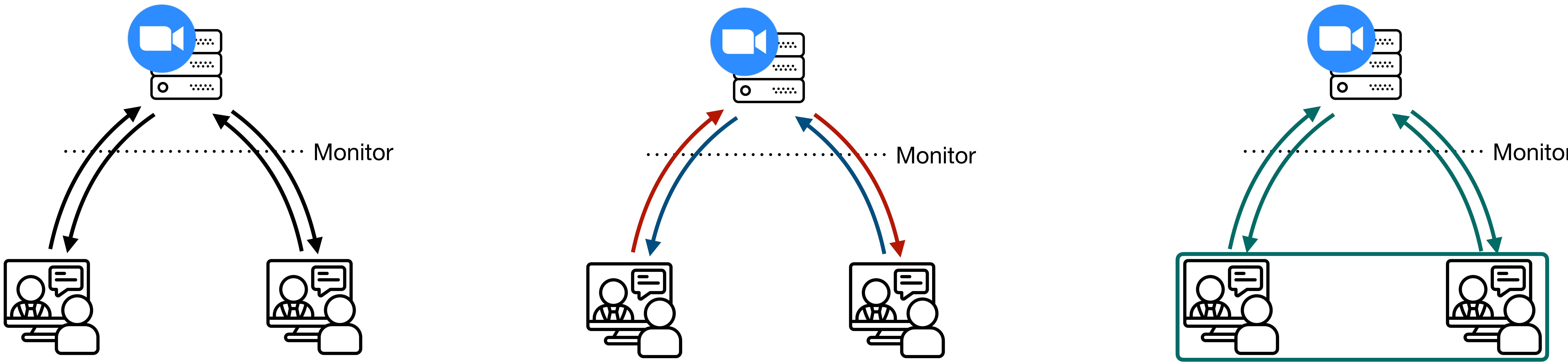


Meeting Topology → (1) Finding “Duplicate” Streams

Demystifying Zoom

3 Grouping Packets by Meeting

Knowledge about individual streams not sufficient, e.g. for latency measurement

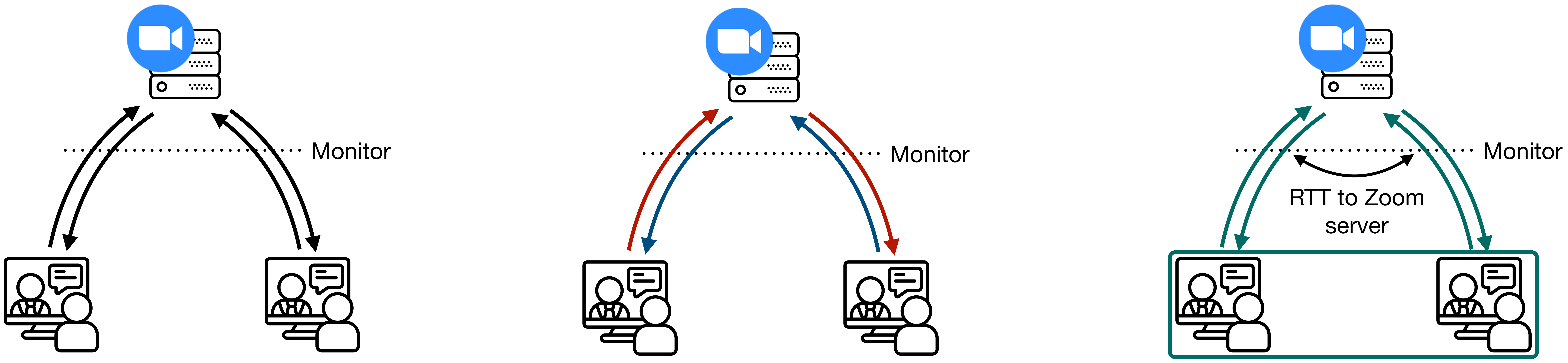


Meeting Topology → (1) Finding "Duplicate" Streams → (2) Assigning Streams to Meetings

Demystifying Zoom

3 Grouping Packets by Meeting

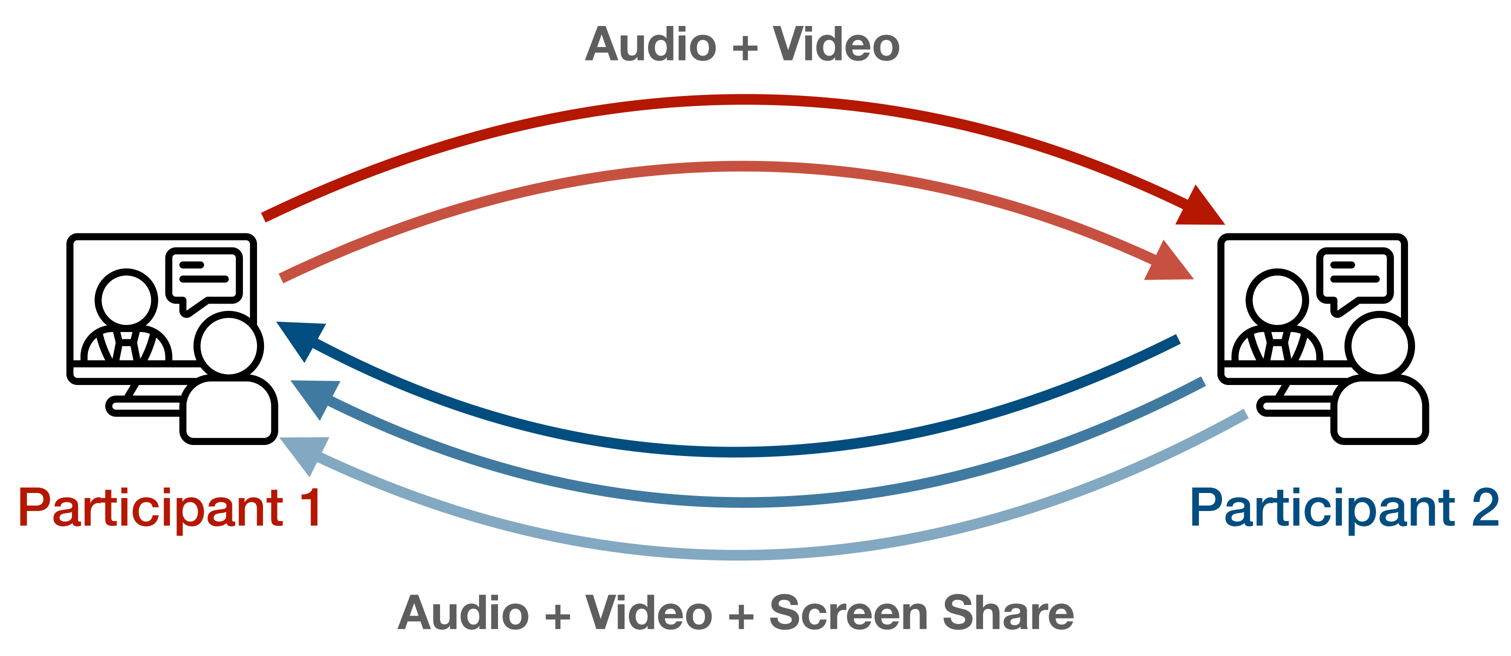
Knowledge about individual streams not sufficient, e.g. for latency measurement



Meeting Topology → (1) Finding “Duplicate” Streams → (2) Assigning Streams to Meetings

