Loss Detection for ICN: Probe-based approach

◎**Kazuaki Ueda**, Takahiko Kato, Chikara Sasaki, Atsushi Tagami

KDDI Research, Inc.

IETF116 ICNRG

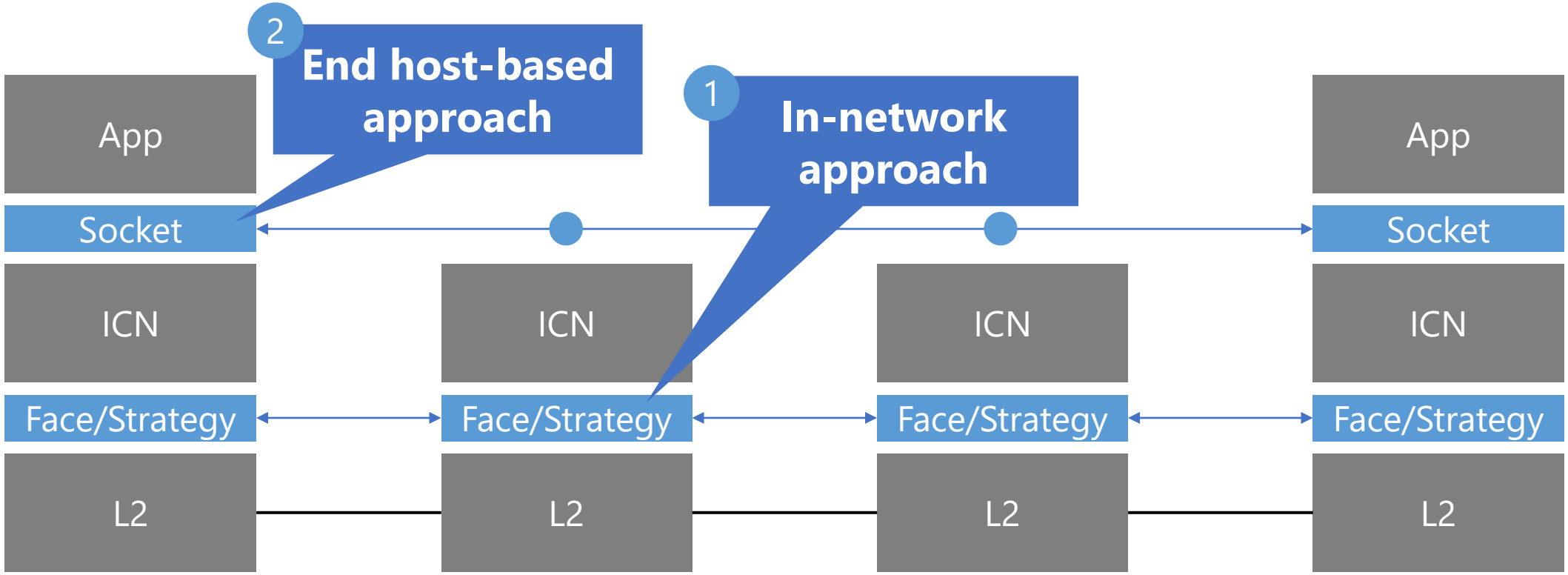
KDDI Research

Ueda, Kazuaki, et al. "Revisiting Loss Detection in NDN: Detecting Spurious Timeout using Probe Interest." *GLOBECOM 2022-2022 IEEE Global Communications Conference*. IEEE, 2022.

Agenda

- Loss detection in ICN
  - （Congestion Control is out of scope）
- Problem: Spurious timeouts in the end host-based approach
- Proposal: NDN-PTO
- Evaluation & Conclusion

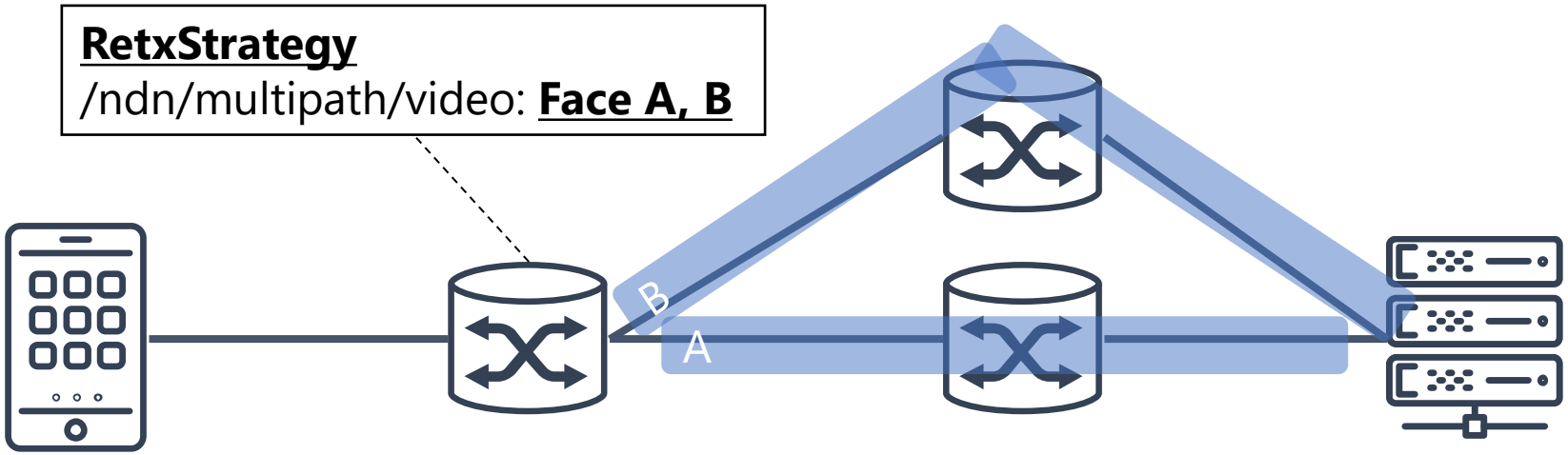
Loss Detection and Recovery in ICN



Anders Eriksson and Börje Ohlman, “Framework for Loss Detection, Retransmission, and Congestion Control in ICN”, ICNRG, 2017

In-Network Loss Detection and Recovery in ICN

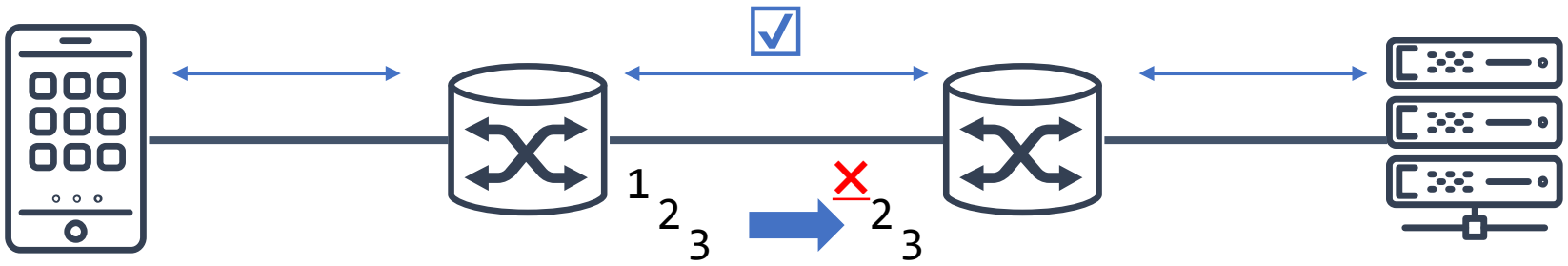
- **Forwarding Strategy**
  - Detect packet losses for the path
  - 👍: Rapid reaction for the path failure
  - 🤔: Complex interaction between App and strategy's behavior[1]



[1] Abraham, Hila Ben, and Patrick Crowley. "Controlling strategy retransmissions in named data networking." 2017 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS). IEEE, 2017.

