

HF²T: Host-Based Flowlet Fine-Tuning for RDMA Load Balancing

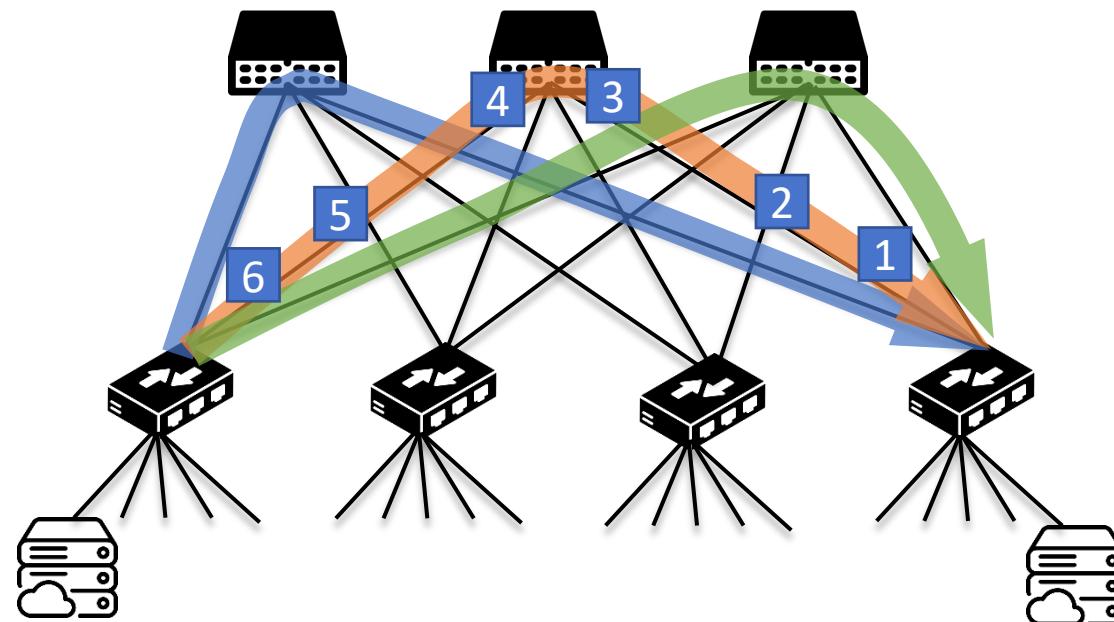
(APNet'24)