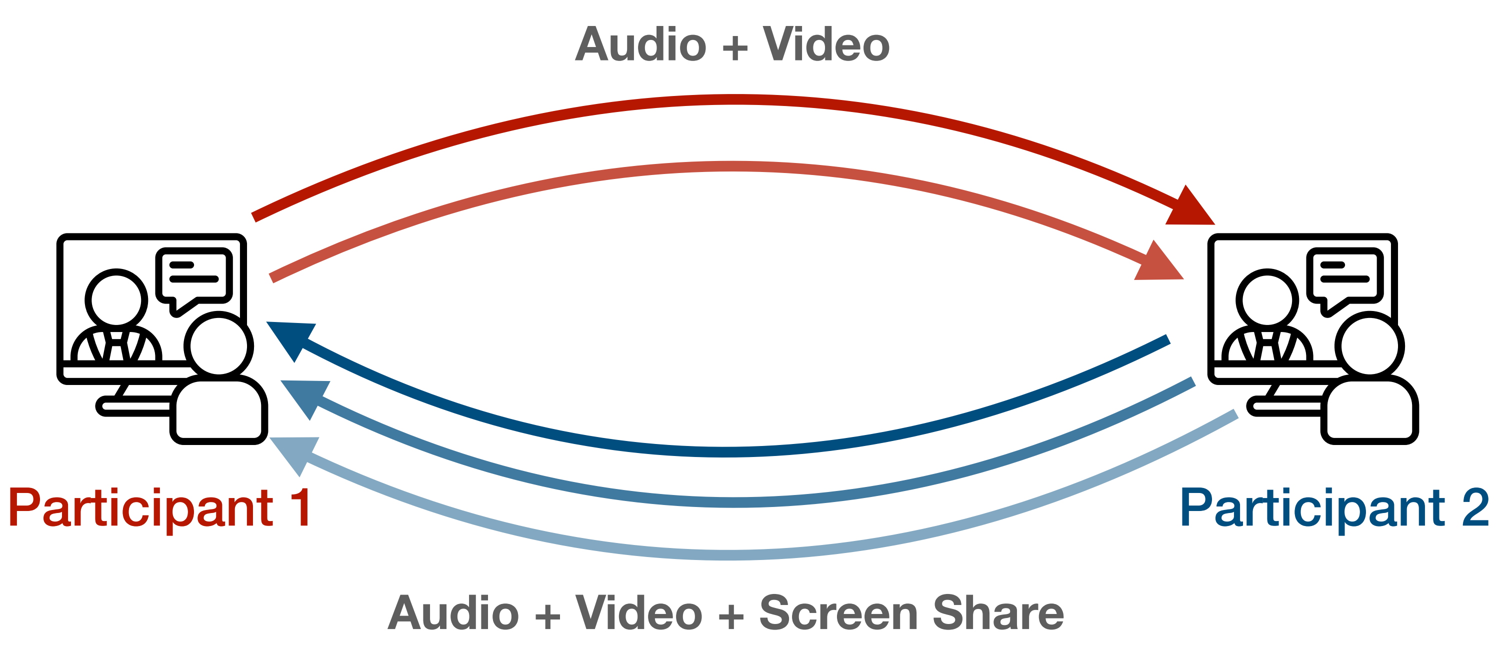
Demystifying Zoom

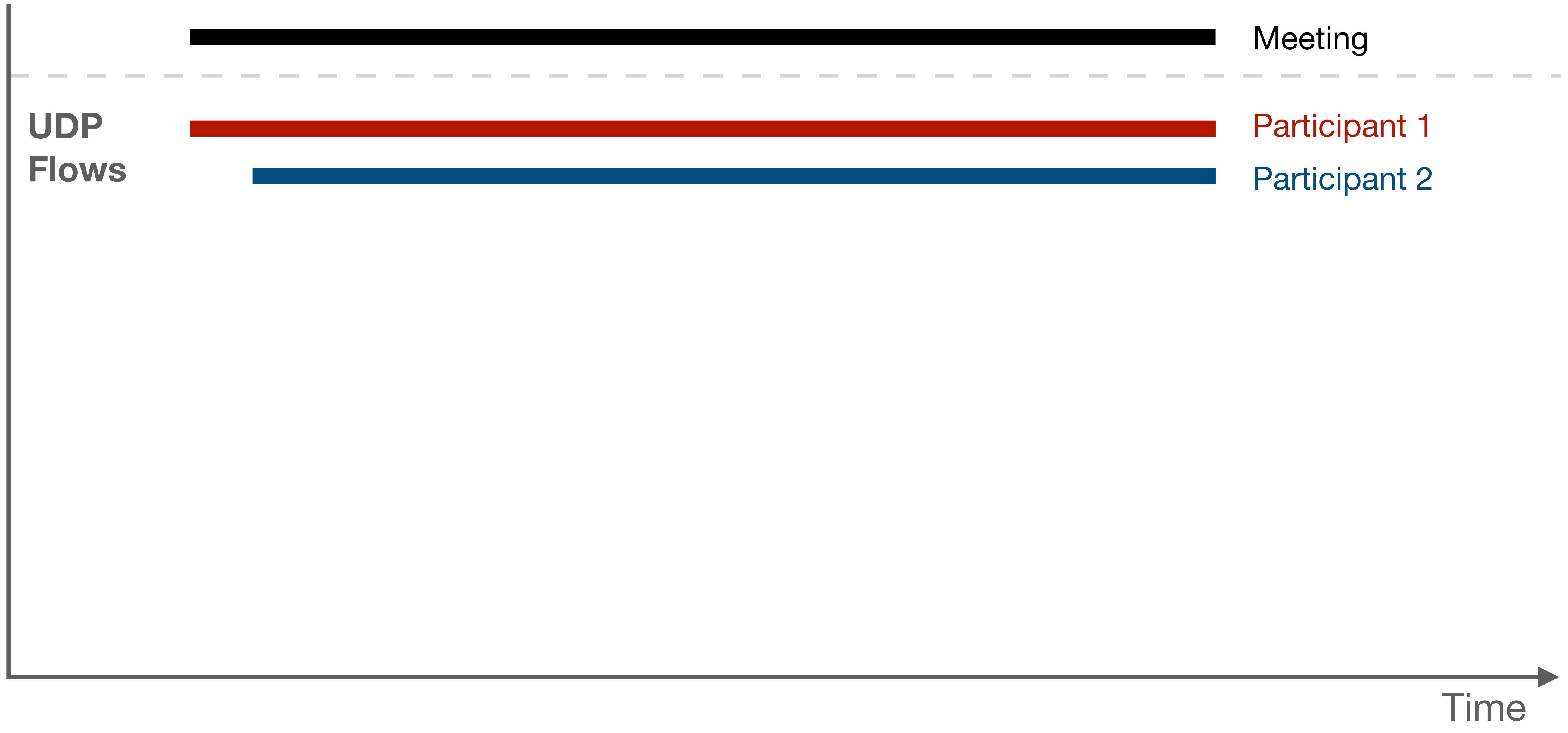
3 Grouping Packets by Meeting



Demystifying Zoom

3 Grouping Packets by Meeting

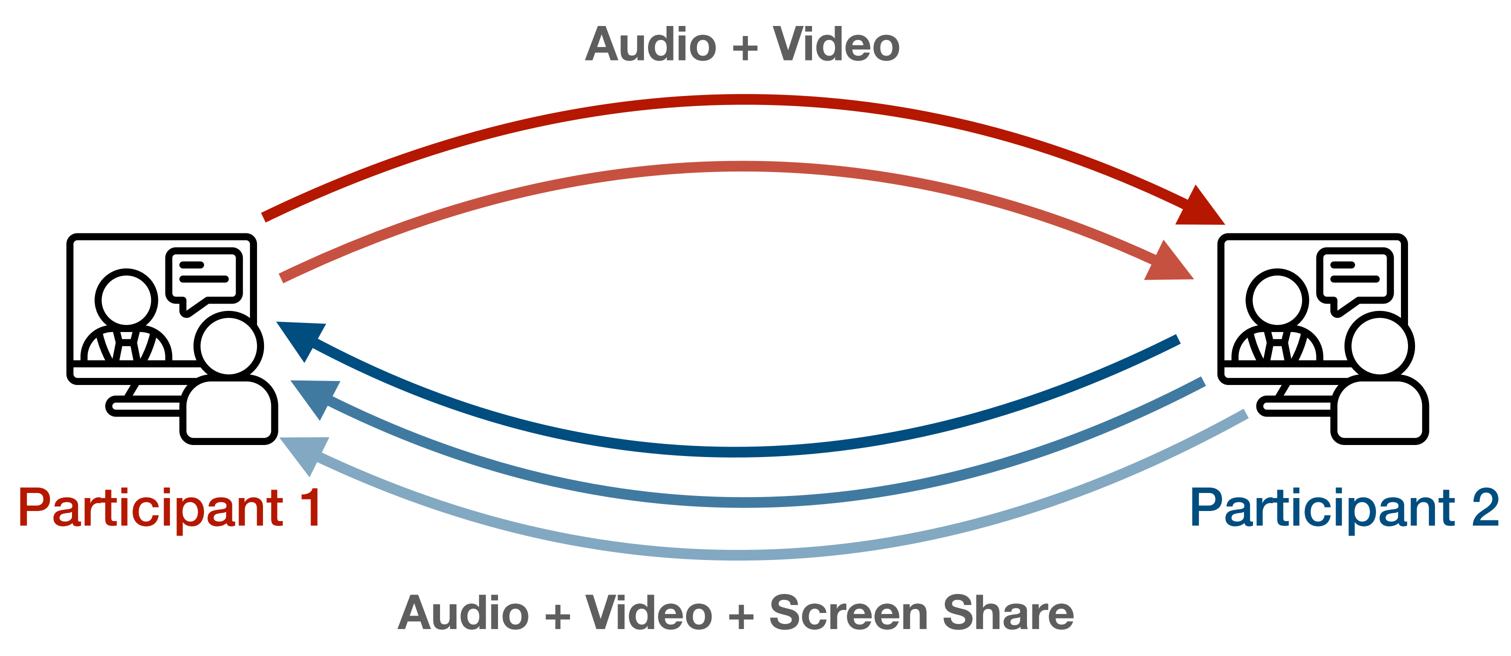


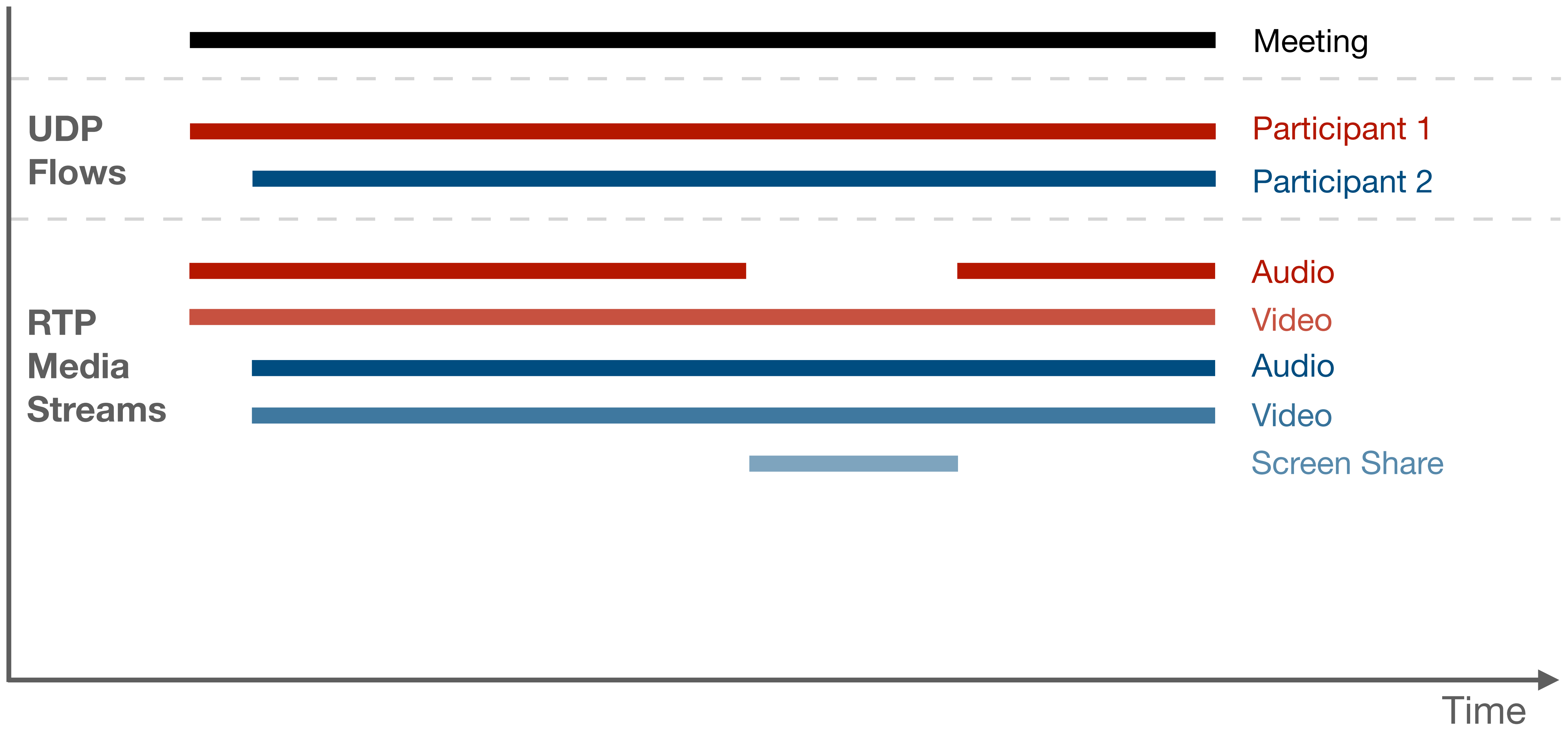


Stream Activity

Demystifying Zoom

3 Grouping Packets by Meeting

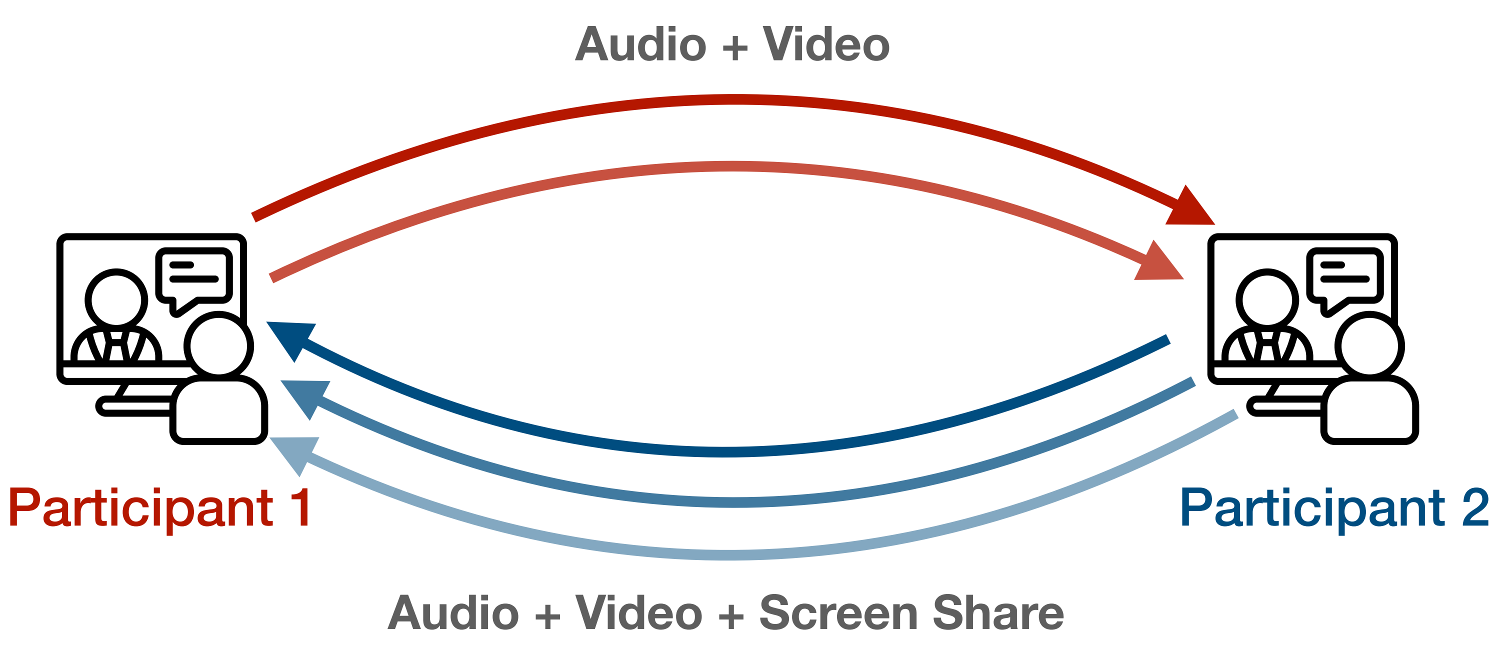


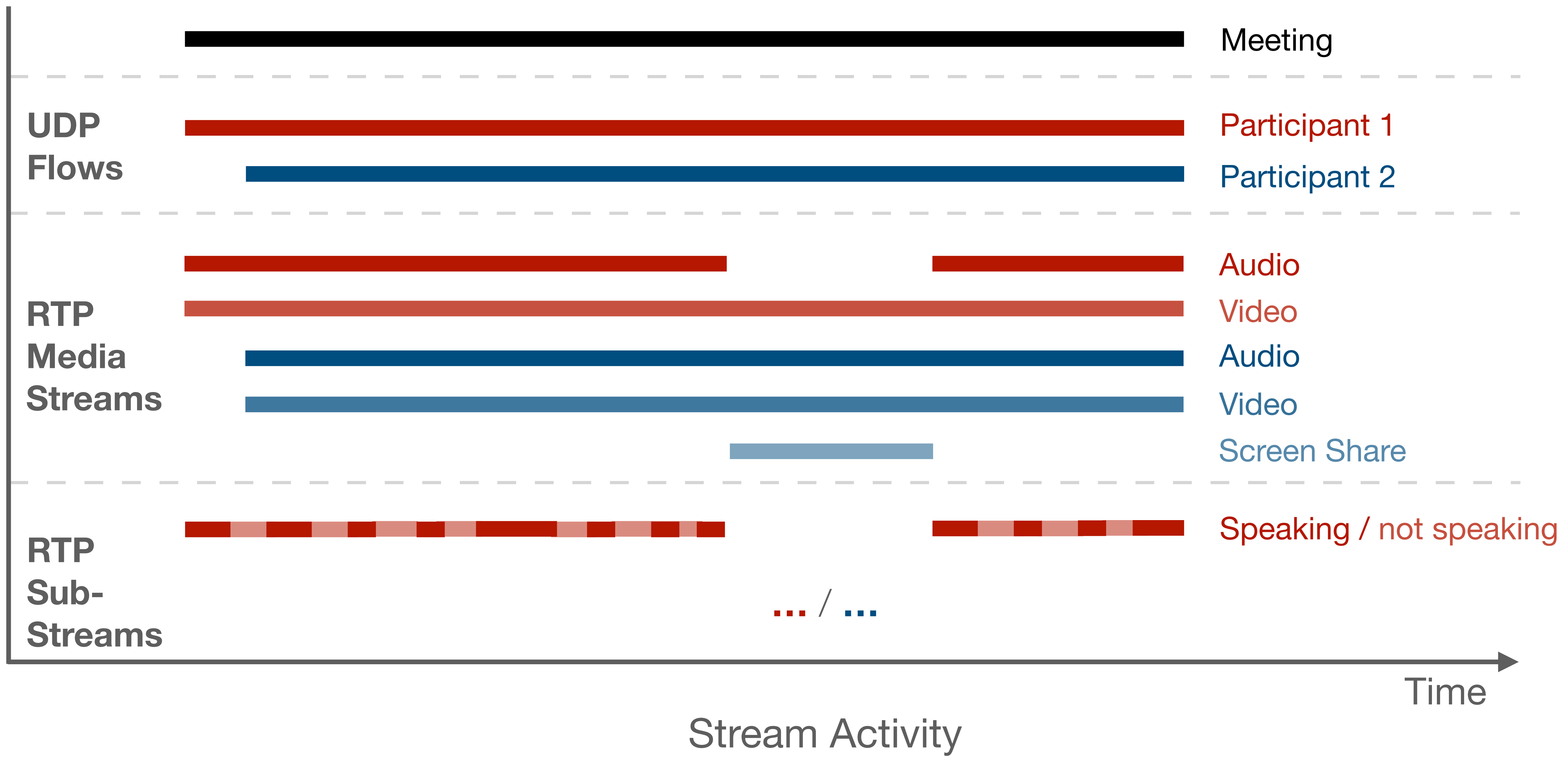


Stream Activity

Demystifying Zoom

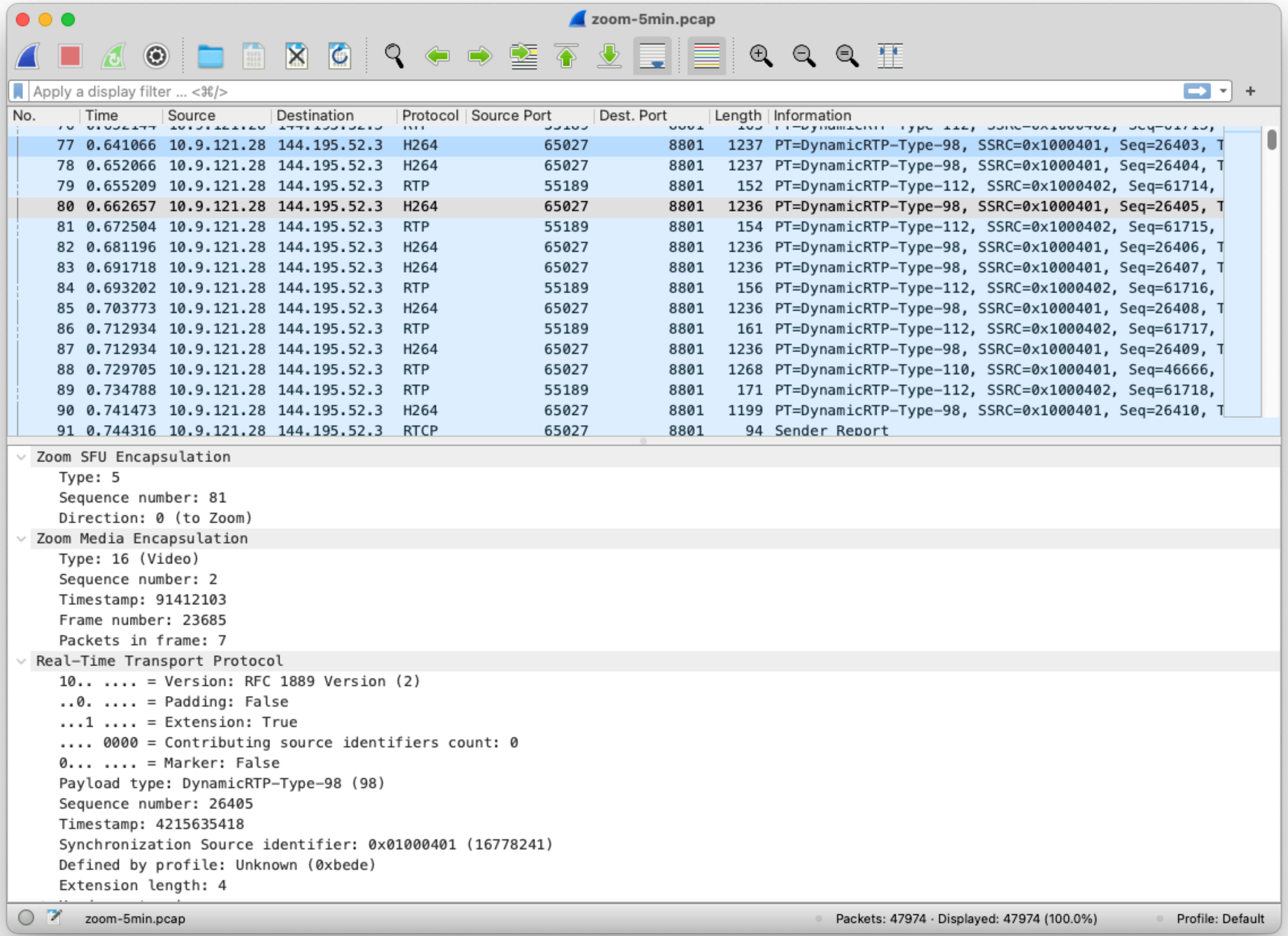
3 Grouping Packets by Meeting





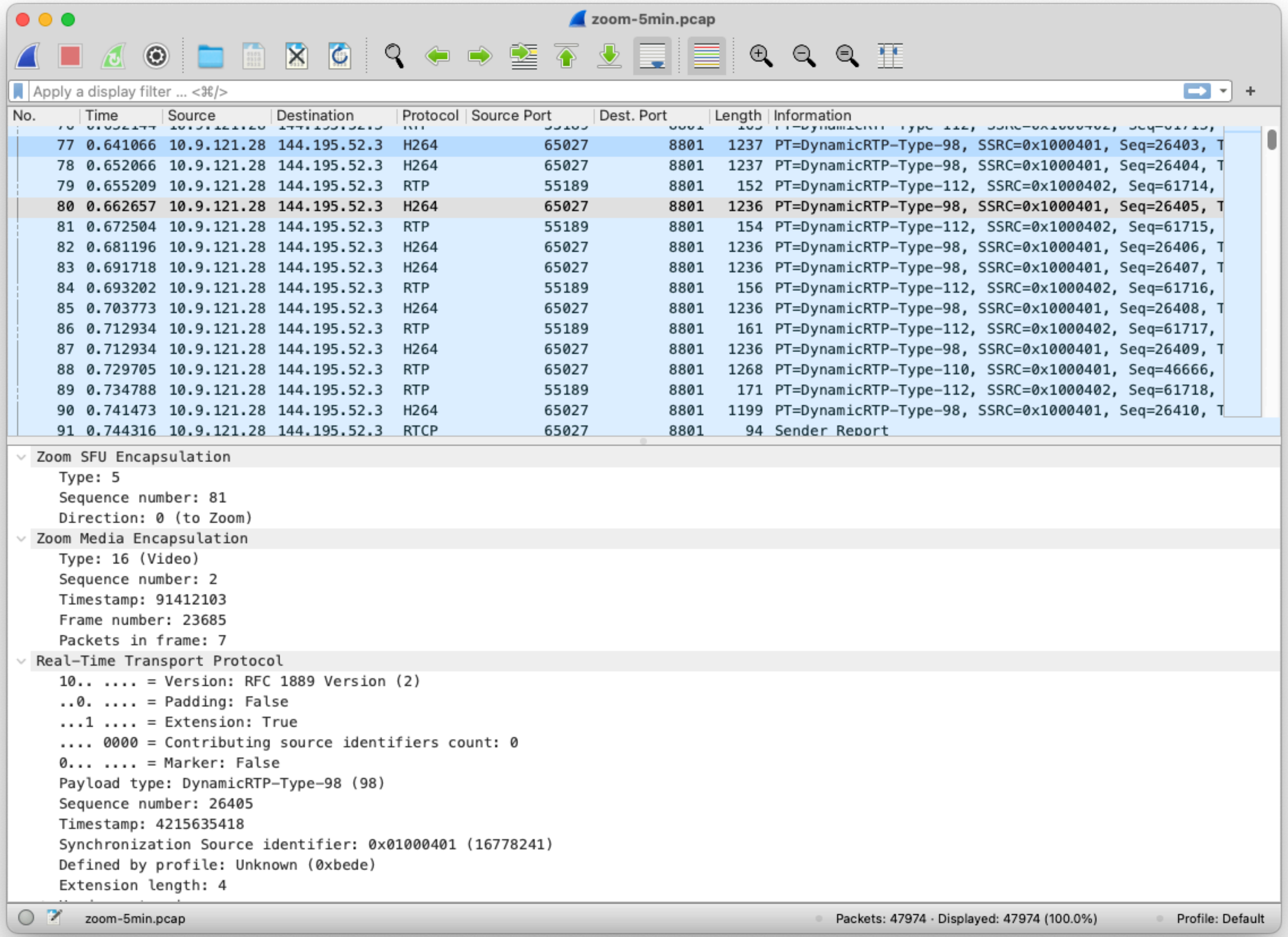
Measuring Zoom Performance

Packet Format → Metrics



Measuring Zoom Performance

Packet Format → Metrics

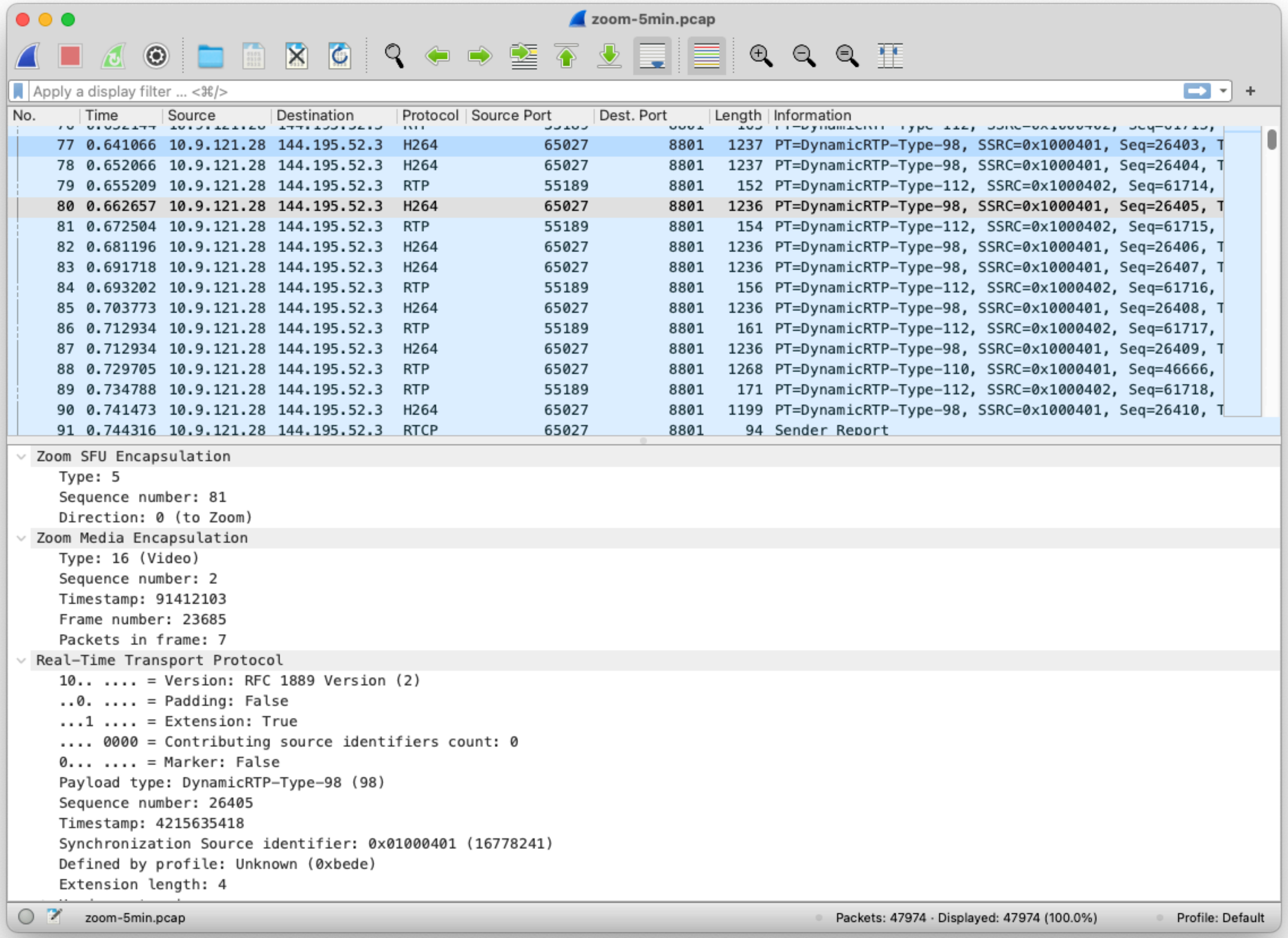


→

Metric
Media Bit Rate
Frame Rate
Frame Size
Latency
Jitter
Loss, RTX, out-of-order