In-Network Loss Detection and Recovery in ICN

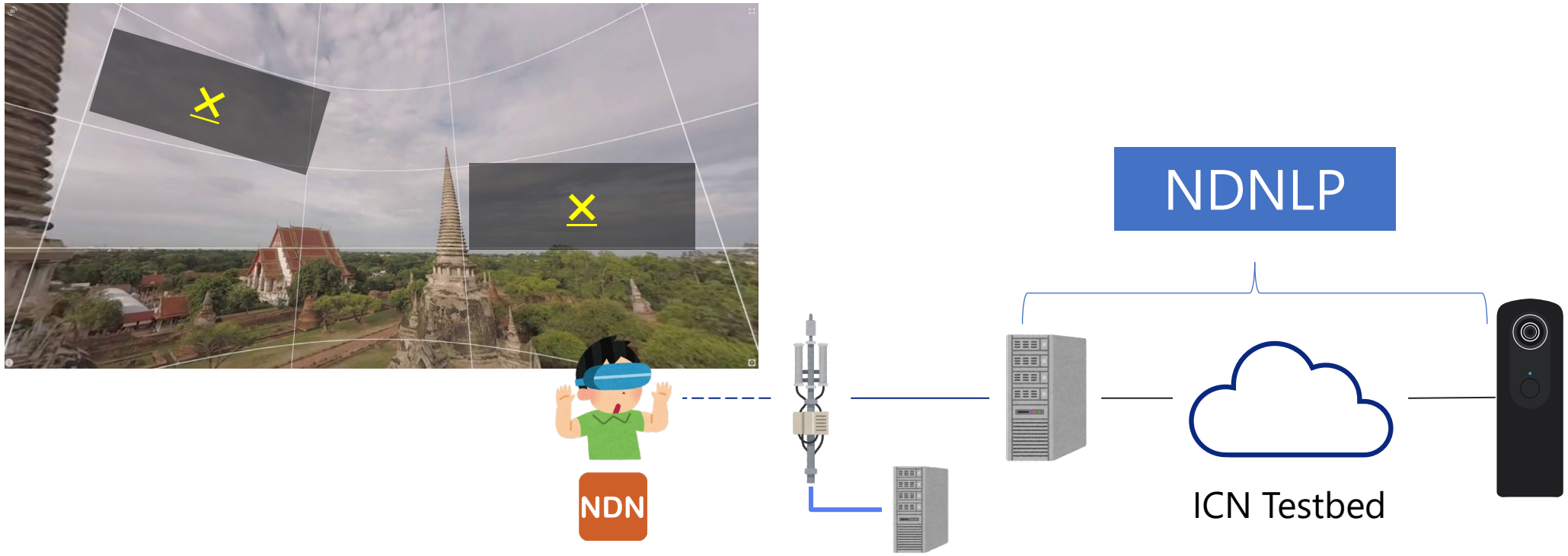
- **Face/Link Layer Protocol**
  - Detect packet losses **between two ICN routers**
  - WLDR (Wireless Loss Detection and Recovery) in hICN[2]
  - NDNLP Reliability Mode in NDN
  - 👍: Simple mechanism and Fast recovery



[2] Carofiglio, Giovanna, et al. "Leveraging ICN in-network control for loss detection and recovery in wireless mobile networks." Proceedings of the 3rd ACM Conference on Information-Centric Networking. 2016.

KDDI Research

Failures in the past demo



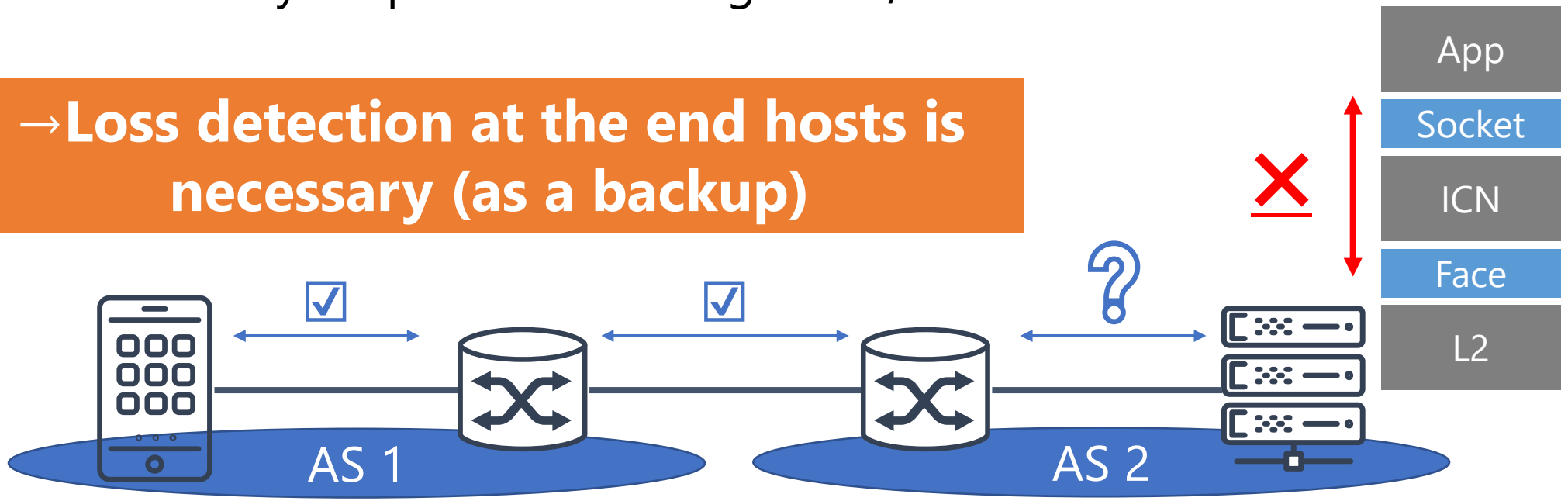
Tagami, Atsushi, et al. "Tile-based panoramic live video streaming on ICN."
2019 IEEE International Conference on Communications Workshops (ICC Workshops). IEEE, 2019.