Chuhao Chen, Jiarui Ye, Yongbo Gao, Sen Liu, Yang Xu

Fudan University

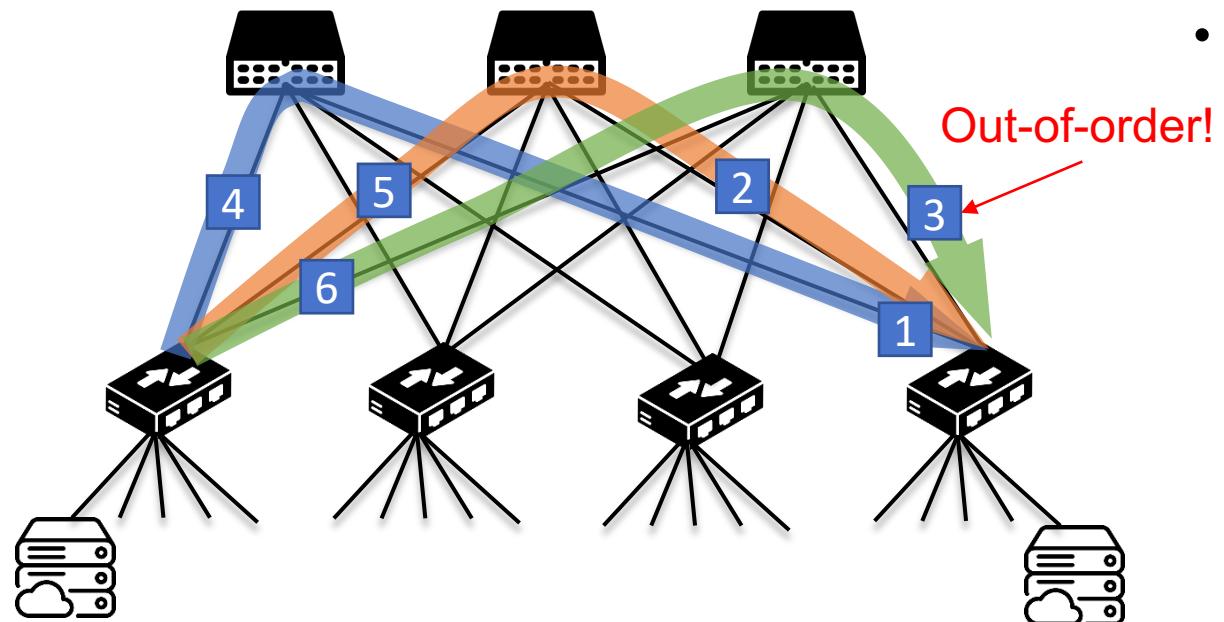


Load Balancing in Data Centers



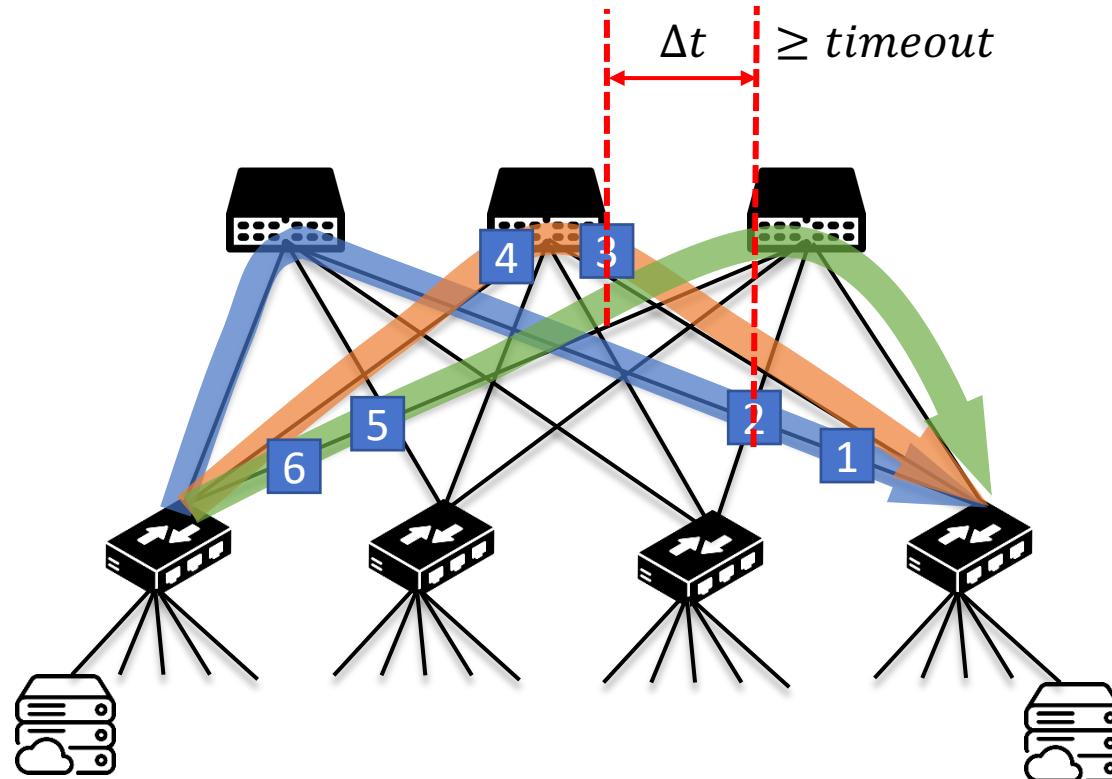
- Multiple Equal-Cost Paths
- Load Balancing Schemes
 - Flow-level
 - Fixed path, low flexibility

Load Balancing in Data Centers



- Multiple Equal-Cost Paths
- Load Balancing Schemes
 - Flow-level
 - Fixed path, low flexibility
 - Packet-level
 - Out-of-order Packets

Load Balancing in Data Centers



- Multiple Equal-Cost Paths
- Load Balancing Schemes
 - Flow-level
 - Fixed path, low flexibility
 - Packet-level
 - Out-of-order Packets
-  Flowlet-level
 - Utilize parallel paths while avoiding out-of-order packets

RDMA Load Balancing

RDMA(Remote Direct Memory Access) has been widely applied in modern data centers!