Measuring Zoom Performance

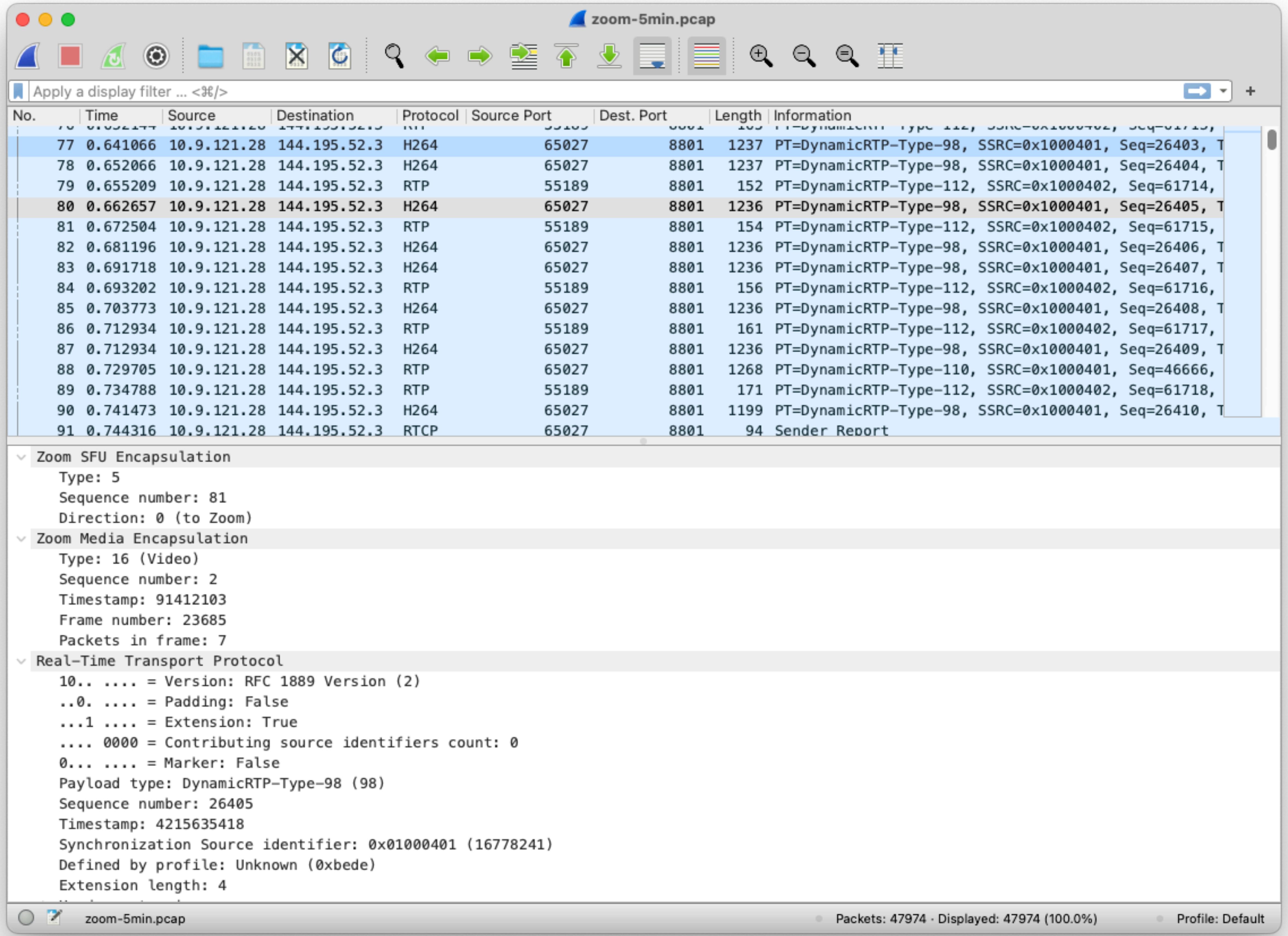
Packet Format → Metrics



Metric	Requires Headers
Media Bit Rate	●
Frame Rate	●
Frame Size	●
Latency	●
Jitter	●
Loss, RTX, out-of-order	●

Measuring Zoom Performance

Packet Format → Metrics

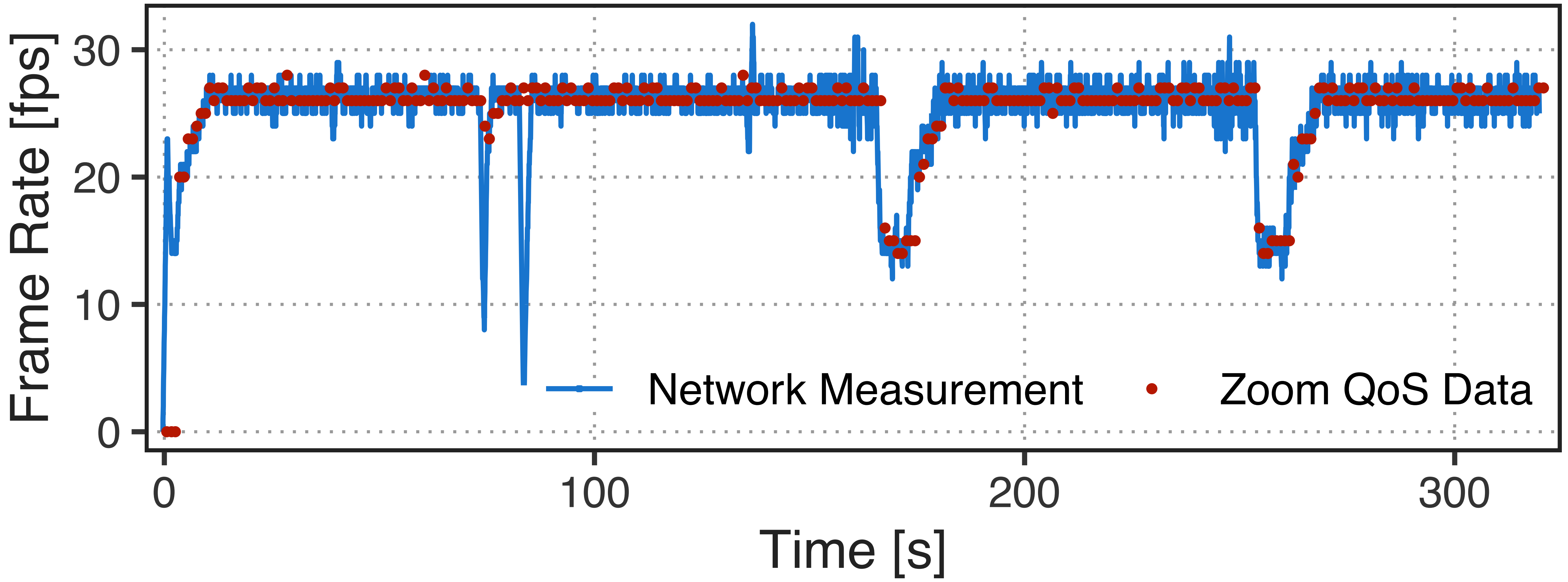


Metric	Requires Headers	Available in Zoom SDK	Validated
Media Bit Rate	●		
Frame Rate	●	●	●
Frame Size	●		
Latency	●	●	●
Jitter	●	●	●
Loss, RTX, out-of-order	●		

Measuring Zoom Performance

Frame Rate Validation

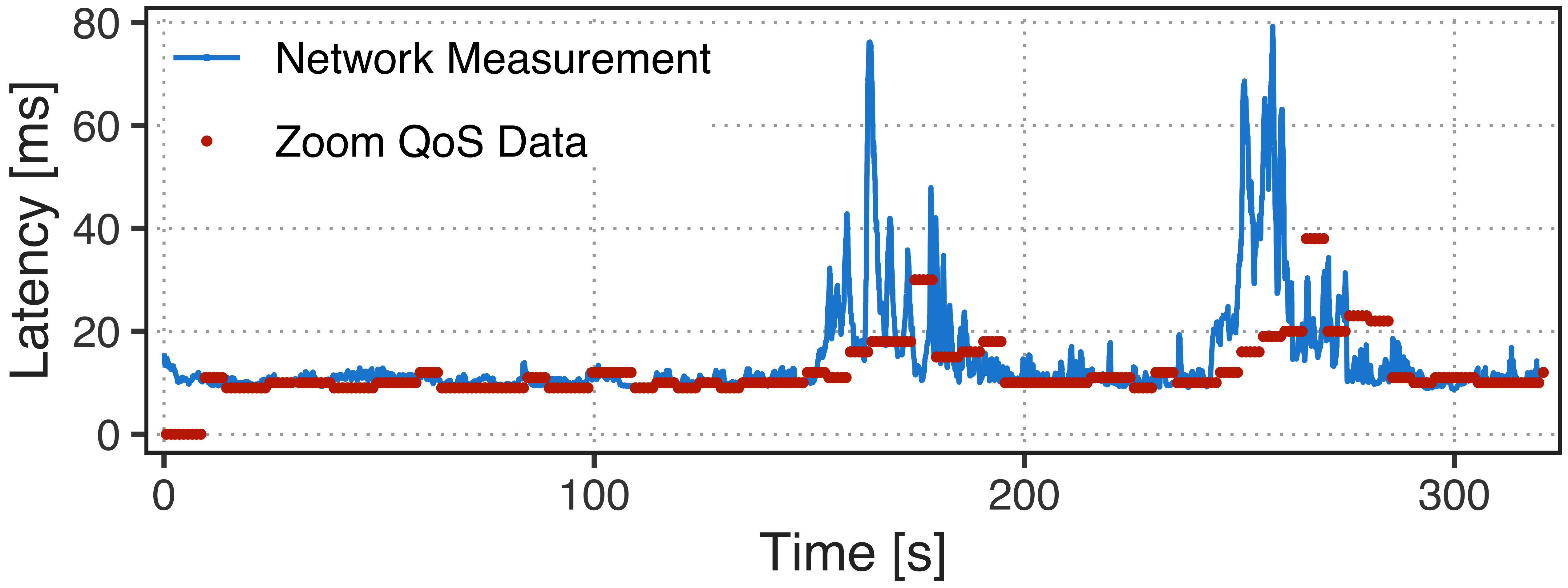
- Controlled experiments with instrumented Zoom Client using Zoom SDK
- Frame rate measurement matches Zoom QoS data exactly
- Finer-grained data, detect short-term variations (< 1s)
  (1 per frame vs. 1 per sec. → up to 30/sec.)



Measuring Zoom Performance

Latency + Frame-level Jitter Validation

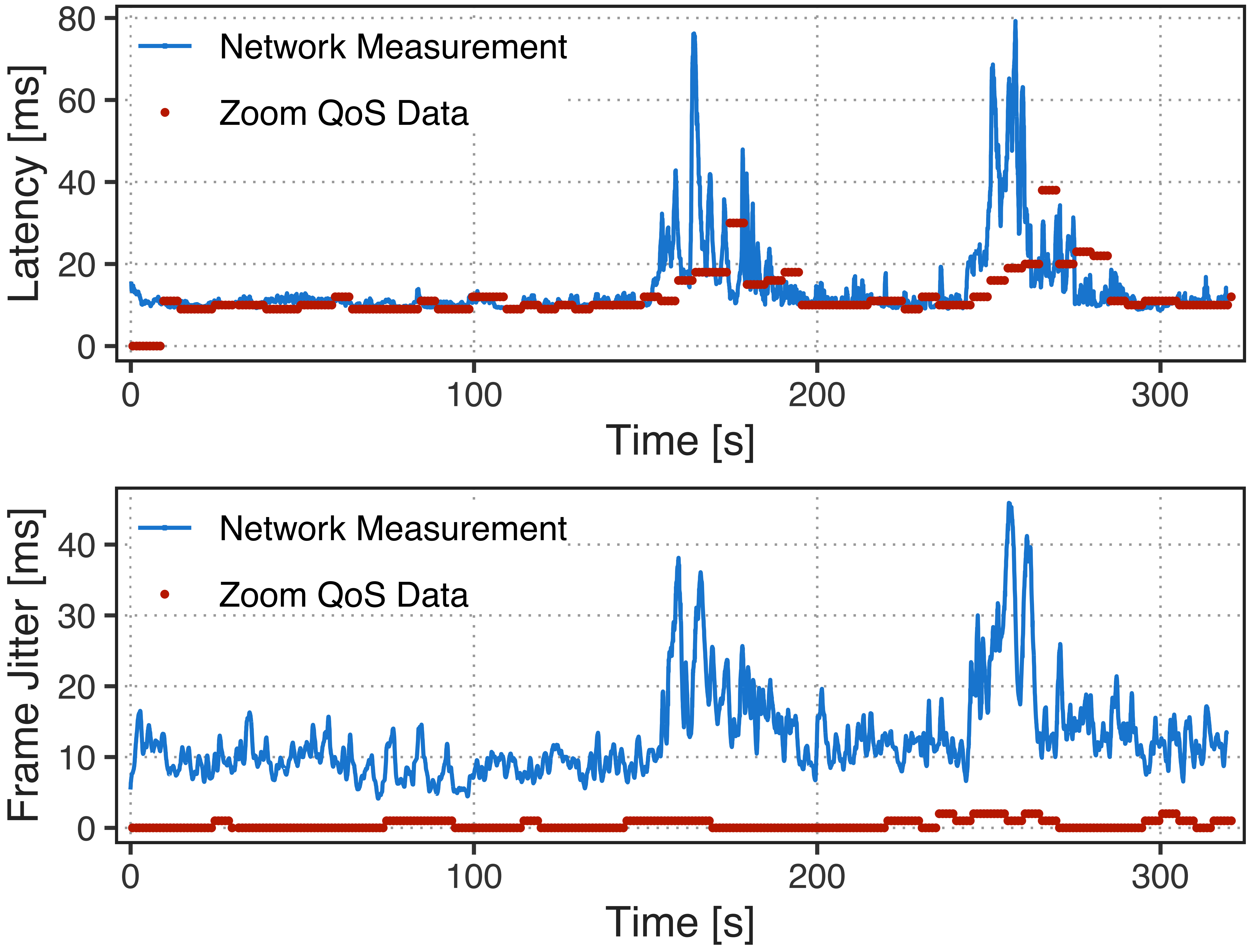
- Latency to Zoom SFU measured by matching RTP sequence numbers
- Latency measurements match despite long smoothing
- Finer-grained measurement
  (1 per 5 sec. vs. 1 per packet → 100s per sec.)



Measuring Zoom Performance

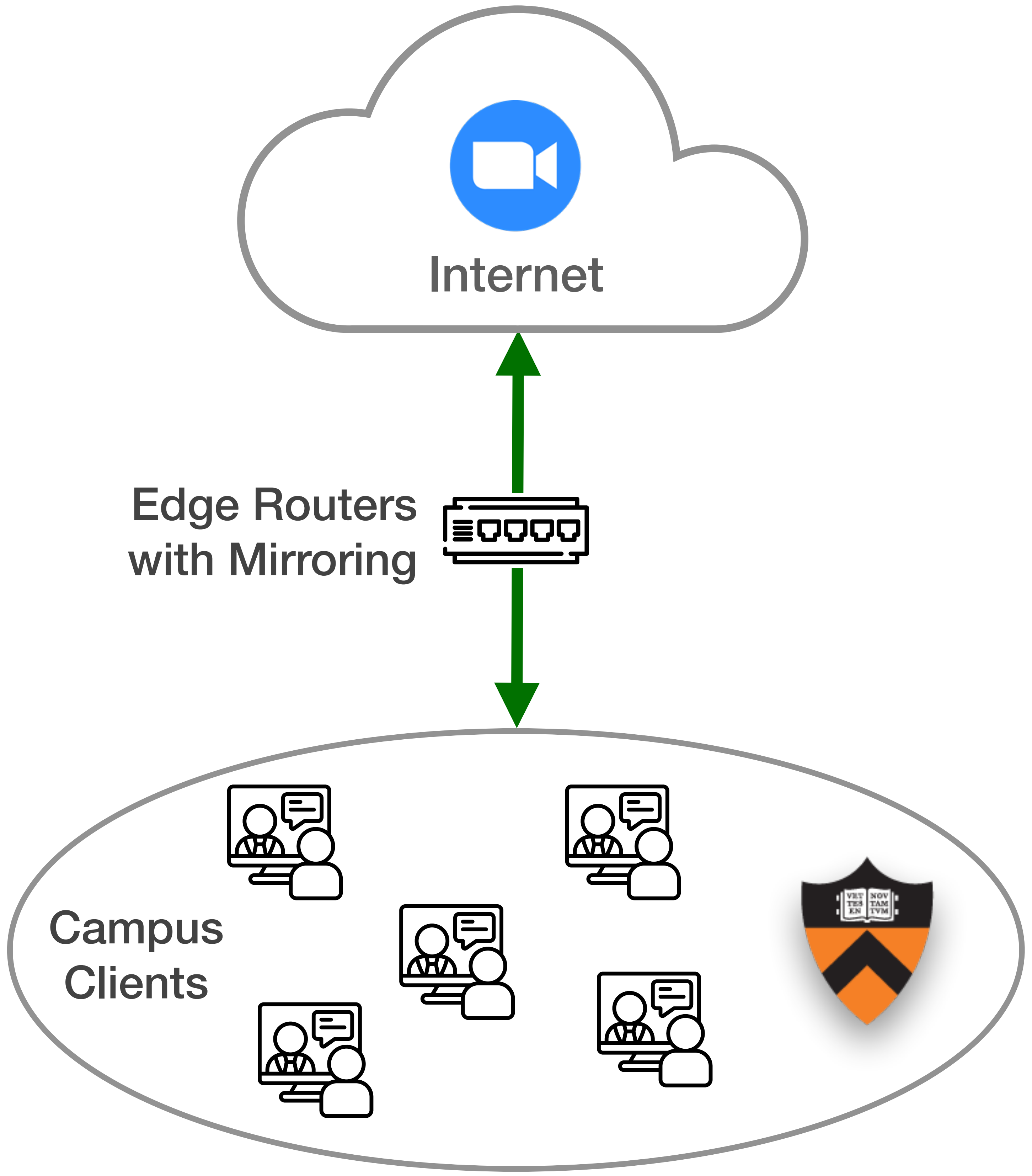
Latency + Frame-level Jitter Validation

- Latency to Zoom SFU measured by matching RTP sequence numbers
- Latency measurements match despite long smoothing
- Finer-grained measurement
  (1 per 5 sec. vs. 1 per packet → 100s per sec.)
- Discrepancy in jitter