KDDI Research

In-Network Loss Detection and Recovery in ICN

- This feature can provide loss-less communication in ICN, but…
  - Must be configured on **all hops** to rely on this feature
  - It cannot always cope with the congestion/overload

→Loss detection at the end hosts is necessary (as a backup)

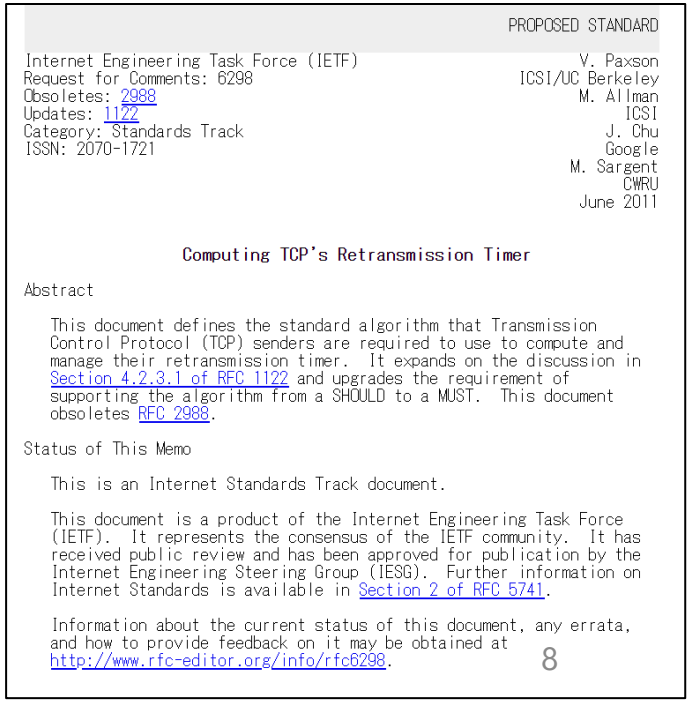


End host-based Loss detection and recovery in NDN

Current solution: **Retransmission timeout**

- Duplicate ACKs: cannot be applied to NDN (No ACK packets)
- Timer value (RTO) is calculated from **recent RTTs**
  - $RTO = \max(SRTT + 4 \times RTTVAR, \text{minRTO})$

"ndncatchunks" uses PCON[3] which uses timer-based loss detection



PROPOSED STANDARD

Internet Engineering Task Force (IETF)
Request for Comments: 6298
Obsoletes: 2988
Updates: 1122
Category: Standards Track
ISSN: 2070-1721

V. Paxson
ICSI/UC Berkeley
M. Allman
ICSI
J. Chu
Google
M. Sargent
CWRU
June 2011

Computing TCP's Retransmission Timer

Abstract

This document defines the standard algorithm that Transmission Control Protocol (TCP) senders are required to use to compute and manage their retransmission timer. It expands on the discussion in Section 4.2.3.1 of RFC 1122 and upgrades the requirement of supporting the algorithm from a SHOULD to a MUST. This document obsoletes RFC 2988.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6298.

[3] Schneider, Klaus, et al. "A practical congestion control scheme for named data networking." *Proceedings of the 3rd ACM Conference on Information-Centric Networking*. 2016.

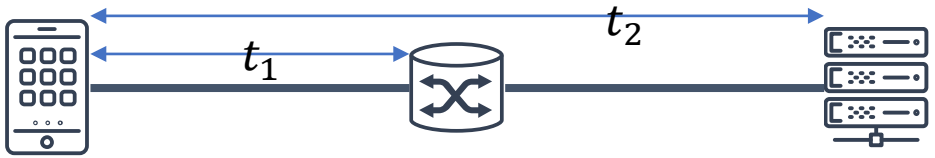
Problem: Spurious timeout

Spurious timeout (STO) (E.g., RFC5682)

- Known issues in TCP: **false loss detection** in the timer-based algorithm
- When an RTT of packet exceeds RTO, the packet is considered as lost
  - Delay: retransmission at wireless link, queuing delay, etc.
- →Decreased throughput

Timer-based approach in NDN introduces STOs more frequently than current Internet

- RTT of packet is not **constant** in the multi-path/producer model

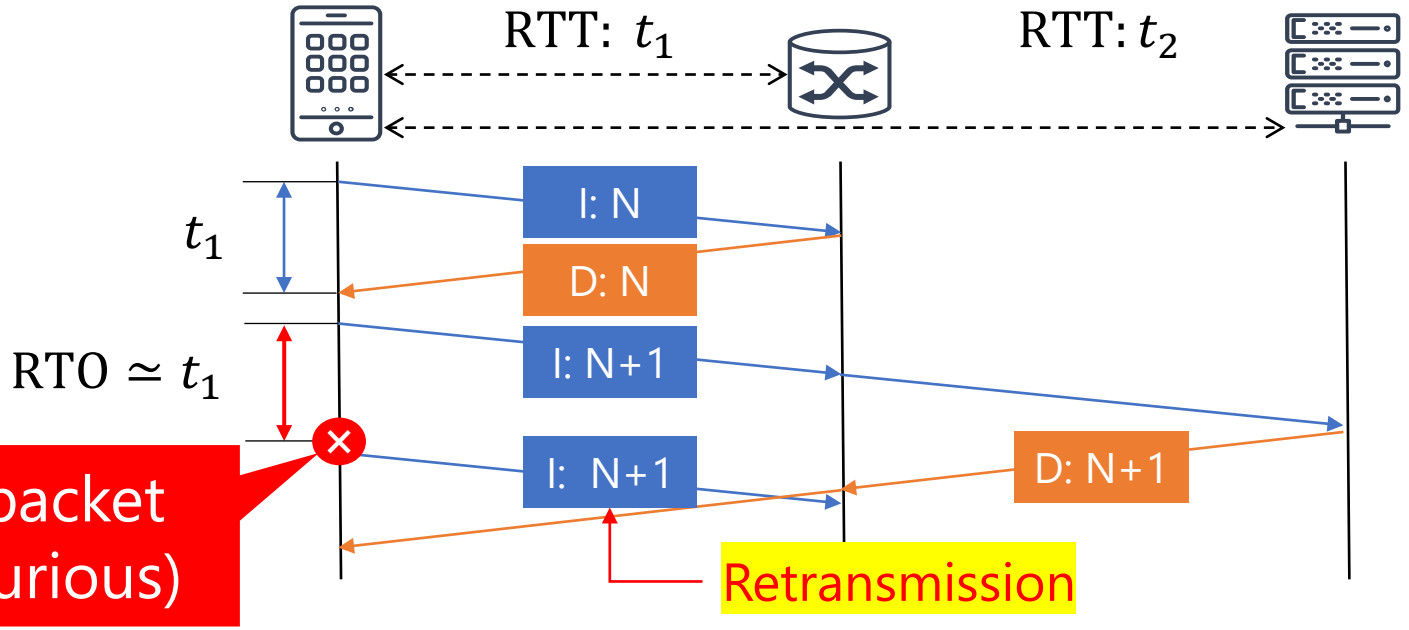


KDDI Research

STO from in-network cache

Example scenario: in-network cache causes STO

- First N Data are served from closer cache (RTT: t_1)
- The rest of Data are served from producer (RTT: t_2)
- RTT of $N+1$ Data (t_2) exceeds RTO (t_1), then consumer detects **packet loss**