High-Performance Computing



Distributed Storage



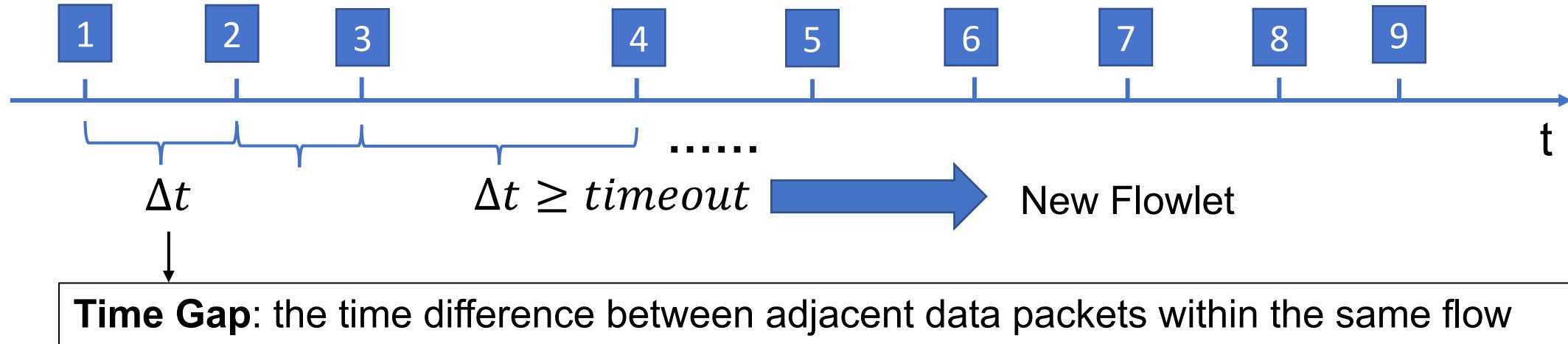
Machine Learning



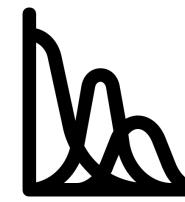
The flowlet-level load balancing that performs well in TCP networks fails in RDMA networks!

(SIGCOMM'23 "Network Load Balancing with In-network Reordering Support for RDMA")

Measure Packet Time Gaps



Task:
Measure time gaps in RDMA
and TCP respectively



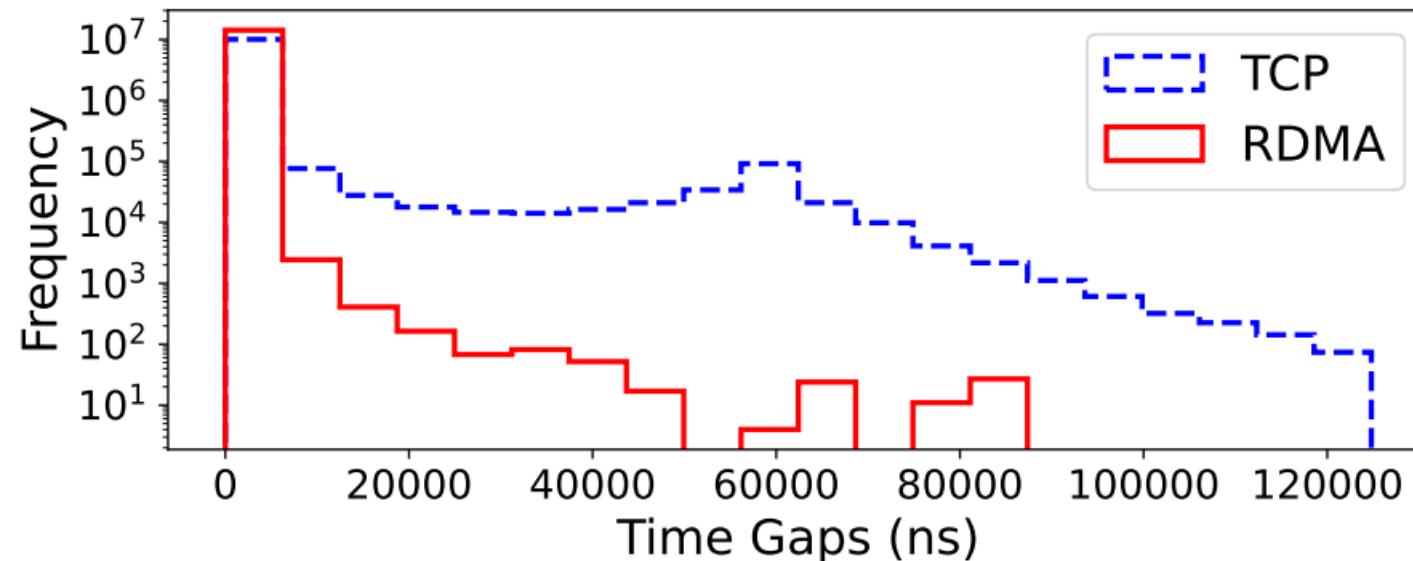
Time Gap Distribution



Time Gaps Statistics Table

Traffic Pattern in RDMA

Time Gap Distribution of RDMA and TCP



→ Compared to TCP, RDMA has:

- Time gaps cluster more narrowly
- Smaller numerical ranges, lacks “larger” time gaps

Traffic Pattern in RDMA

Time Gaps Statistics Table:

calculate the proportion of larger time gaps which exceeds flowlet timeout

Protocol \ Time Gap	$\geq RTT$	$\geq 2RTT$	$\geq 3RTT$
RDMA	0.012%	0.006%	0.003%
TCP	2.376%	1.854%	1.550%

RDMA / TCP

1/200

1/300

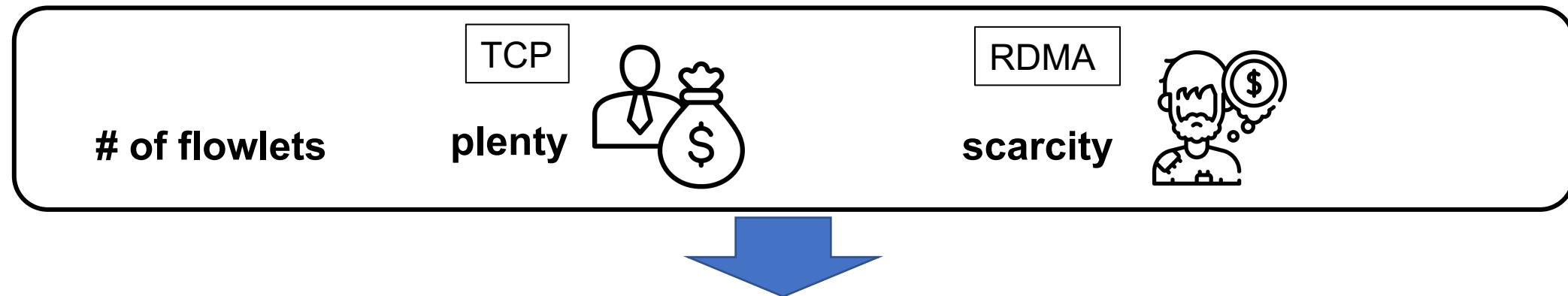
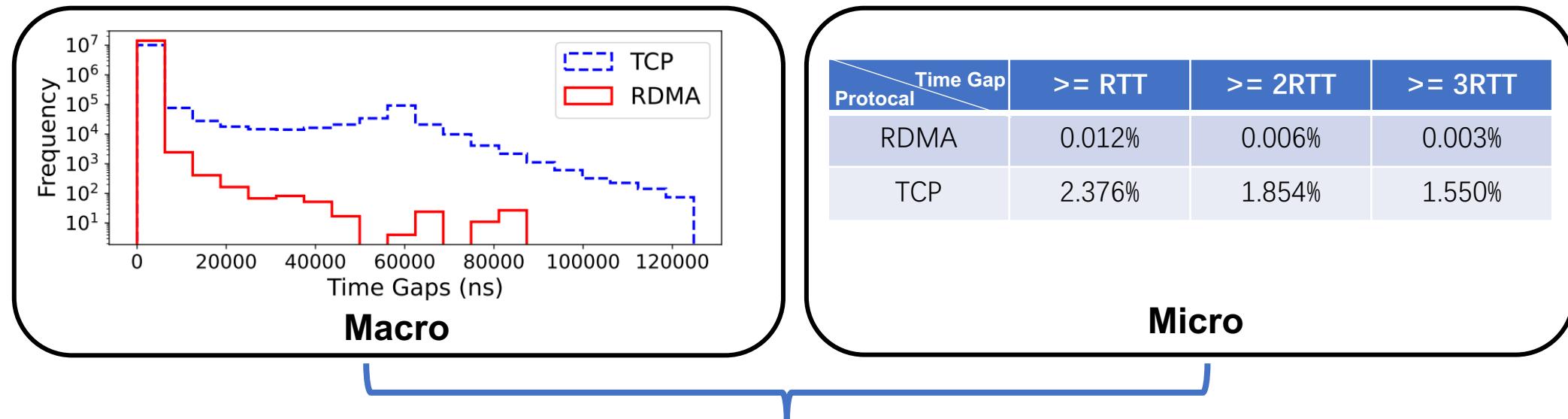
1/500