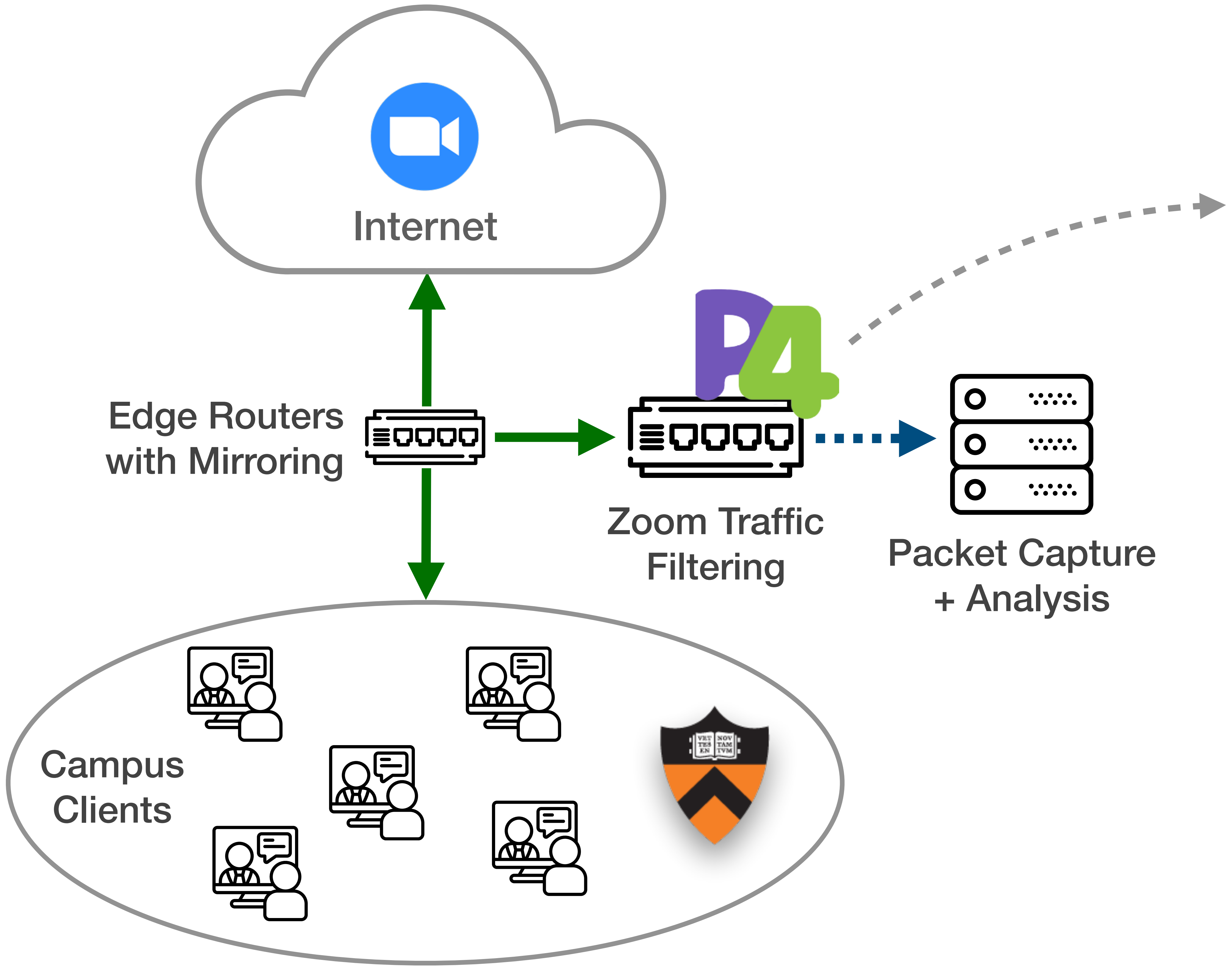
Analyzing Zoom Campus Traffic

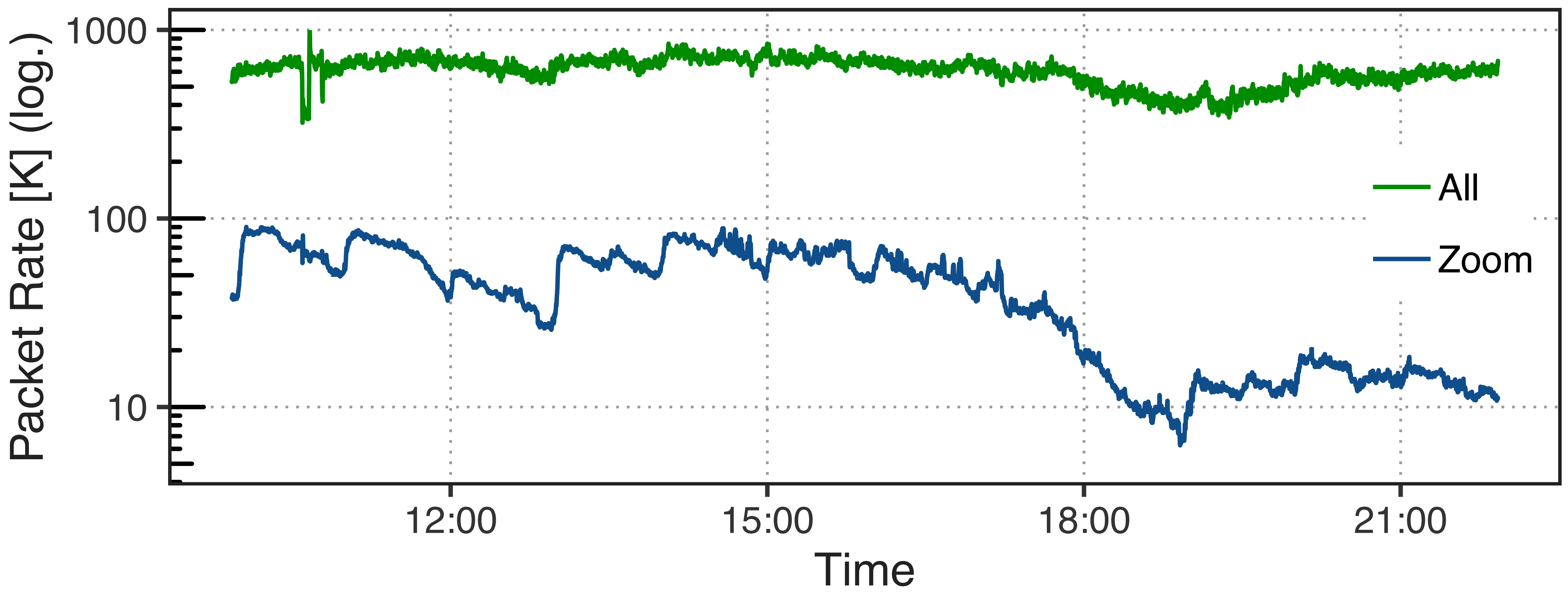
Traffic Capture



Analyzing Zoom Campus Traffic

Traffic Capture





Analyzing Zoom Campus Traffic

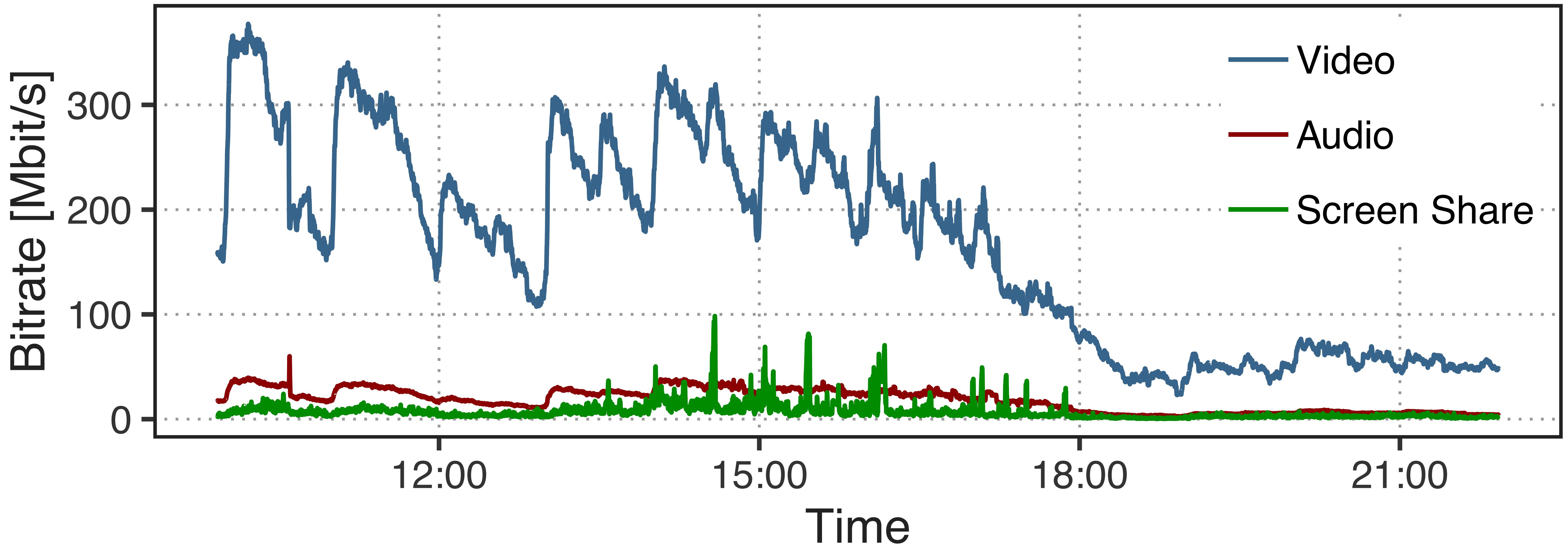
Data Rate per Media Type

Capture Duration	12 hours
Zoom Packets	1,846 M (~43K/s)
Zoom Data Rate	222.9 Mbit/s avg.
Zoom RTP Media Streams	59,020

Analyzing Zoom Campus Traffic

Data Rate per Media Type

Capture Duration	12 hours
Zoom Packets	1,846 M (~43K/s)
Zoom Data Rate	222.9 Mbit/s avg.
Zoom RTP Media Streams	59,020

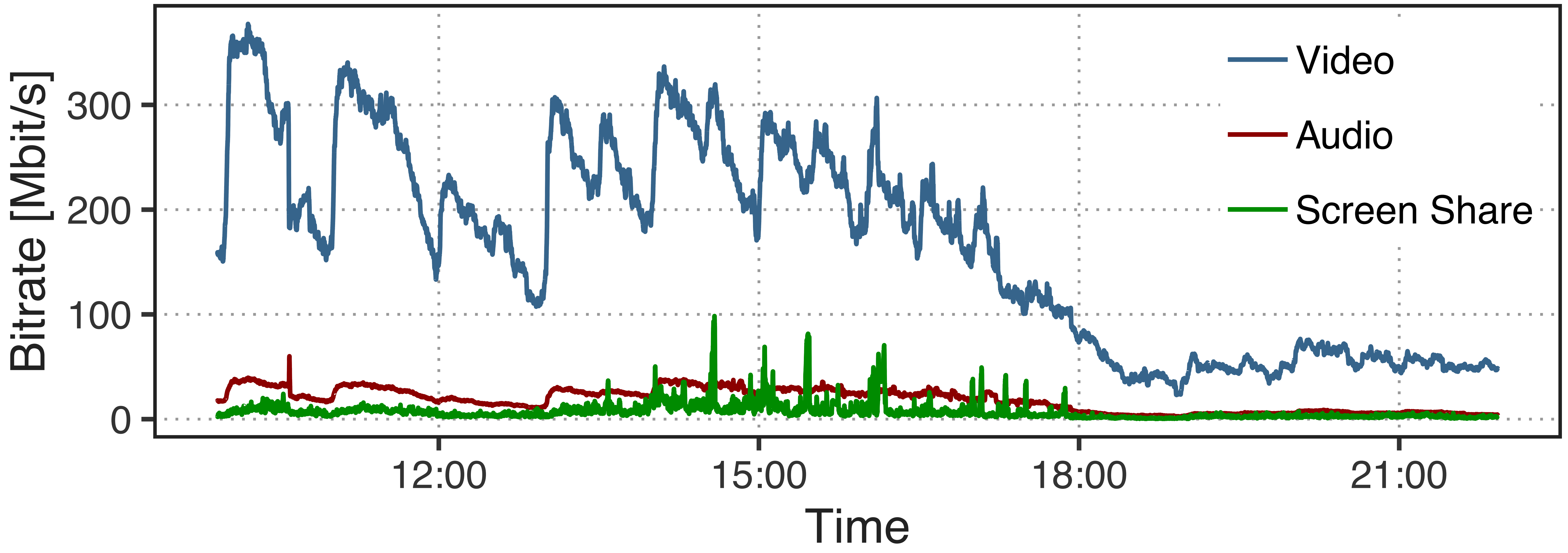


Data Rate of Zoom Media Streams by Media Type

Analyzing Zoom Campus Traffic

Data Rate per Media Type

Capture Duration	12 hours
Zoom Packets	1,846 M (~43K/s)
Zoom Data Rate	222.9 Mbit/s avg.
Zoom RTP Media Streams	59,020



Data Rate of Zoom Media Streams by Media Type

1. Number of participants per meeting

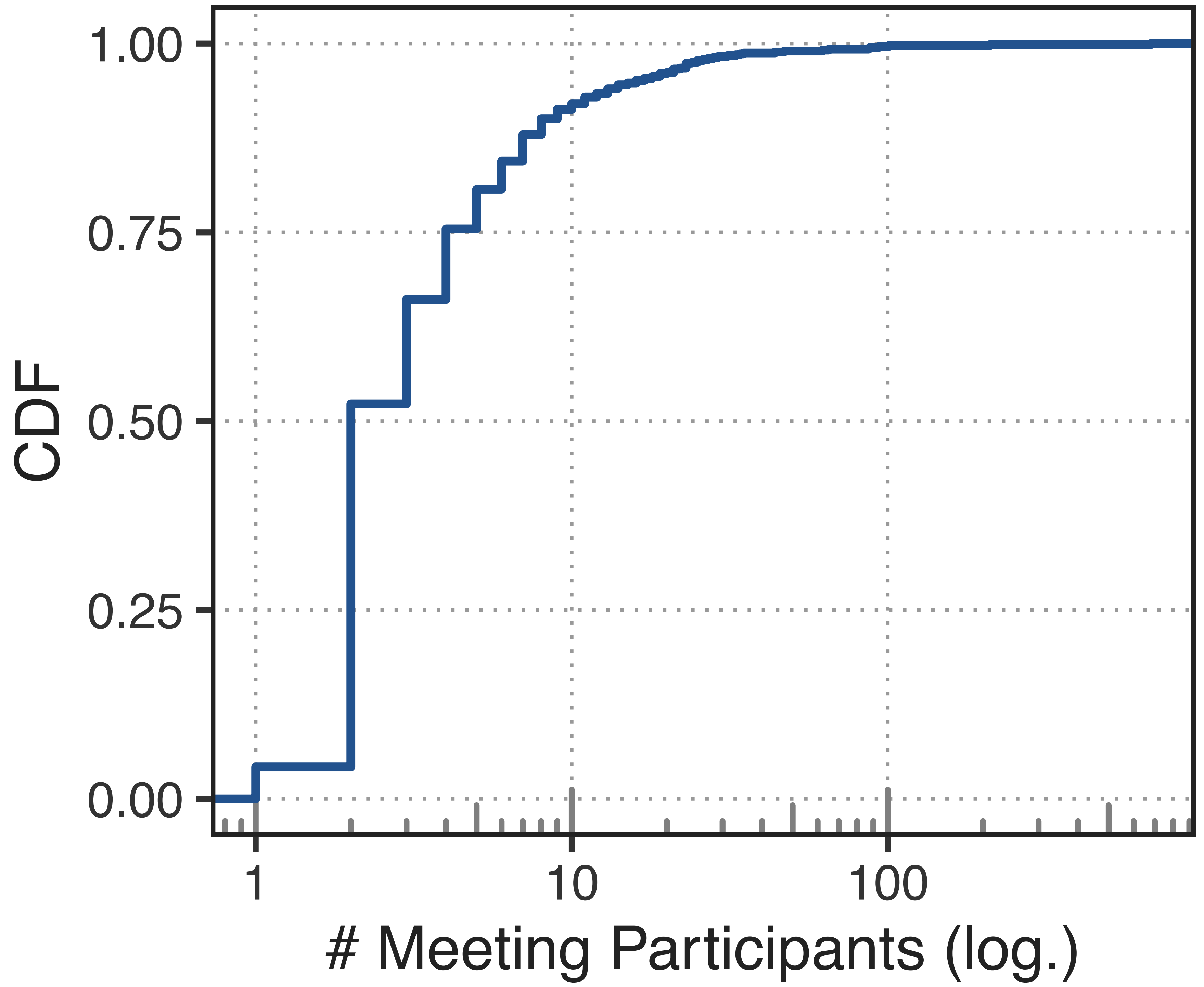
2. RTT to Zoom Server

3. Video Frame Rate

Analyzing Zoom Campus Traffic

Number of Participants per Meeting

- Grouping heuristic to arrange media streams by meeting
- 803 distinct meetings identified

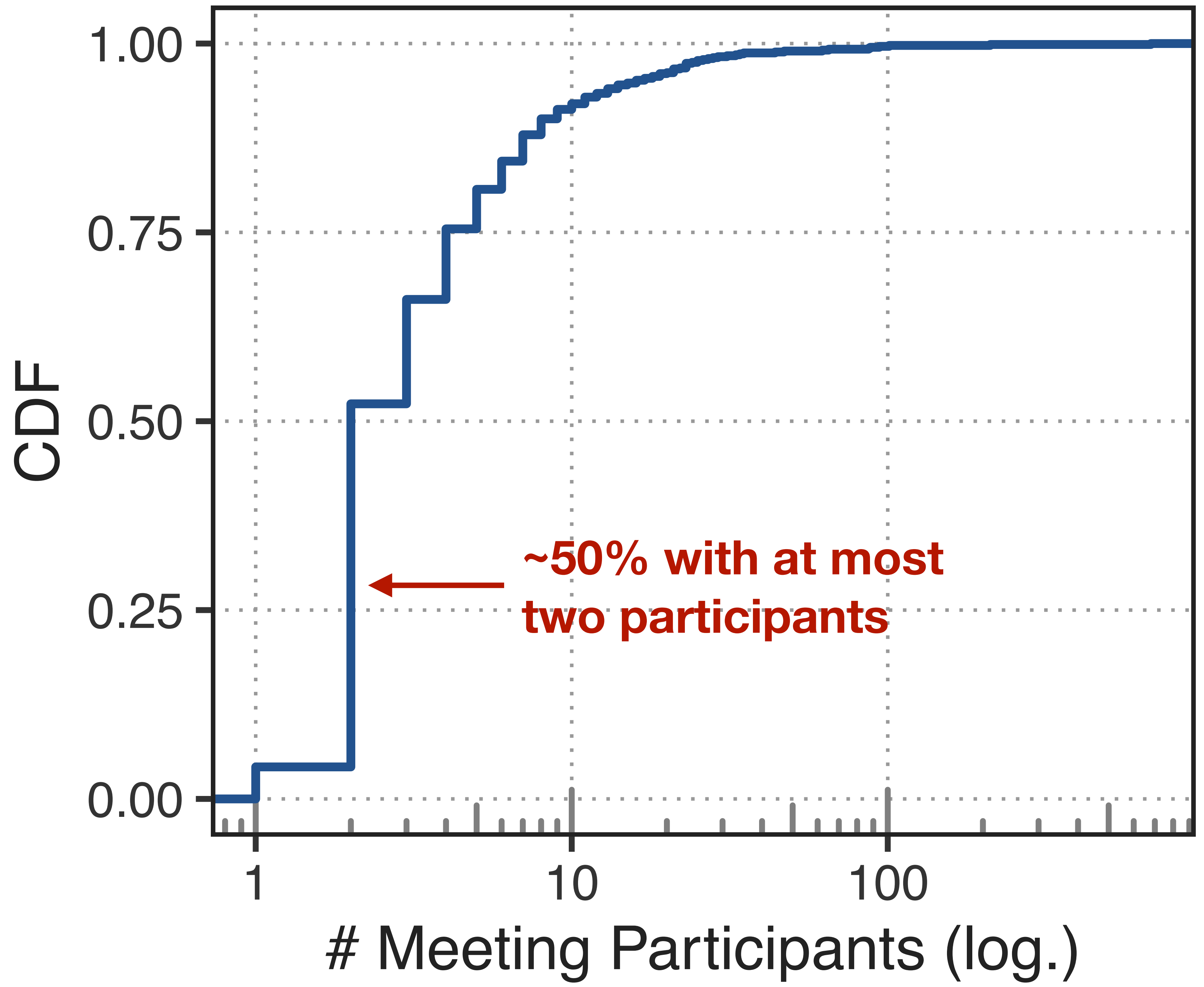


Max. Number of Participants per Meeting in 12-hour Campus Trace.