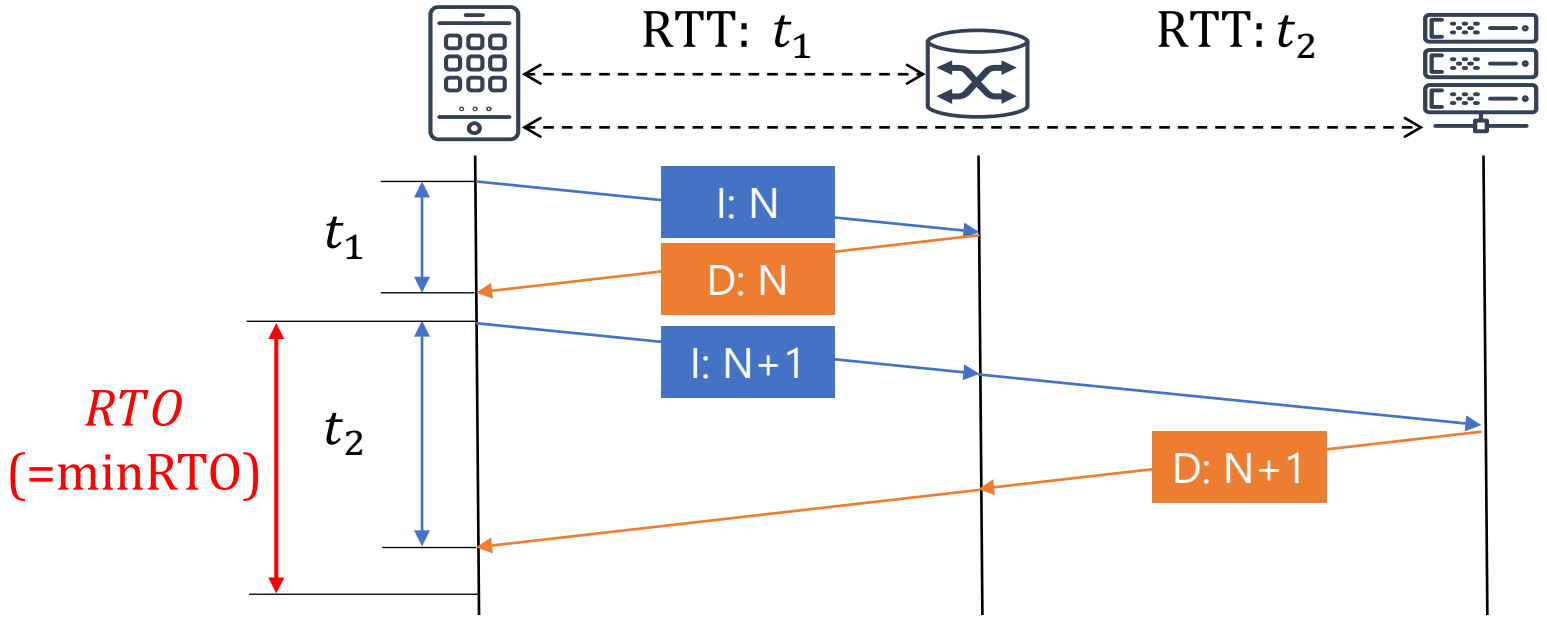


KDDI Research

Existing countermeasures for STO

$$RTO = \mathbf{max}(SRTT + 4 \times RTTVAR, \mathbf{minRTO})$$

- PCON: Set **large minRTO** parameter to increase RTO



Decrease spurious timeout at the expense of **loss response delay**
Optimal minRTO is topology dependent and unknown a priori

How to deal with STO in NDN?

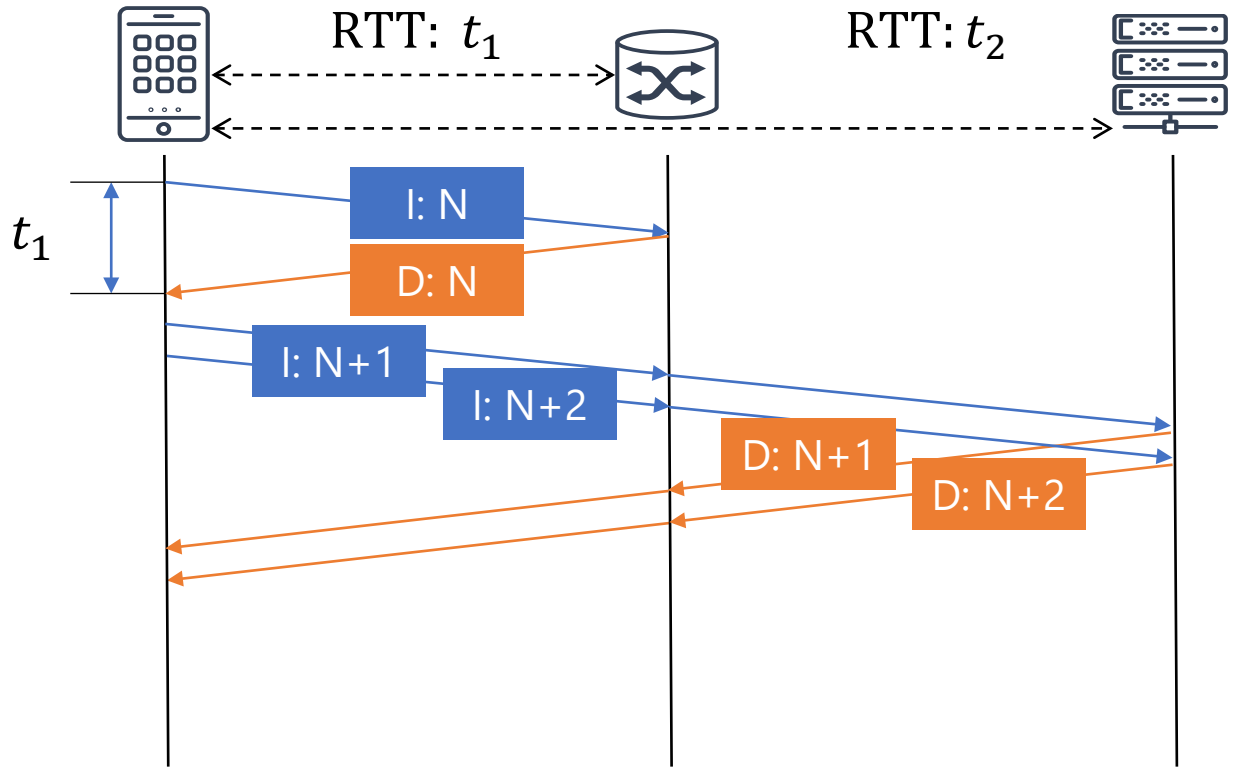
Problem: Traditional ReTx Timer alone does not match the NDN
(RTT can change significantly due to existence of multiple source nodes)

Idea: Obtain and update the RTT of new path
to determine “actual” packet loss

Solution: NDN-PTO (Probe TimeOut)
Use a “Probe Interest” to get information of current path info.