Compared to TCP, RDMA has:

- Time gaps larger than the flowlet timeout value are rare
- Constitute only a fraction of those in TCP, averaging 1/300

Traffic Pattern in RDMA

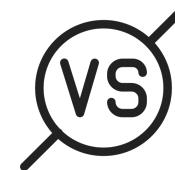


RDMA vs TCP

TCP and RDMA use different software and hardware technologies:

TCP

- window-based packet transmission
- batch optimization of ACKs
- TCP Segmentation Offload



RDMA

- rate-based packet transmission
- hardware-based pacing per connection
- kernel bypassing and polling

**TCP naturally forms flowlets,
whereas RDMA tends to exhibit more continuous traffic patterns.**

How to generate more flowlets in RDMA?

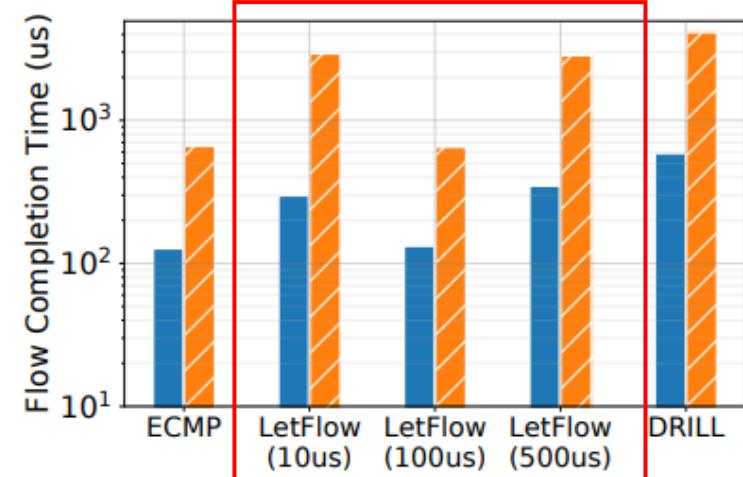


What about decreasing the flowlet timeout value?

Out-of-order packets issues

→ Performance degradation

(SIGCOMM'23 "Network Load Balancing with In-network Reordering Support for RDMA")



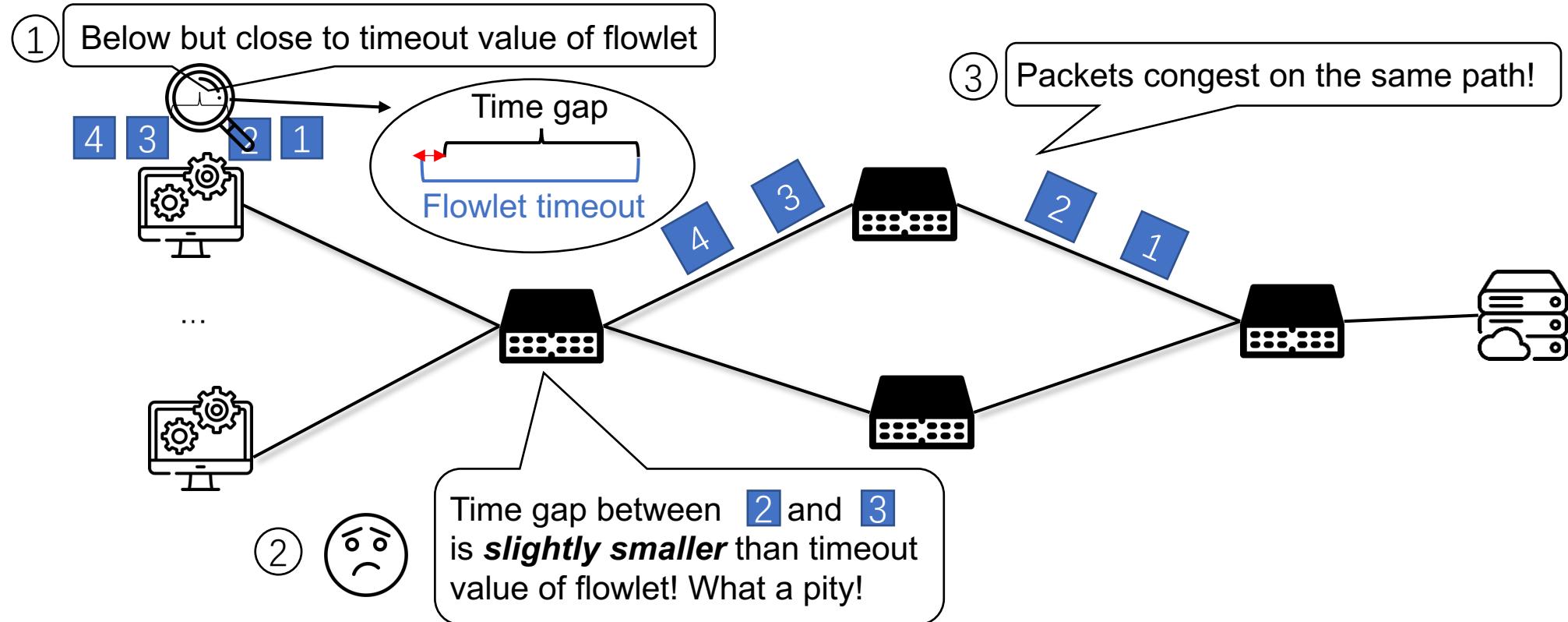
Is there a way to manually create flowlets?