Analyzing Zoom Campus Traffic

Number of Participants per Meeting

- Grouping heuristic to arrange media streams by meeting
- 803 distinct meetings identified

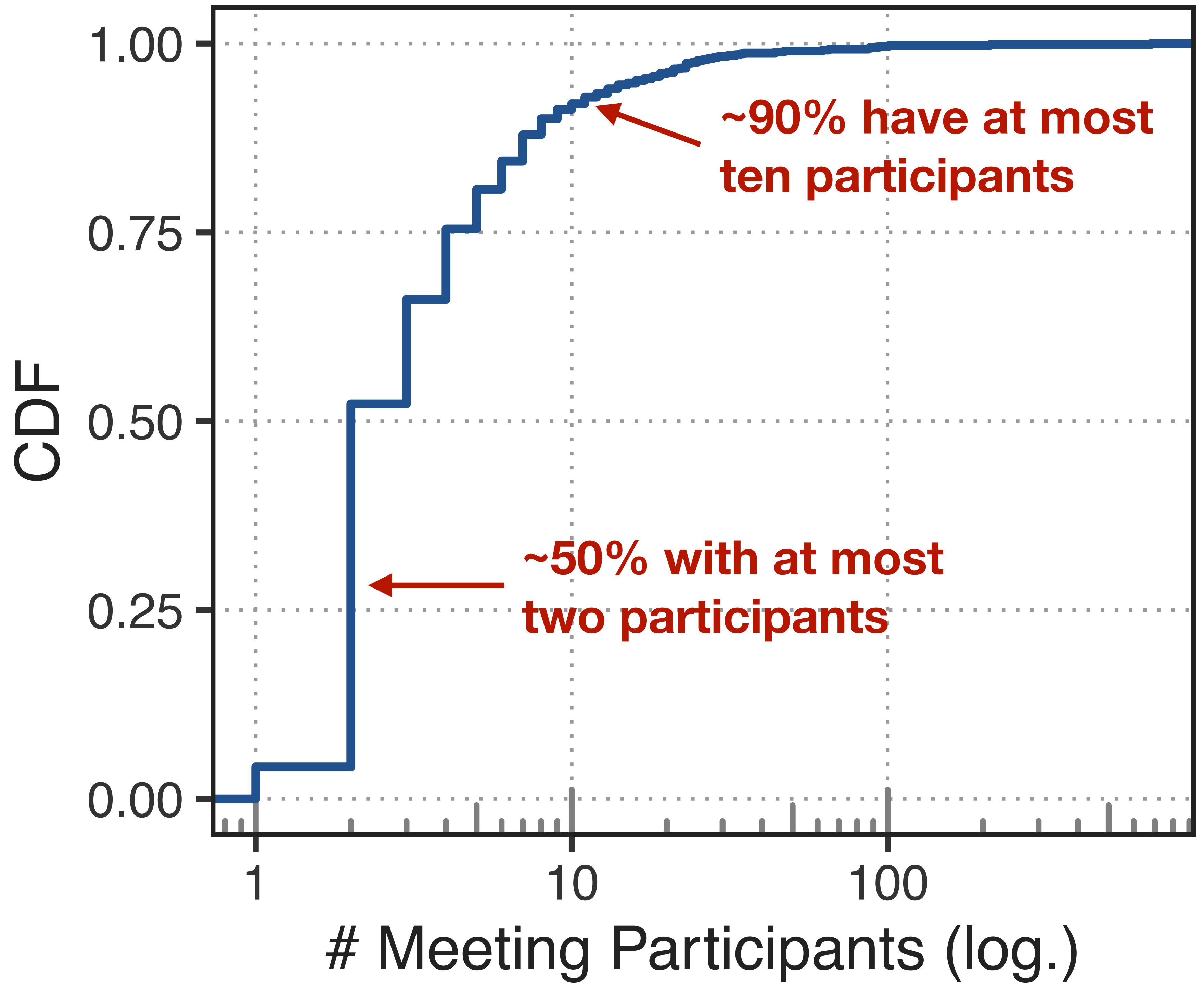


Max. Number of Participants per Meeting in 12-hour Campus Trace.

Analyzing Zoom Campus Traffic

Number of Participants per Meeting

- Grouping heuristic to arrange media streams by meeting
- 803 distinct meetings identified

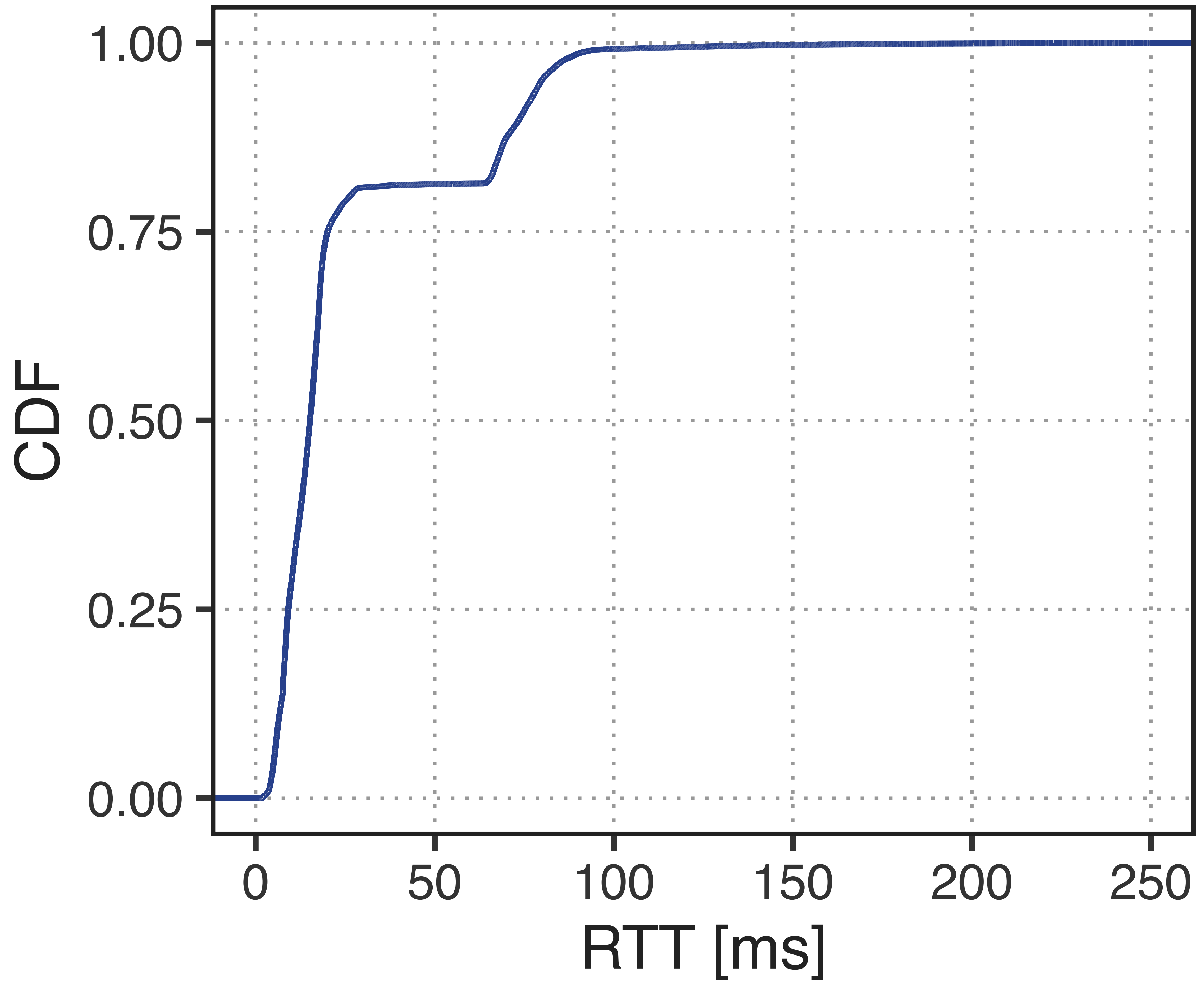


Max. Number of Participants per Meeting in 12-hour Campus Trace.

Analyzing Zoom Campus Traffic

Latency to Zoom Server

- RTT to server by matching RTP sequence numbers
- 30.1 M latency samples to 521 distinct Zoom IP addresses
- Vast majority of meetings connected to NYC and California data centers

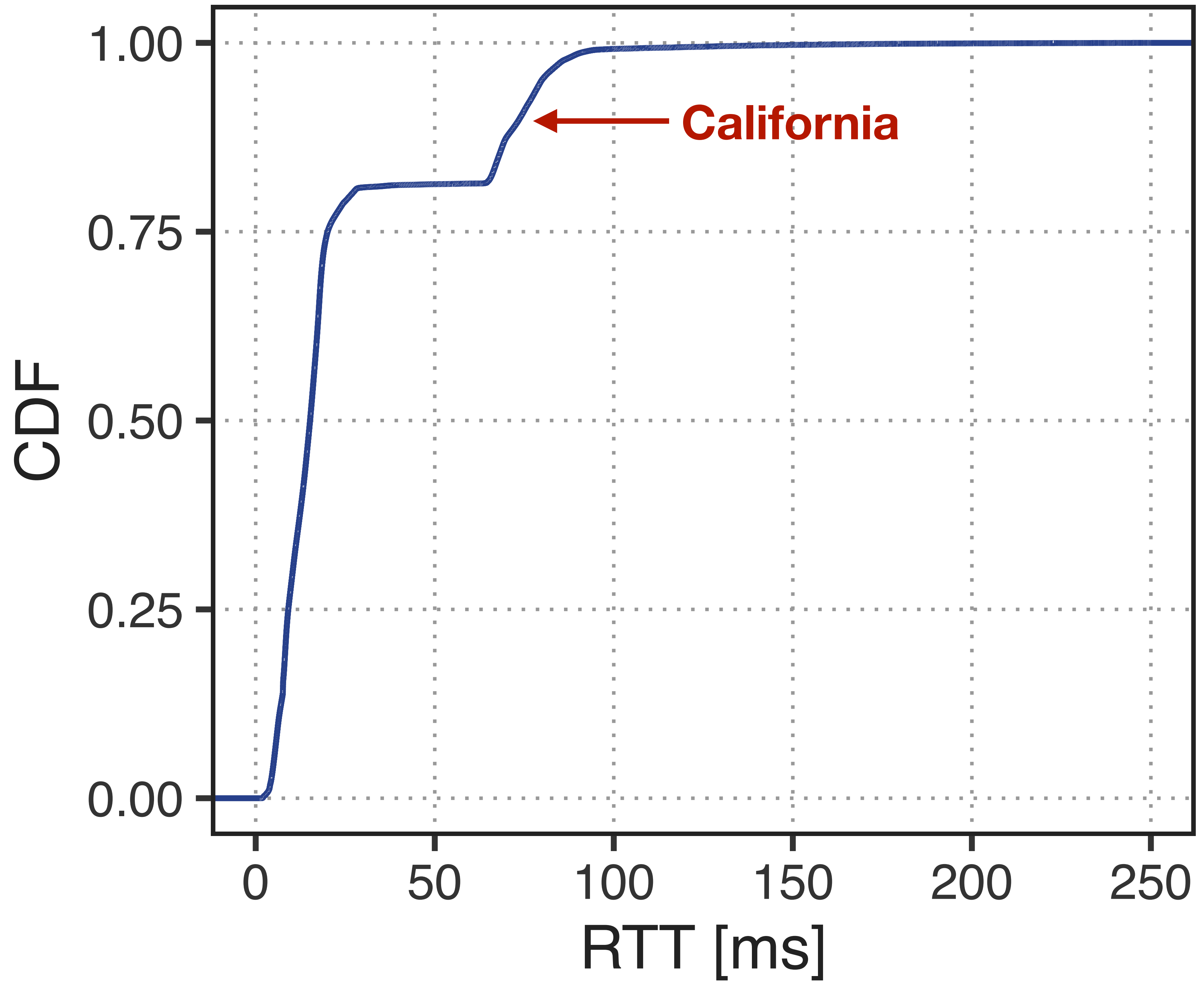


Round-Trip-Times to Zoom Servers in 12-hour Campus Trace.

Analyzing Zoom Campus Traffic

Latency to Zoom Server

- RTT to server by matching RTP sequence numbers
- 30.1 M latency samples to 521 distinct Zoom IP addresses
- Vast majority of meetings connected to NYC and California data centers

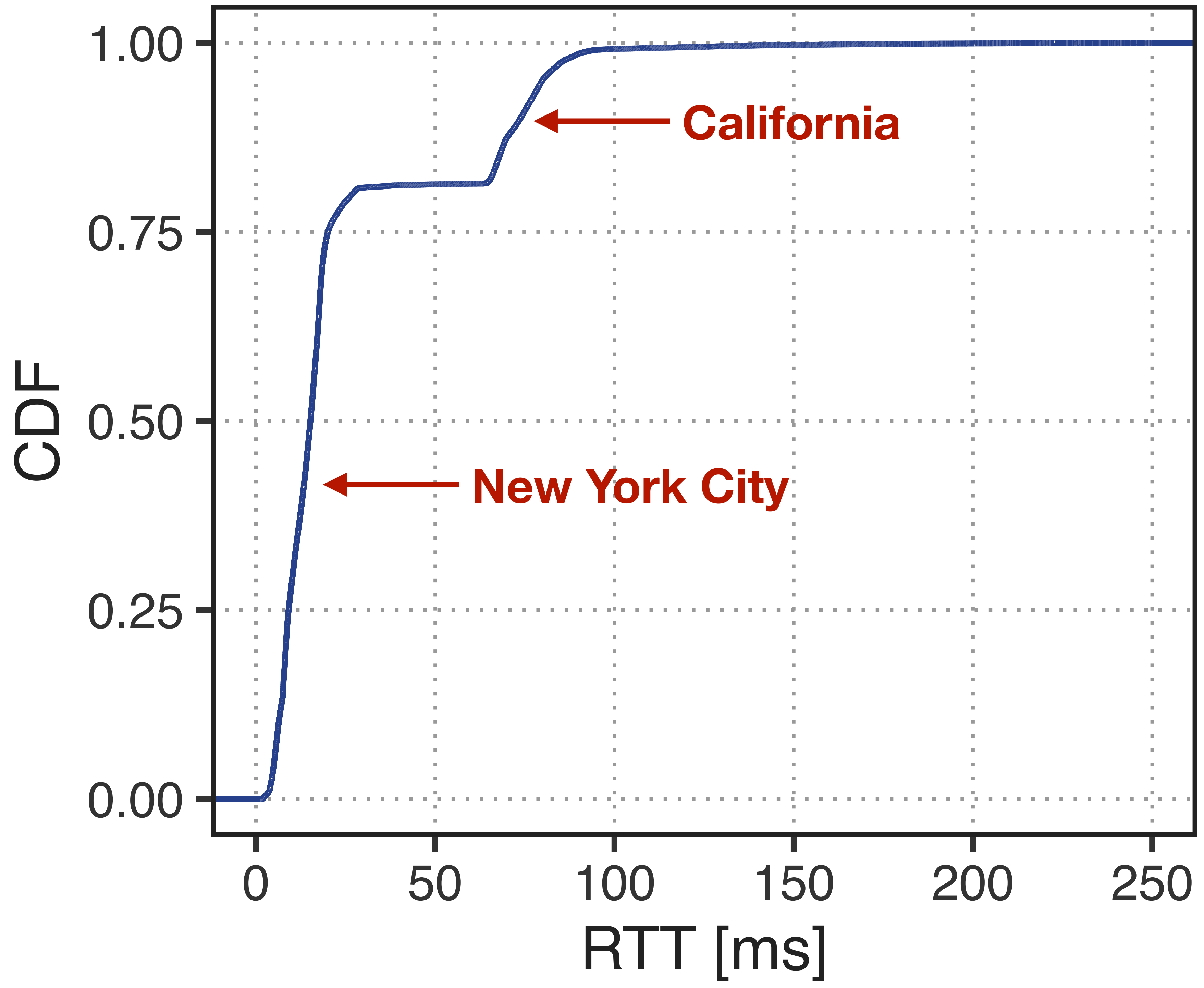


Round-Trip-Times to Zoom Servers
in 12-hour Campus Trace.

Analyzing Zoom Campus Traffic

Latency to Zoom Server

- RTT to server by matching RTP sequence numbers
- 30.1 M latency samples to 521 distinct Zoom IP addresses
- Vast majority of meetings connected to NYC and California data centers

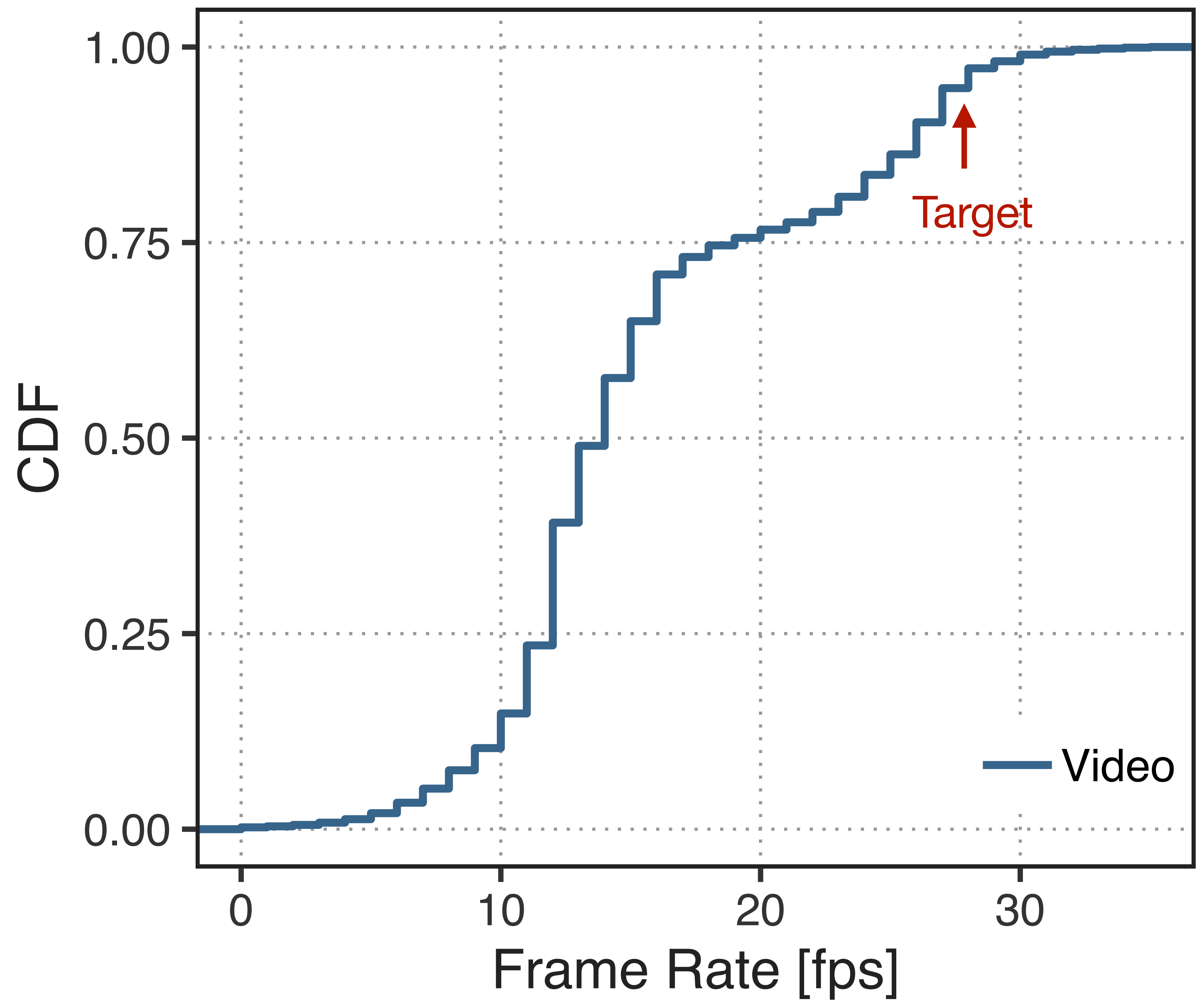


Round-Trip-Times to Zoom Servers in 12-hour Campus Trace.

Analyzing Zoom Campus Traffic

Frame Rate

- Zoom aims at encoding video at 28 fps
- ~75% of samples show frame rate of less than 20 fps → network problem?