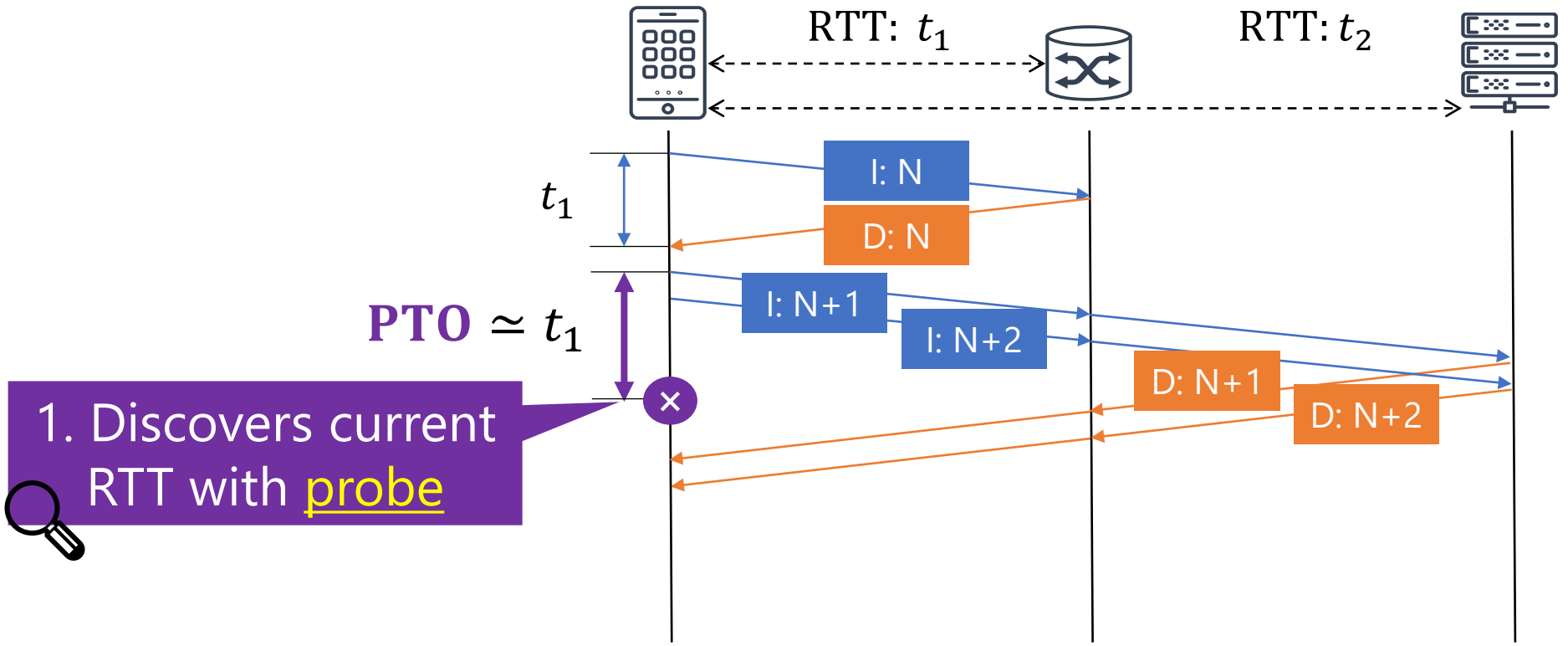
KDDI Research

Proposal: NDN-PTO

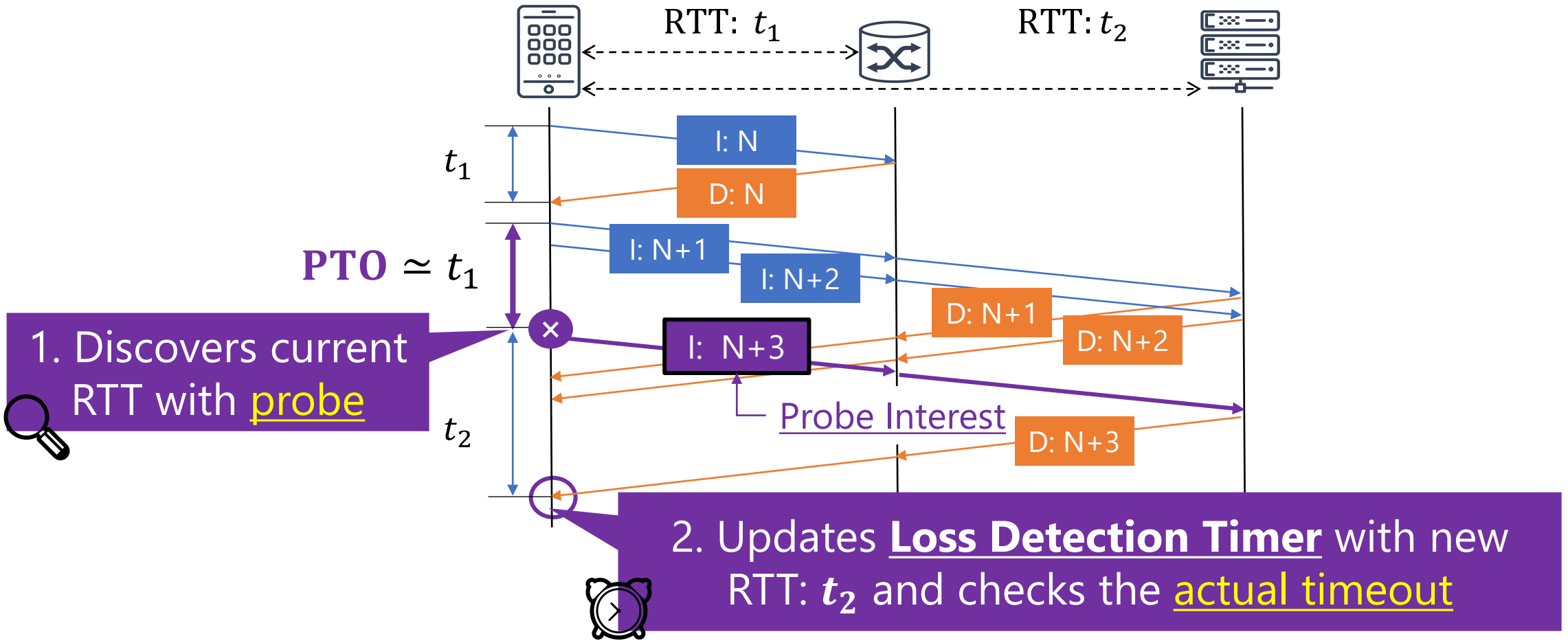


KDDI Research

Proposal: NDN-PTO



Proposal: NDN-PTO

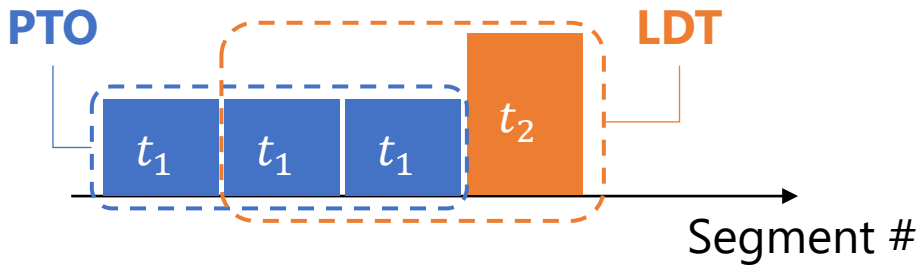


Detailed design of NDN-PTO

Core components: two timers and probe Interest

Probe Timeout (PTO) and Loss Detection Timer (LDT)

- PTO: **Short timer** calculated from recent RTTs and triggers probe
- LDT: **Long timer** calculated from the highest RTT and determines actual packet losses
  - Updated from probe response OR inflight Data



※ $PTO = 2^{c_p} \times \max(4 \times RTTVAR, \text{kGranularity}) + \text{SRTT}$

$LDT = \text{kTimeThreshold} \times highestRTT$