Goals:

1. Produce more flowlets
2. Maintain the advantages of RDMA(high throughput, low latency)
3. Compatible with existing commodity switches that support flowlet load balancing

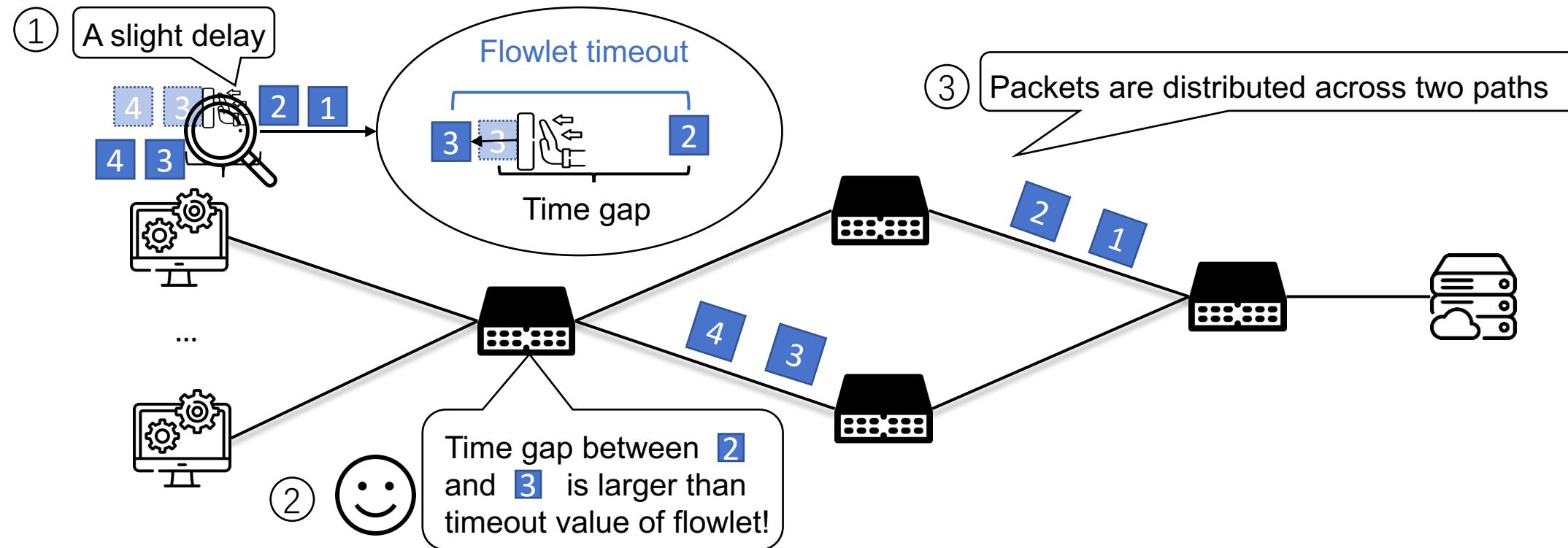
An Opportunity Brought by Time Gap



It's also an opportunity, just seize it:

A minor delay can produce a new flowlet and create a valuable opportunity for rerouting!