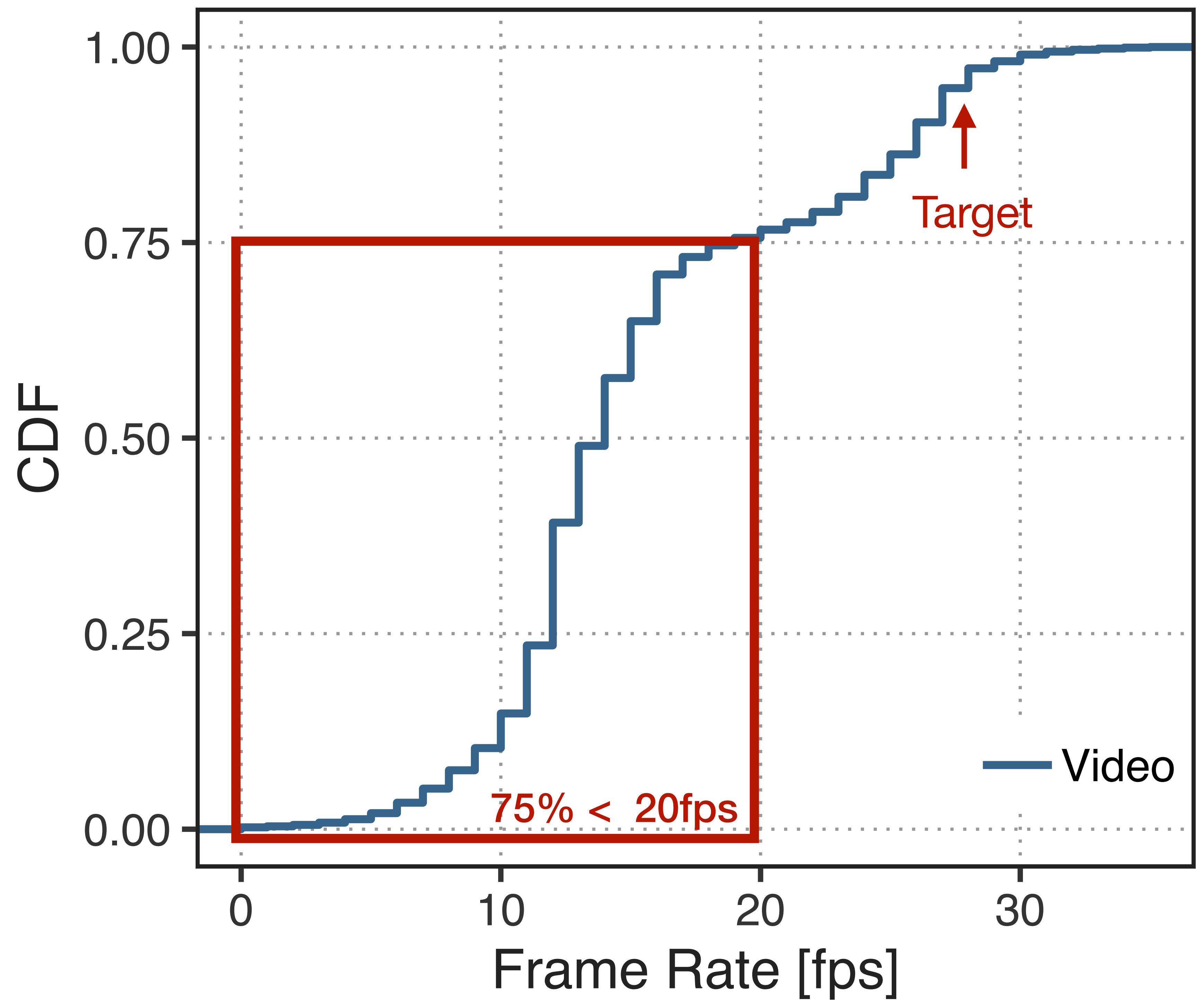


Frame Rates Observed in 12-hour Campus Trace.

Analyzing Zoom Campus Traffic

Frame Rate

- Zoom aims at encoding video at 28 fps
- ~75% of samples show frame rate of less than 20 fps → network problem?

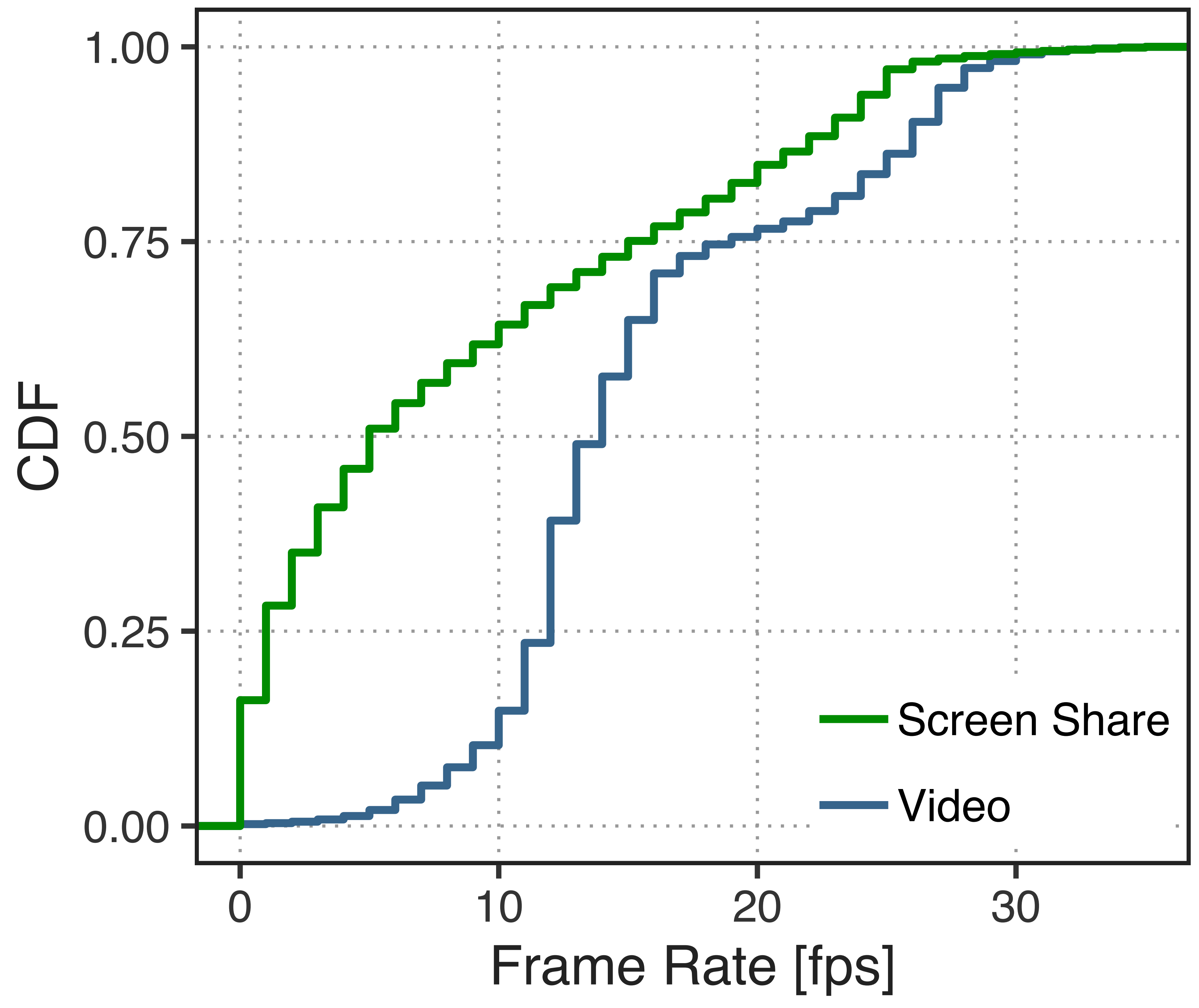


Frame Rates Observed in 12-hour Campus Trace.

Analyzing Zoom Campus Traffic

Frame Rate

- Zoom aims at encoding video at 28 fps
- ~75% of samples show frame rate of less than 20 fps → network problem?
- Content-adaptive encoding of screen sharing content



Frame Rates Observed in 12-hour Campus Trace.

Conclusion

Enable future research and performance measurement on Zoom by

(1) Demystifying its network protocol and operation

(2) Showing how to extract useful performance- and quality-related metrics from passive measurements

Conclusion

Use Cases & Future Directions

(1) Better QoE metrics, *e.g.*, estimation of stall likelihood

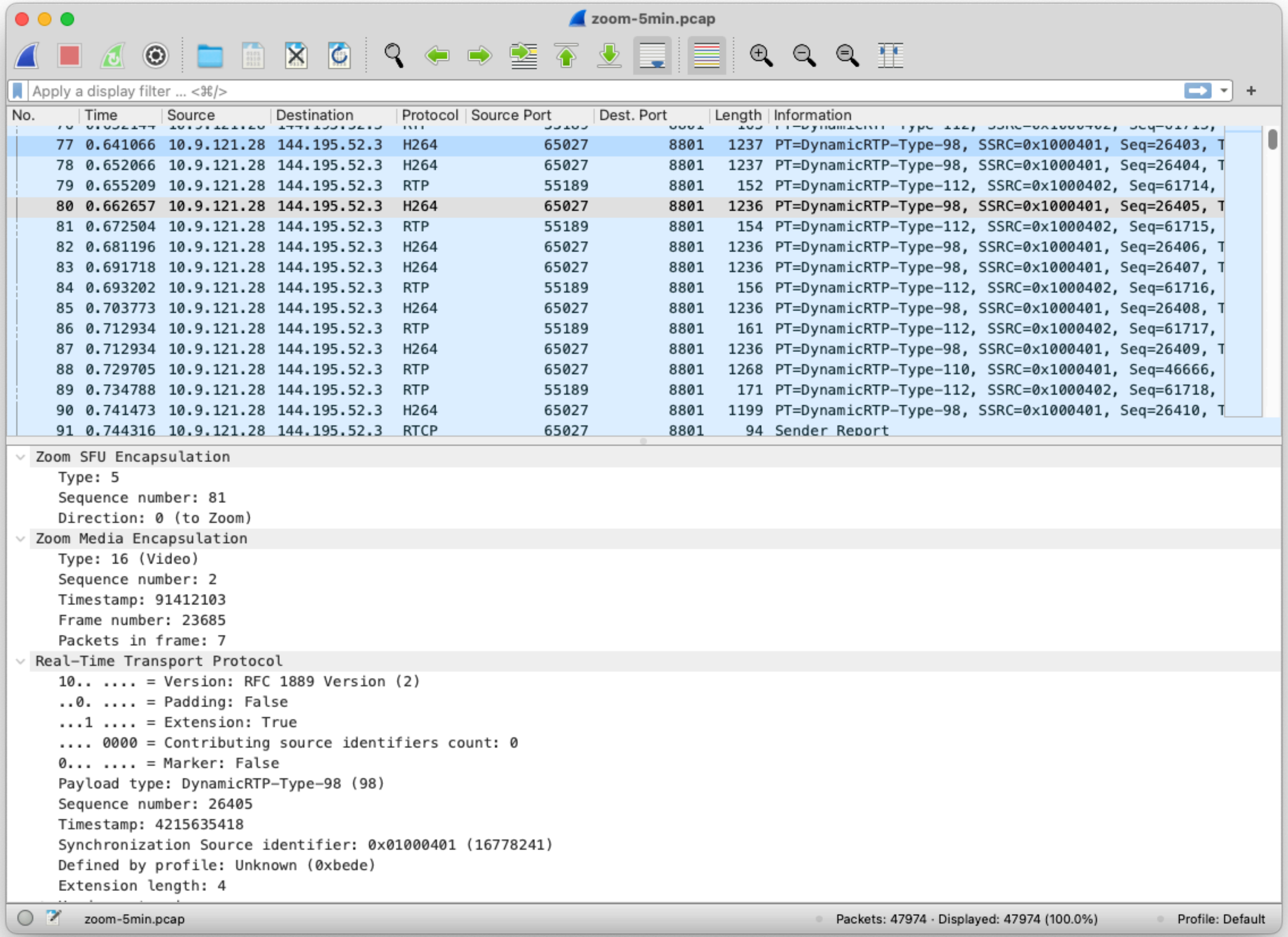
(2) Real-time monitoring of video-conferencing performance in programmable switches

(3) Offloading SFU functionality to programmable switches

Artifacts

- Analysis Tools
- R Notebooks
- Wireshark Plugin
- P4 Capture Program

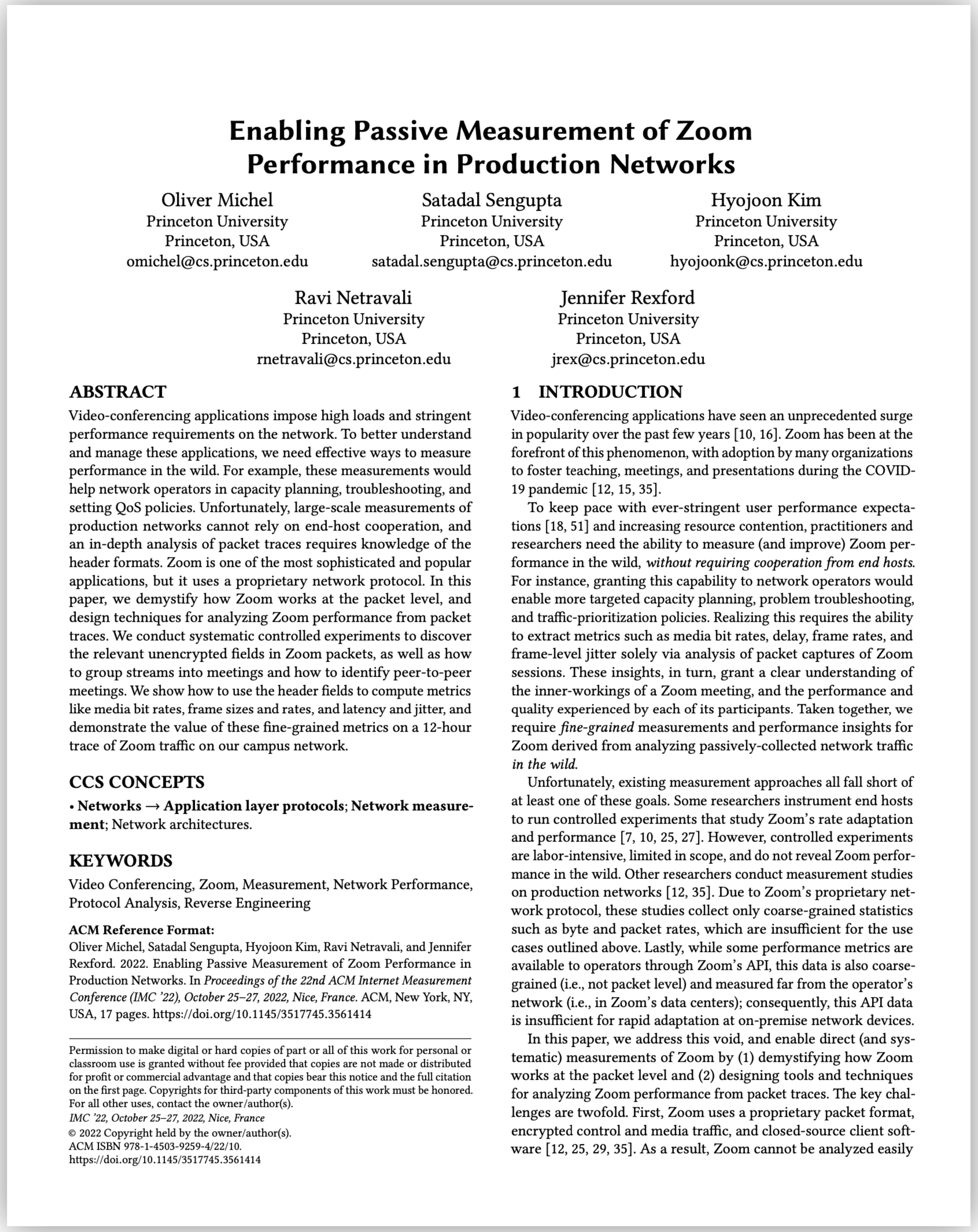
github.com/princeton-cabernet/zoom-analysis



Q&A

github.com/princeton-cabernet/zoom-analysis

omichel@cs.princeton.edu



Enabling Passive Measurement of Zoom Performance in Production Networks

Oliver Michel
Princeton University
Princeton, USA
omichel@cs.princeton.edu

Satadal Sengupta
Princeton University
Princeton, USA
satadal.sengupta@cs.princeton.edu

Hyojoon Kim
Princeton University
Princeton, USA
hyojoonk@cs.princeton.edu

Ravi Netravali
Princeton University
Princeton, USA
rnetravali@cs.princeton.edu

Jennifer Rexford
Princeton University
Princeton, USA
jrex@cs.princeton.edu

ABSTRACT

Video-conferencing applications impose high loads and stringent performance requirements on the network. To better understand and manage these applications, we need effective ways to measure performance in the wild. For example, these measurements would help network operators in capacity planning, troubleshooting, and setting QoS policies. Unfortunately, large-scale measurements of production networks cannot rely on end-host cooperation, and an in-depth analysis of packet traces requires knowledge of the header formats. Zoom is one of the most sophisticated and popular applications, but it uses a proprietary network protocol. In this paper, we demystify how Zoom works at the packet level, and design techniques for analyzing Zoom performance from packet traces. We conduct systematic controlled experiments to discover the relevant unencrypted fields in Zoom packets, as well as how to group streams into meetings and how to identify peer-to-peer meetings. We show how to use the header fields to compute metrics like media bit rates, frame sizes and rates, and latency and jitter, and demonstrate the value of these fine-grained metrics on a 12-hour trace of Zoom traffic on our campus network.

CCS CONCEPTS

• **Networks → Application layer protocols**; **Network measurement**; Network architectures.

KEYWORDS

Video Conferencing, Zoom, Measurement, Network Performance, Protocol Analysis, Reverse Engineering

ACM Reference Format:

Oliver Michel, Satadal Sengupta, Hyojoon Kim, Ravi Netravali, and Jennifer Rexford. 2022. Enabling Passive Measurement of Zoom Performance in Production Networks. In *Proceedings of the 22nd ACM Internet Measurement Conference (IMC '22), October 25–27, 2022, Nice, France.* ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3517745.3561414

IMC '22, October 25–27, 2022, Nice, France

ACM ISBN 978-1-4503-9259-4/22/10.
https://doi.org/10.1145/3517745.3561414

1 INTRODUCTION

Video-conferencing applications have seen an unprecedented surge in popularity over the past few years [10, 16]. Zoom has been at the forefront of this phenomenon, with adoption by many organizations to foster teaching, meetings, and presentations during the COVID-19 pandemic [12, 15, 35].

To keep pace with ever-stringent user performance expectations [18, 51] and increasing resource contention, practitioners and researchers need the ability to measure (and improve) Zoom performance in the wild, *without requiring cooperation from end hosts.* For instance, granting this capability to network operators would enable more targeted capacity planning, problem troubleshooting, and traffic-prioritization policies. Realizing this requires the ability to extract metrics such as media bit rates, delay, frame rates, and frame-level jitter solely via analysis of packet captures of Zoom sessions. These insights, in turn, grant a clear understanding of the inner-workings of a Zoom meeting, and the performance and quality experienced by each of its participants. Taken together, we require *fine-grained* measurements and performance insights for Zoom derived from analyzing passively-collected network traffic *in the wild.*

Unfortunately, existing measurement approaches all fall short of at least one of these goals. Some researchers instrument end hosts to run controlled experiments that study Zoom's rate adaptation and performance [7, 10, 25, 27]. However, controlled experiments are labor-intensive, limited in scope, and do not reveal Zoom performance in the wild. Other researchers conduct measurement studies on production networks [12, 35]. Due to Zoom's proprietary network protocol, these studies collect only coarse-grained statistics such as byte and packet rates, which are insufficient for the use cases outlined above. Lastly, while some performance metrics are available to operators through Zoom's API, this data is also coarse-grained (i.e., not packet level) and measured far from the operator's network (i.e., in Zoom's data centers); consequently, this API data is insufficient for rapid adaptation at on-premise network devices.