(Following the design of QUIC loss detection in RFC9002)

Detailed design of NDN-PTO

🔍 Probe Interest

- A single Interest which requests a next Data
  - Assumption: next Data returns from the similar path of current inflight Data
- Obtain the latest RTT and update LDT with it

CCNinfo (RFC9344) will be a good alternative to obtain path info

Evaluation

We evaluated performance of NDN-PTO with ndnSIM

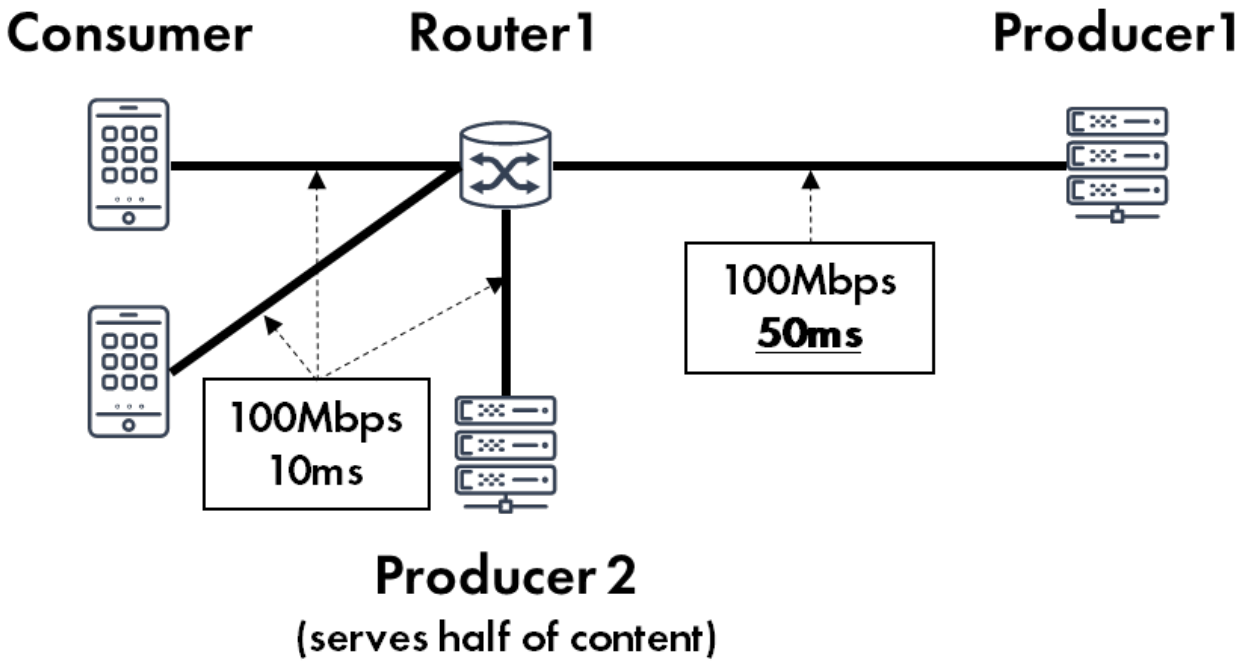
We compare NDN-PTO and PCON with four minRTO settings

Metrics

1. Goodput
2. Packet overhead (# of retransmission, # of spurious timeouts)
3. Data retrieval delay (data reception time – 1st requested time)
   - Rapid loss detection can shorten this delay

Evaluation scenario1: Switching of producer

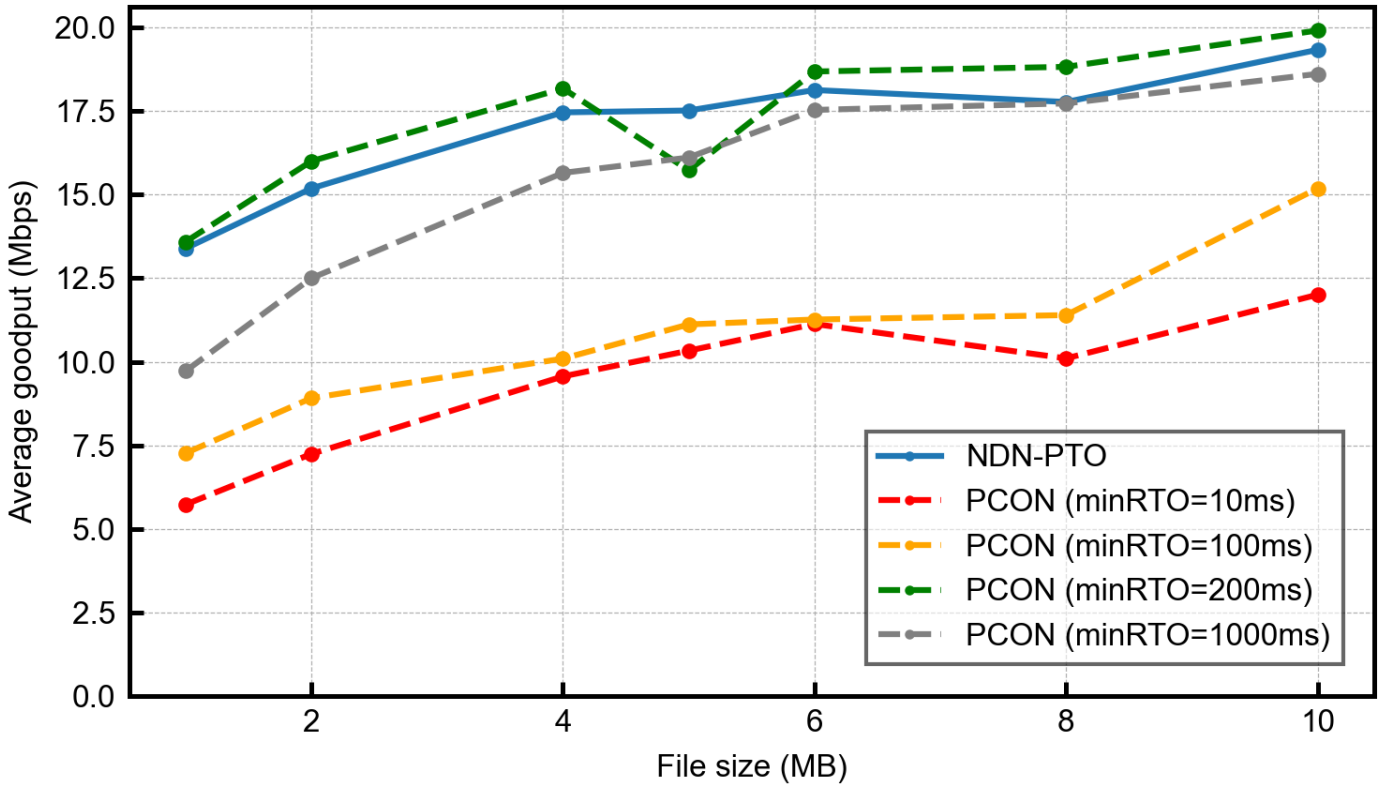
Performance impact of STO



After downloading half of content, RTT changes significantly → **STO**

KDDI Research

Scenario1: Goodput



Goodput of PCON: heavily depends on the minRTO parameter

PCON with minRTO=200ms achieved best goodput (avoid STOs)

NDN-PTO achieved comparable goodput for the optimal PCON (<6%) w/o configuring minRTO

※Optimal minRTO in this topology: 200ms (larger than the RTT for remote producer:120ms)

KDDI Research

Scenario1: Overhead and Delay

Overhead

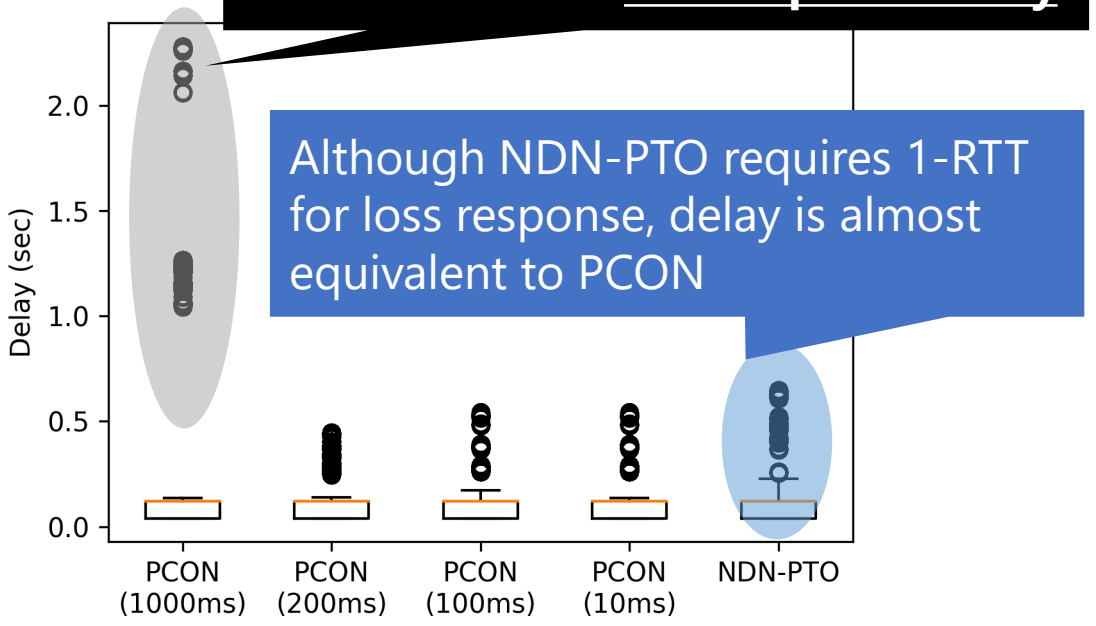
STO occurred in the PCON with low minRTO

File size		1 MB	5 MB	10 MB
Average overhead	NDN-PTO	893	804.5	845.5
	PCON (10 ms)	1081.5	812	914
	PCON (100 ms)	1056	850.5	984.5
	PCON (200 ms)	850.5	981	1256.5
	PCON (1000 ms)	850.5	1856.5	2885
Average # of STO	NDN-PTO	3	3	2.5
	PCON (10 ms)	204	332.5	450.5
	PCON (100 ms)	229.5	369	494.5
	PCON (200 ms)	0	0	0
	PCON (1000 ms)	0	0	0

NDN-PTO dramatically **reduced** the STO and achieved low overhead

Delay

PCON with long minRTO has long outliers due to its **loss response delay**



KDDI Research

Conclusion

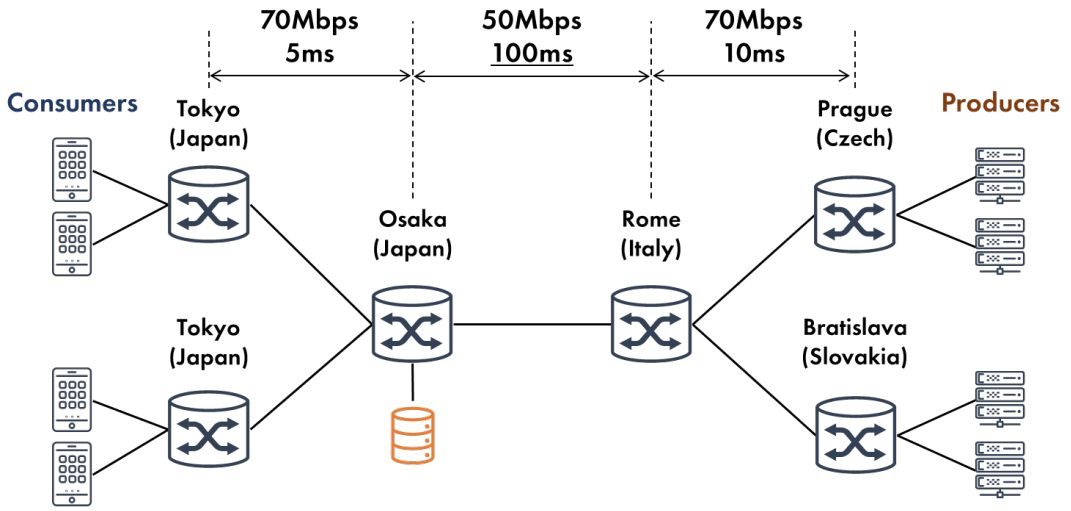
- End host-based loss detection: promising backup plan, but timer-based schemes cause **spurious timeout**
- Proposal: **NDN-PTO** obtain the latest RTT information with "**probe Interest**" to detect packet loss accurately
- From the simulation, NDN-PTO achieved both accurate loss detection and higher goodput **without topology-dependent parameters**
- **Next steps:**
  - Evaluate the performance in the actual testbed
  - Evaluate the interaction between NDN-PTO and the In-network ReTx