In this paper, we address this void, and enable direct (and systematic) measurements of Zoom by (1) demystifying how Zoom works at the packet level and (2) designing tools and techniques for analyzing Zoom performance from packet traces. The key challenges are twofold. First, Zoom uses a proprietary packet format, encrypted control and media traffic, and closed-source client software [12, 25, 29, 35]. As a result, Zoom cannot be analyzed easily

BACKUP SLIDES

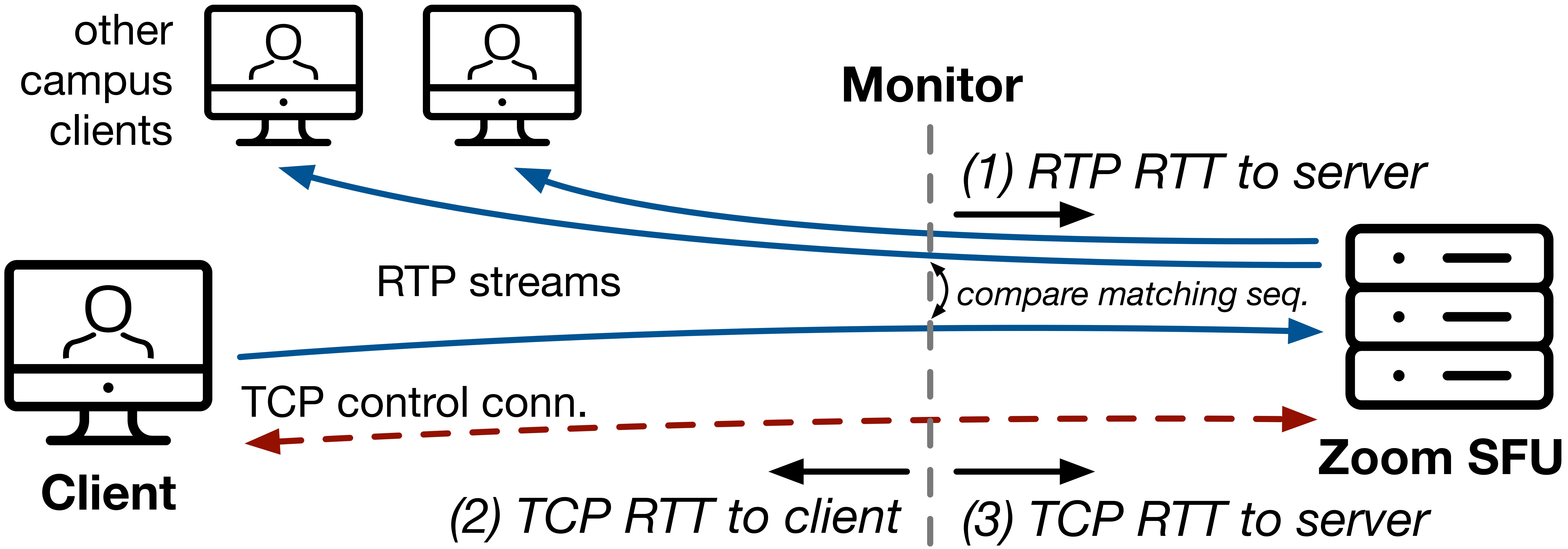
Backup Slides

Index

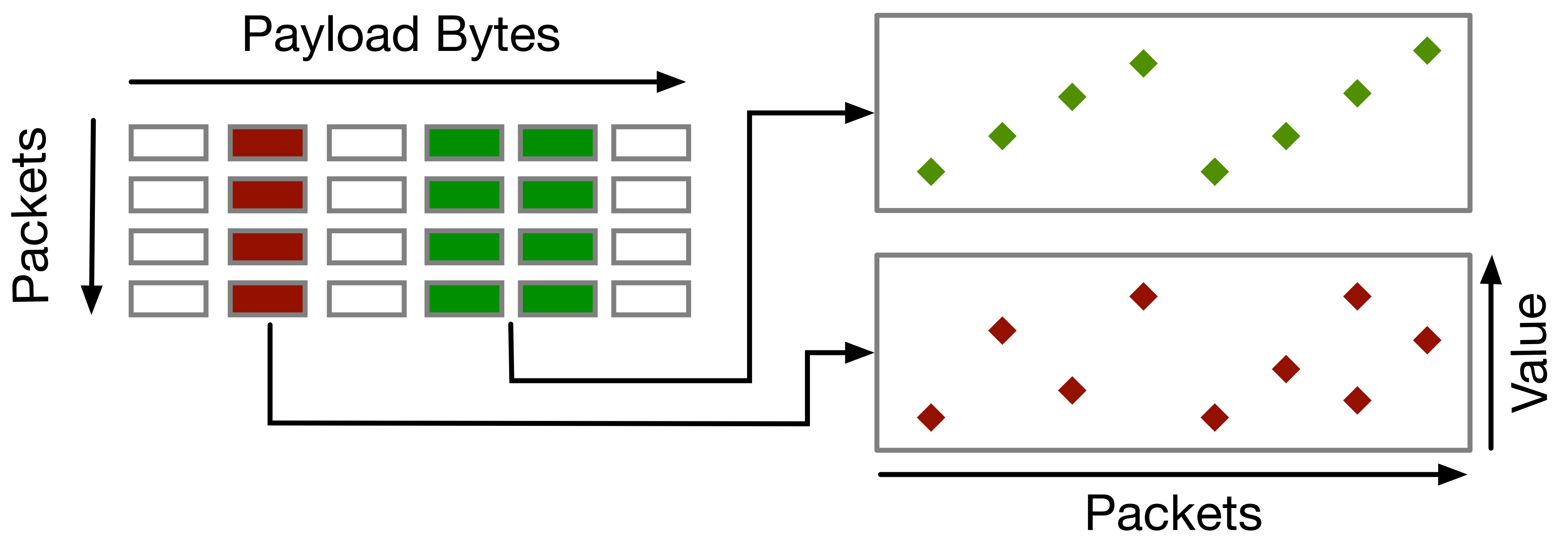
Possible Reasons for Proprietary Header Format

- More advanced/custom congestion control algorithm
- Many customizations that do require additional in-band meta data
- Attempt to hide internals

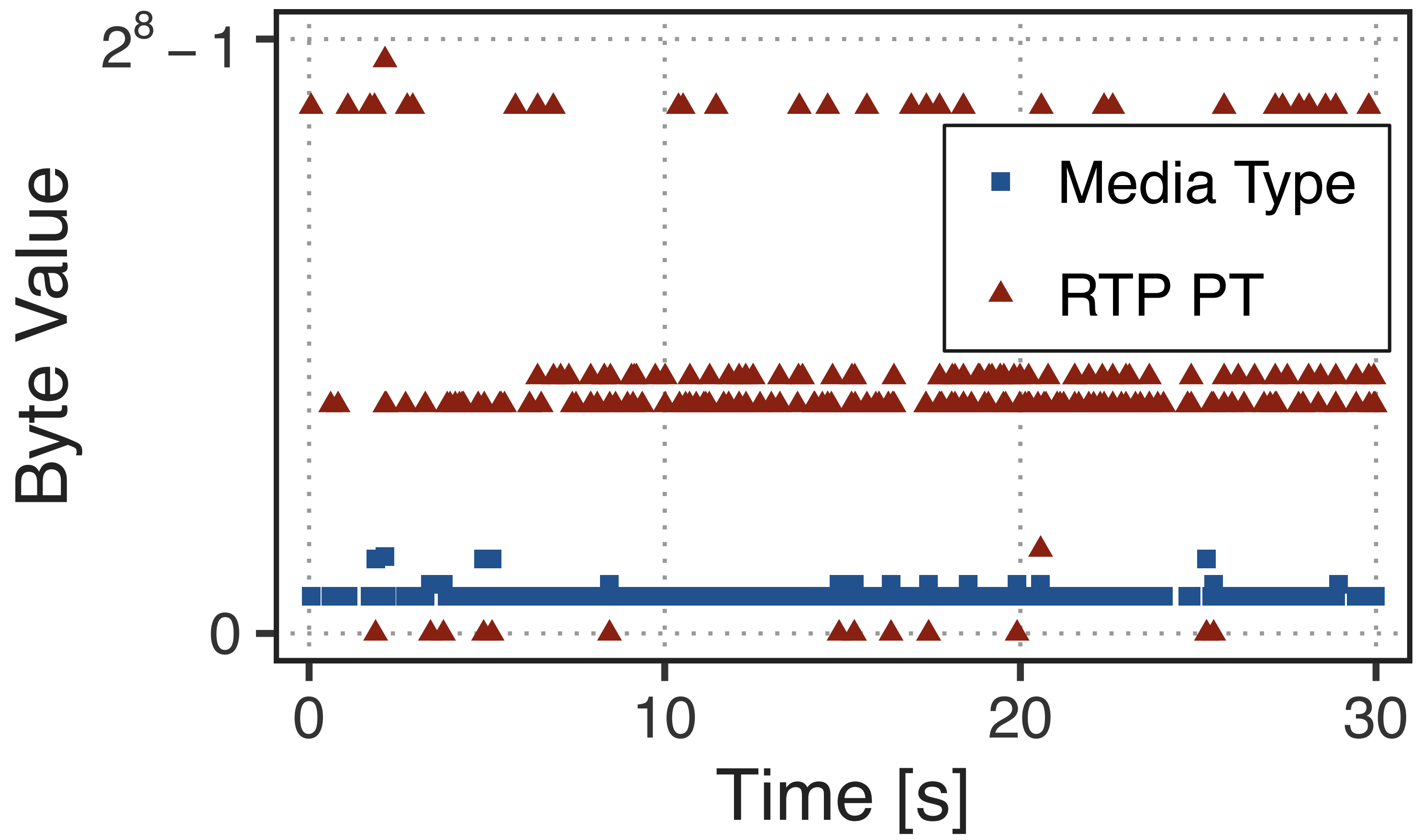
Latency Measurement Methods

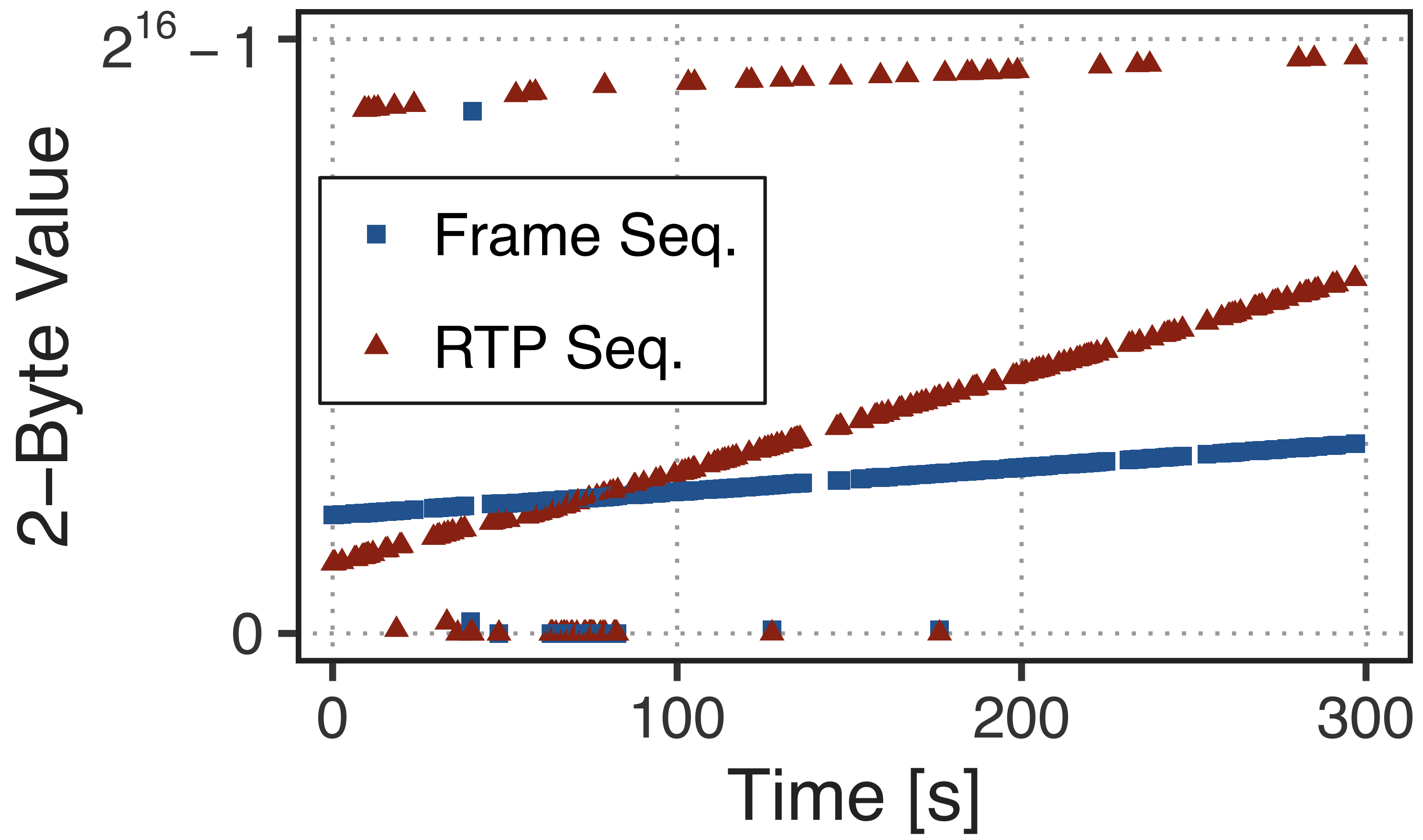


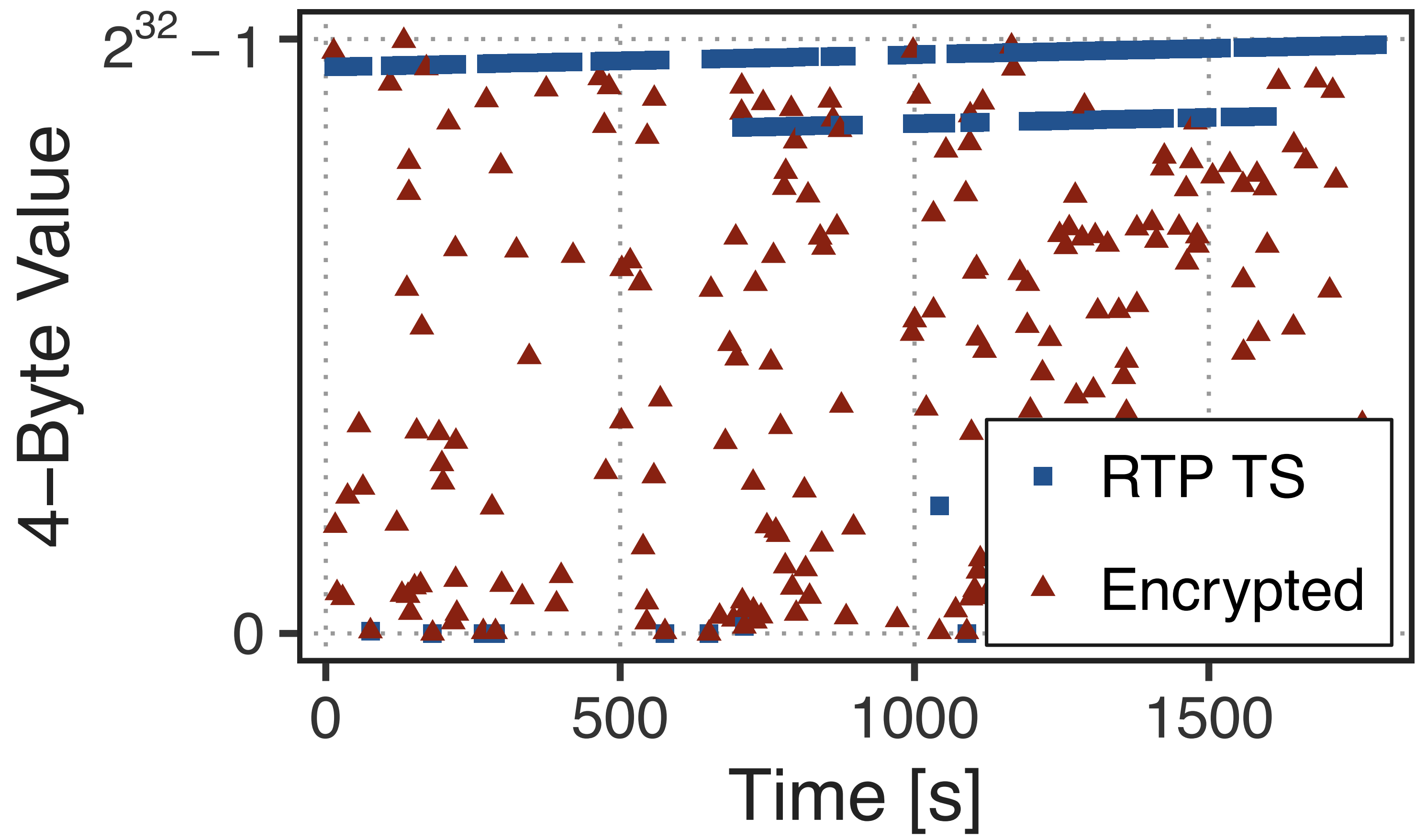
Entropy-based Header Analysis



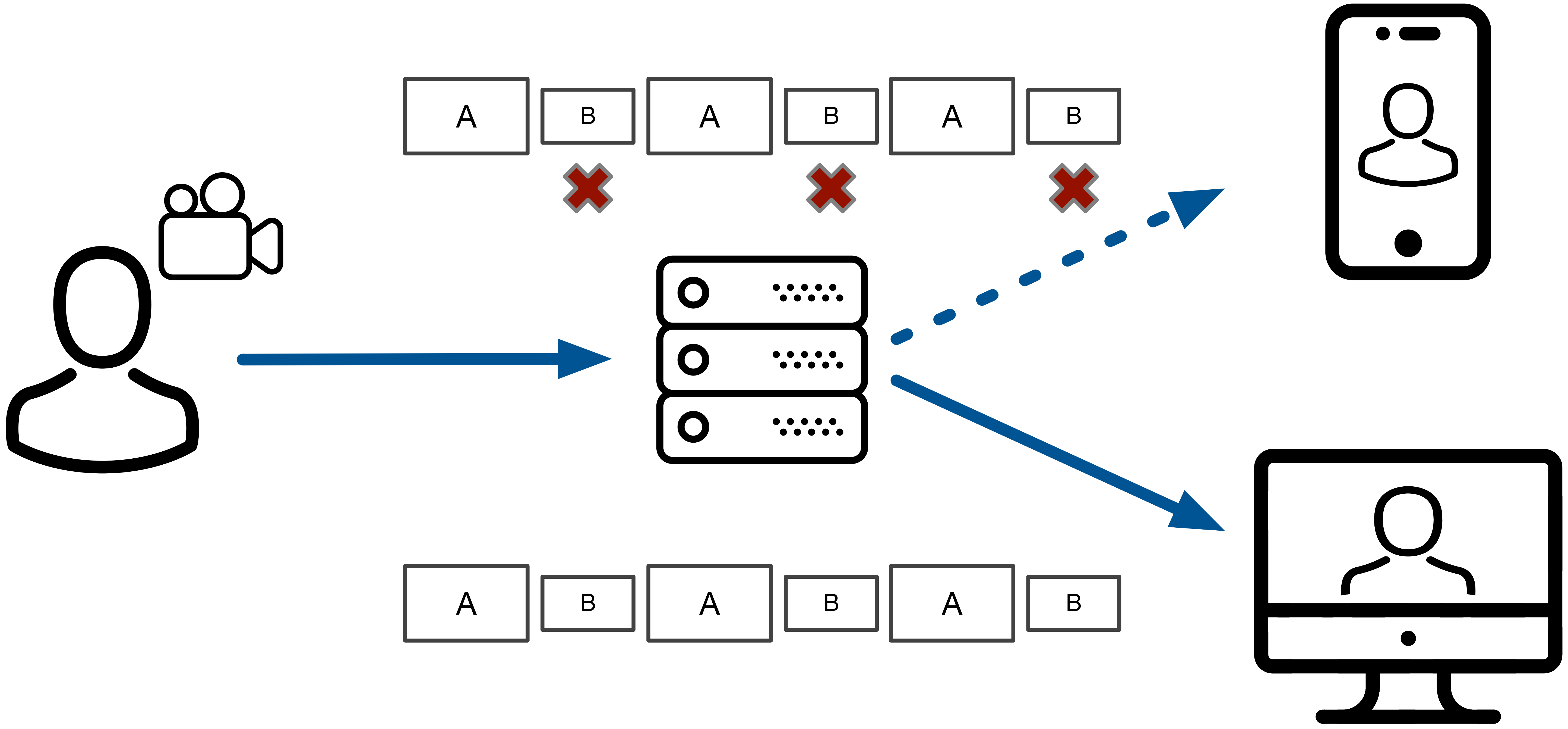
- Extract 1B, 2B, 4B values from all (1B-aligned) offsets after UDP header
- Plot values over time





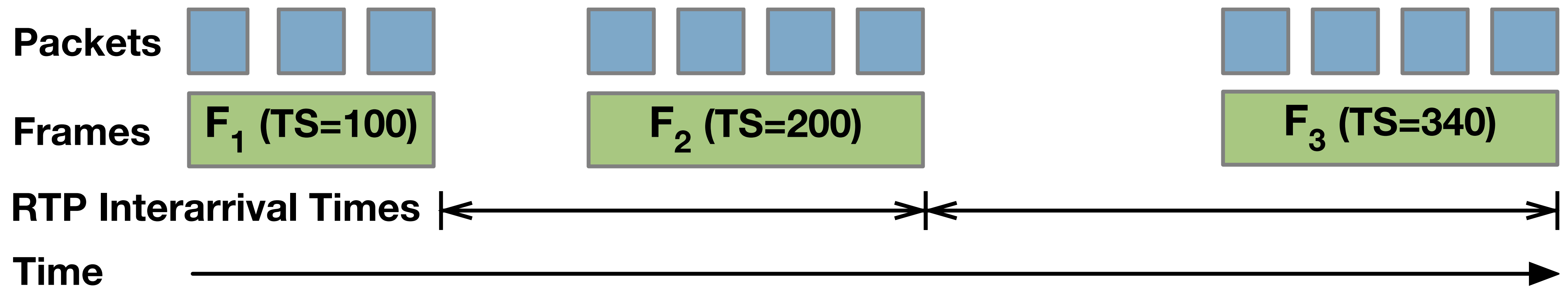


Need for Unencrypted Header Fields/SFU

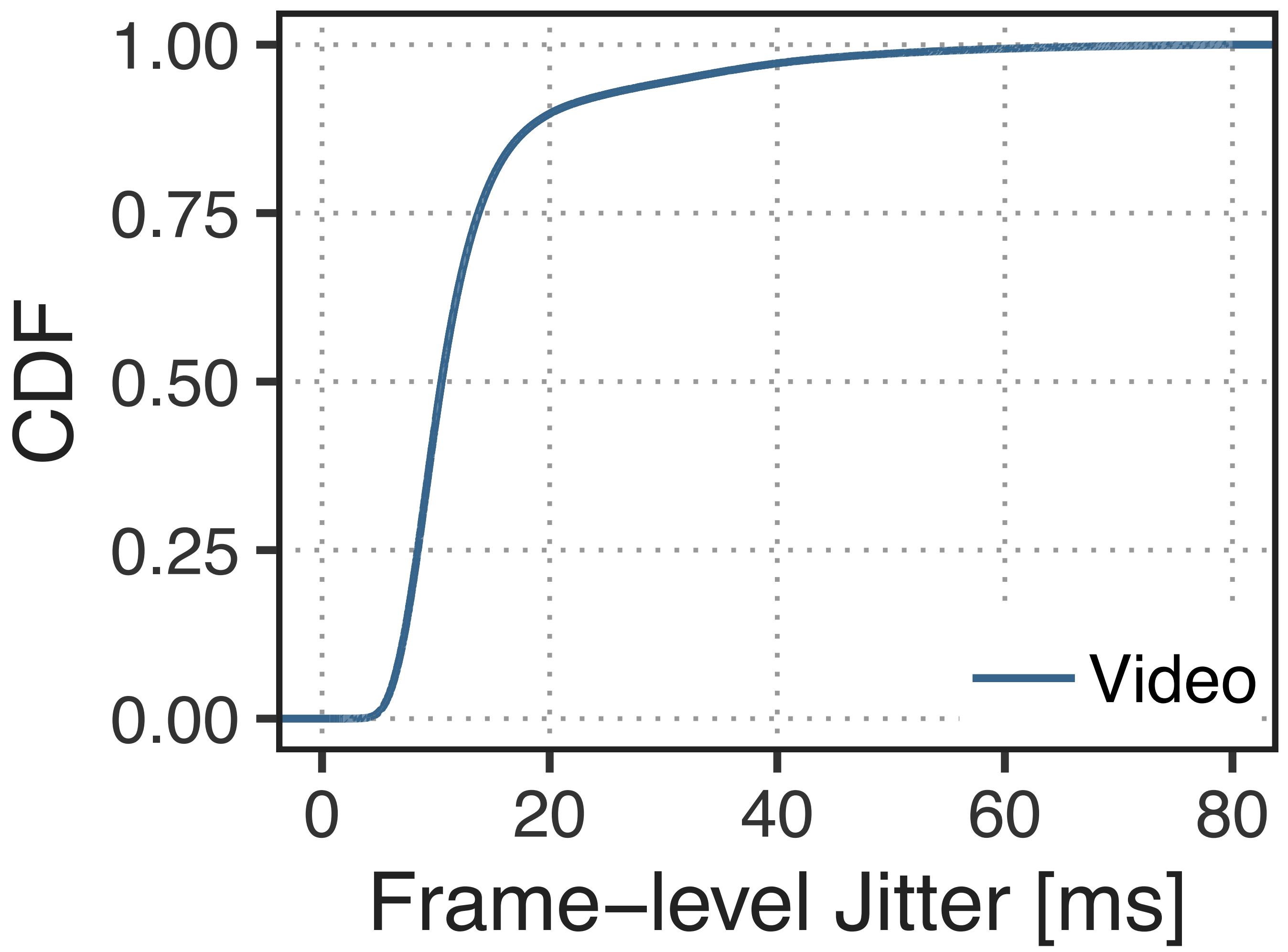


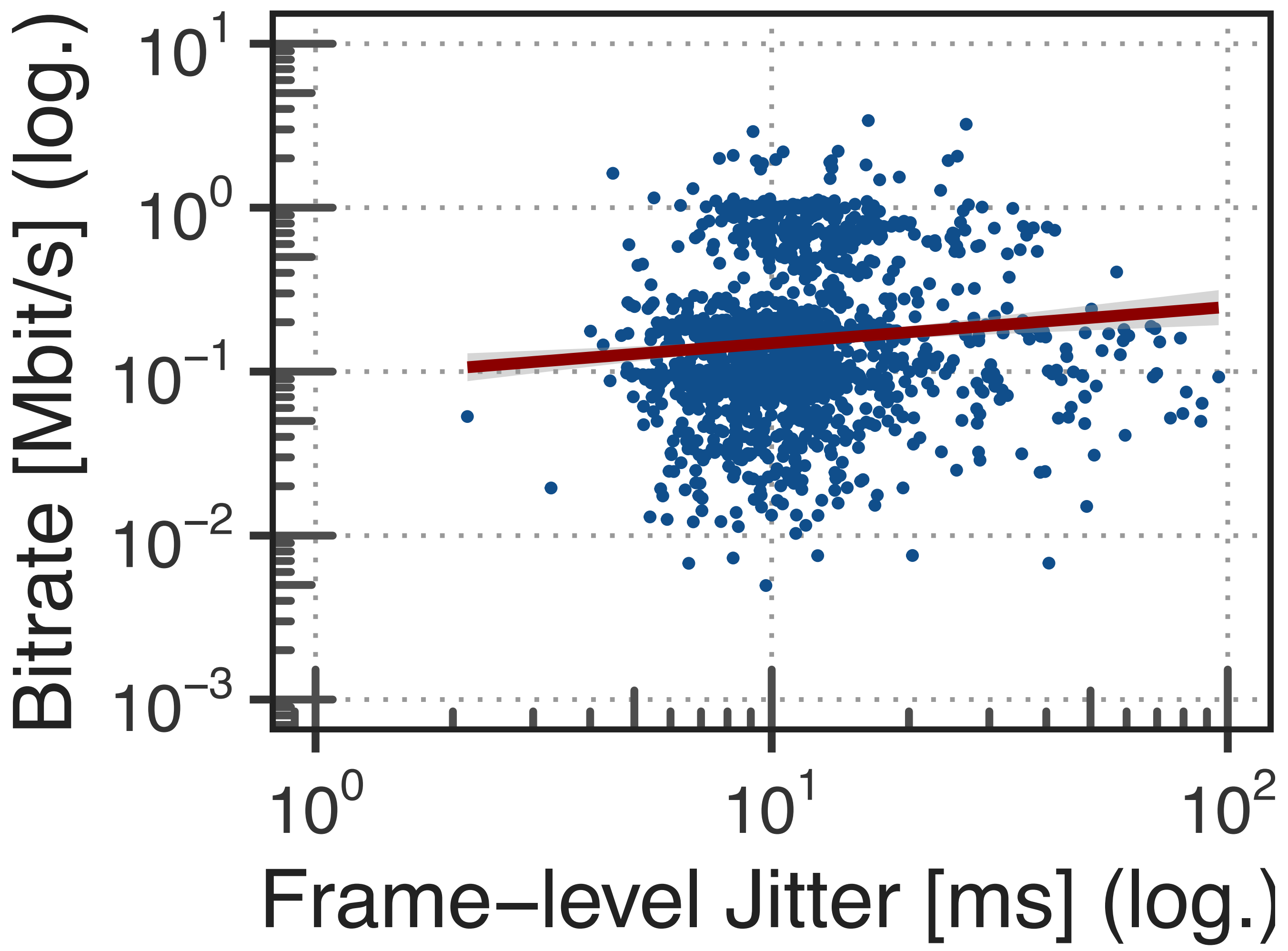
- Selective forwarding based on header meta data
- Encrypted header fields would result in:
  - Decrypt/encrypt for every packet / expensive, not performed by Zoom
  - Not compatible with E2E encryption

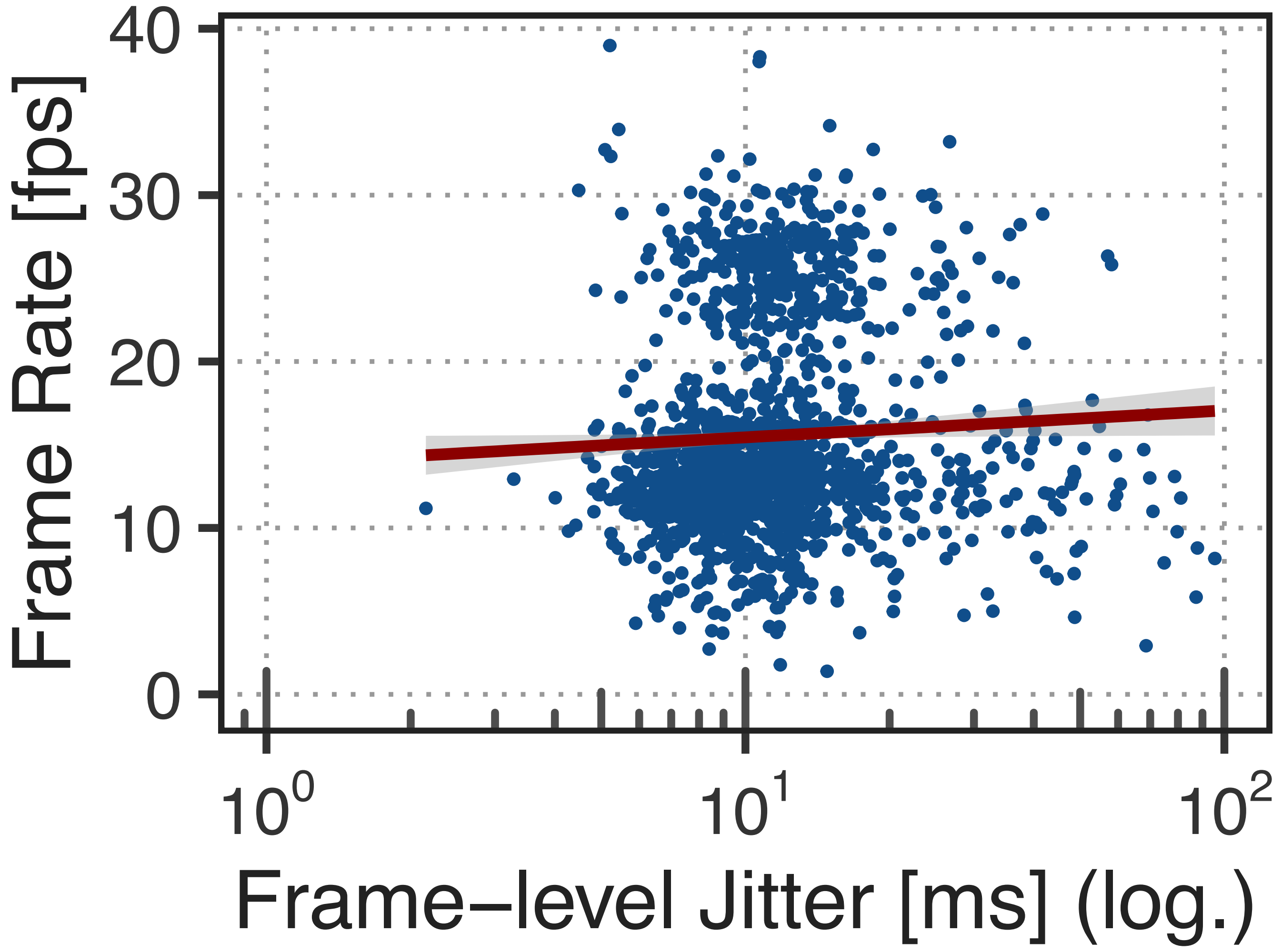
Frame-level Jitter Calculation



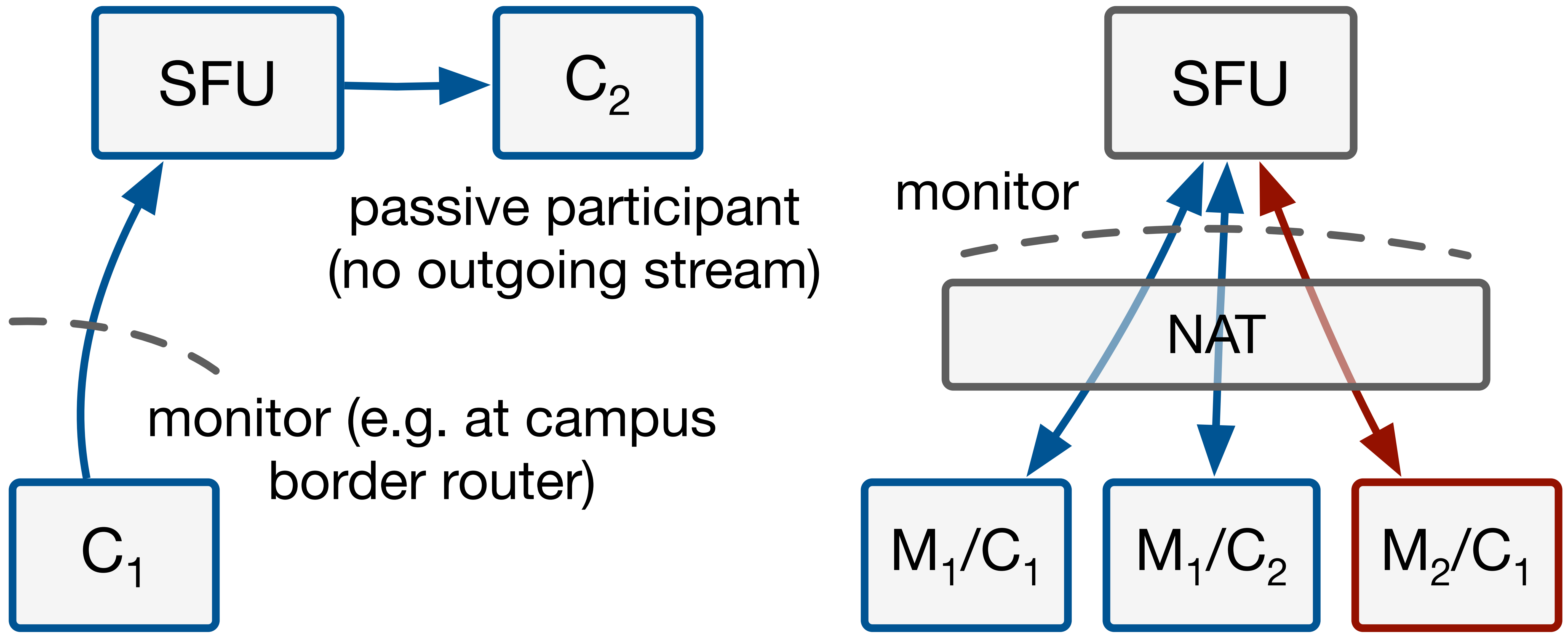
RFC 3550: Appendix A.8
Estimating the Interarrival Jitter



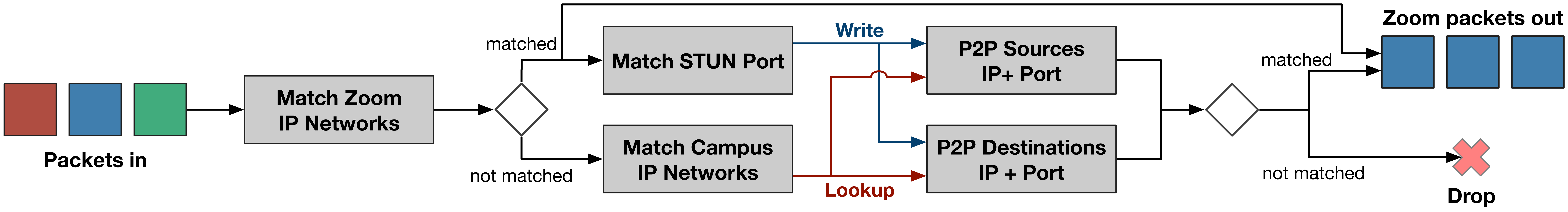




Limitations of Grouping Heuristic



P4 Program



Packets in
Match Zoom
IP Networks
matched
not matched
Match STUN Port
Match Campus
IP Networks
Write
Lookup
P2P Sources
IP+ Port
P2P Destinations
IP + Port
matched
not matched
Zoom packets out
Drop

Relation to ML-based Approaches

- Precise measurements / no estimations
- Performance indicators vs. Quality of Experience
    - Measurements can be used to create features
    - Labels from E2E measurements (e.g., SSIM) or MOS
- Create large, feature-rich data sets from production networks

Use of RTP in Video-Conferencing Applications

TABLE II: Comparison of the RTC applications under test. Under *Redundant data*, “F” stands for FEC and “S” for Simulcast. Under *DNS domains*, “B” stands for easy to block, “C” for company-specific and “S” for social networks. Under *Other*, “N” means it uses less than four server-side ports and “T” means that PTs are used in a static fashion.

Application	Protocols				Operation			Identification		
	RTP	STUN/TURN	DTLS	Other	P2P	Redundant Data	Other	Own AS	DNS Domains	Other
Skype	✓	✓		✓	✓	F,S		✓	B	N,T
Google Meet	✓	✓	✓			S	✓	✓	C	N,T
Jitsi Meet	✓	✓	✓		✓				B	
WhatsApp	✓	✓			✓	F		✓	B	N,T
Telegram		✓		✓	✓			✓	B	
Facebook Messenger	✓	✓	✓		✓			✓	S	T
Instagram Messenger	✓	✓						✓	S	N,T
Facetime	✓	✓			✓		✓	✓	C	N,T
HouseParty	✓	✓	✓						B	T
Microsoft Teams	✓	✓		✓	✓	F,S		✓	B	N,T
Webex Teams	✓	✓				F,S	✓	✓	B	N
Zoom	✓			✓	✓	F			B	N,T
GoTo Meeting				✓					B	N

[A. Nisticò, D. Markudova, M. Trevisan, M. Meo and G. Carofiglio, "A comparative study of RTC applications”, 2020 IEEE International Symposium on Multimedia (ISM), 2020, pp. 1-8, doi: 10.1109/ISM.2020.00007.]