KDDI Research


KDDI
KDDI Research

Scenario2: Global content distribution

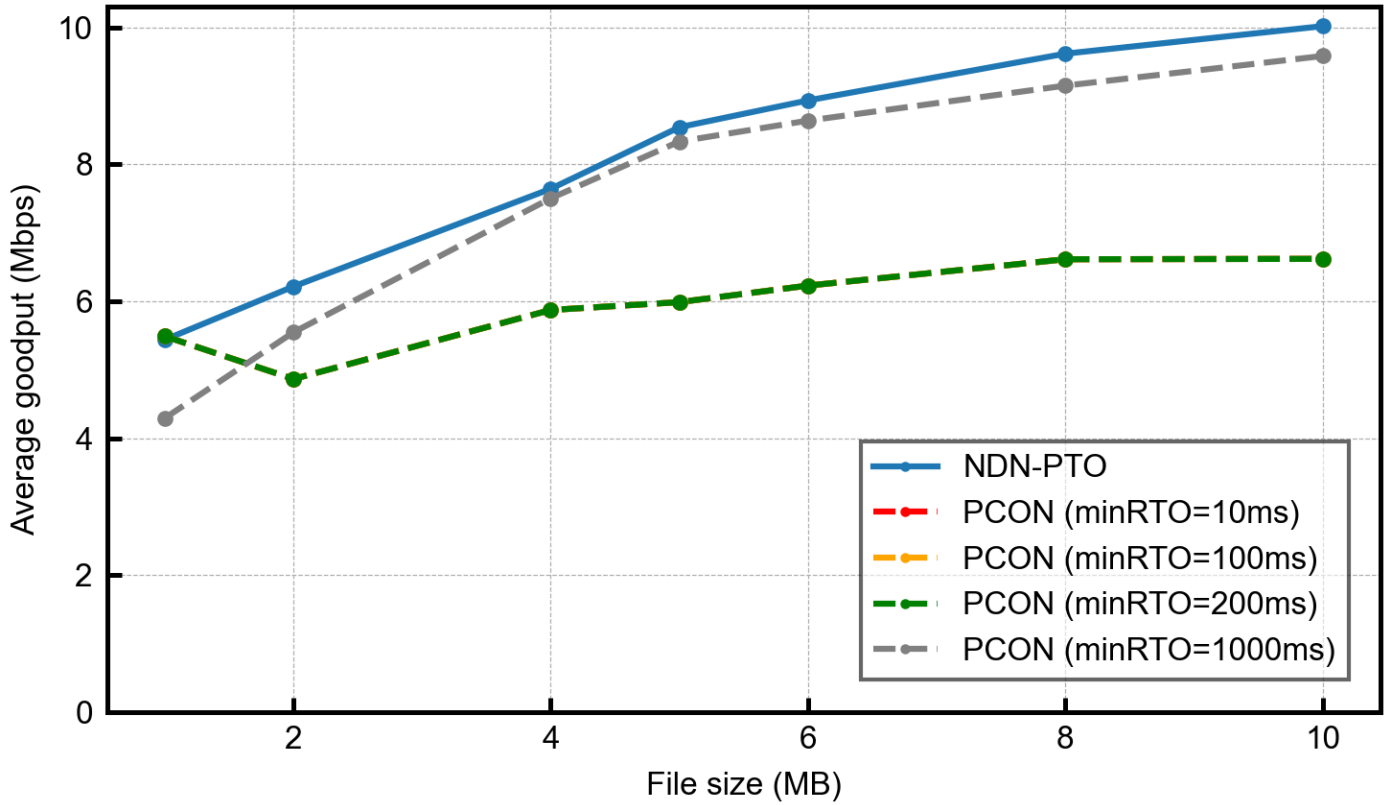
Performance of general content distribution



This dumbbell topology follows the actual **EU-JP global testbed**

KDDI Research

Scenario2: Goodput



NDN-PTO achieved
the best goodput

PCON with minRTO ≦ 200ms
achieved low goodput
(They suffer from STO)

※Optimal minRTO in this topology: 1000ms (larger than the RTT for remote producer:230ms)

KDDI Research

Scenario2: Overhead and delay

Overhead

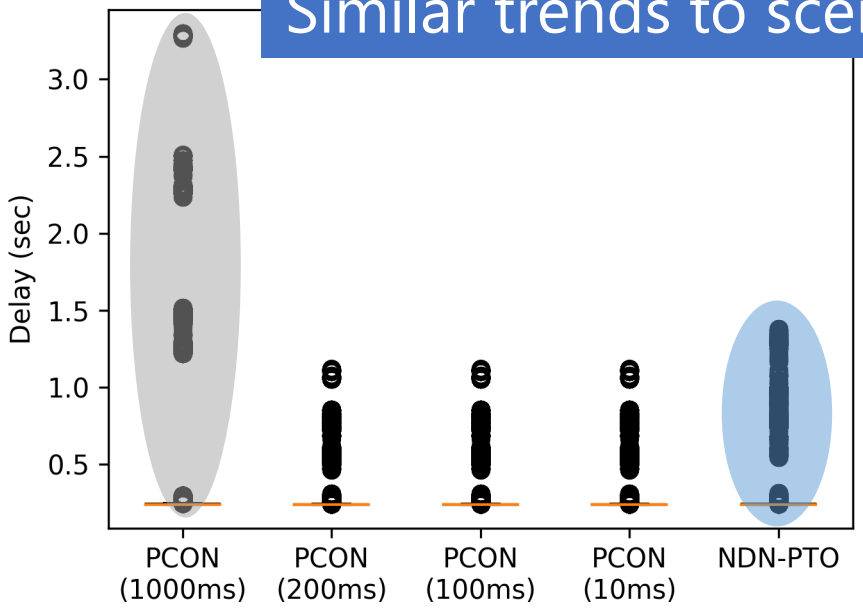
PCON with minRTO ≦ 200ms suffered from STOs in global content delivery

File size		1 MB	5 MB	10 MB
Average overhead	NDN-PTO	2594.75	1666.25	1711.5
	PCON (10 ms)	2560.5	1418	1439.75
	PCON (100 ms)	2560.5	1418	1439.75
	PCON (200 ms)	2560.5	1418	1439.75
	PCON (1000 ms)	2951	2541.75	2272.75
Average # of STO	NDN-PTO	1.25	5.75	12.25
	PCON (10 ms)	10	156.25	134.5
	PCON (100 ms)	10	156.25	134.5
	PCON (200 ms)	10	156.25	134.5
	PCON (1000 ms)	0	0	0

NDN-PTO reduced the STO and achieved low overhead

Delay

Similar trends to scenario 1



KDDI Research