Seize the Opportunity

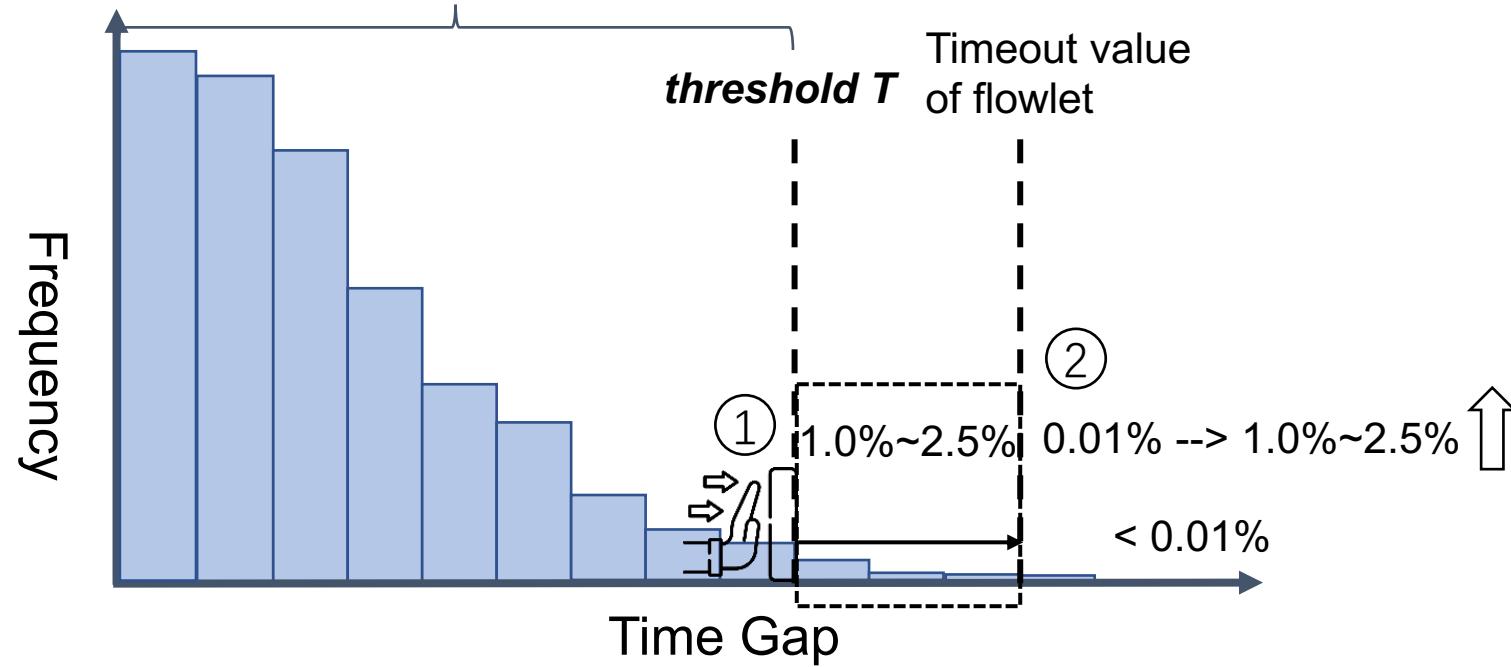


- ✓ Goal 2: Maintain the advantages of RDMA (The delay is minor compared to the flowlet timeout)
- ✓ Goal 3: Compatible with existing flowlet-enable commodity switches (Host-based method)

Elongate Gaps Based on Distribution

At the host:

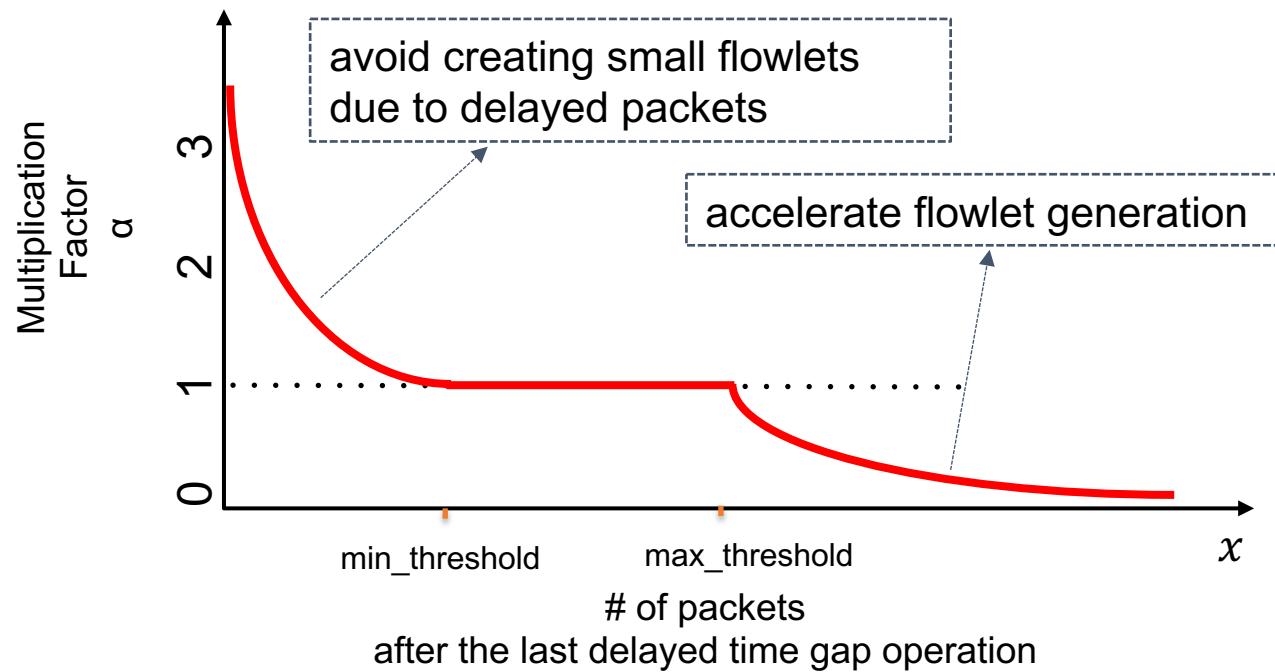
- ③ The vast majority of traffic still maintains the advantages of RDMA.



- Goal 1: Produce more flowlets (Increase the number of flowlets by 100x ~ 250x)
- Goal 2: Maintain the advantages of RDMA (Delay a small number of specific packets)

Optimization

Goal: Optimize the length of flowlet



- Use multiplication factor α to dynamically adjust threshold T
- $$\text{time gap} \geq \text{threshold} * \alpha$$
- Avoid proactively creating small flowlets
 - A lot of packet reordering
 - An accumulative increase in FCT
 - Prevent excessively long flowlets
 - Suboptimal load-balancing effects

Evaluation

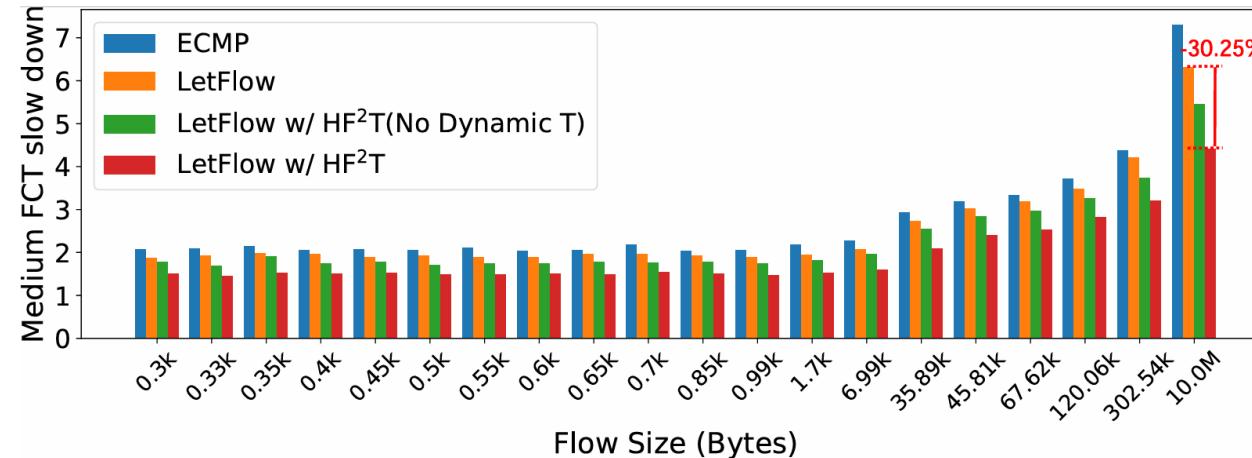
Goal

- Validate whether deploying the HF²T at the host could enhance the performance of flowlet-level load balancing schemes in RDMA networks.

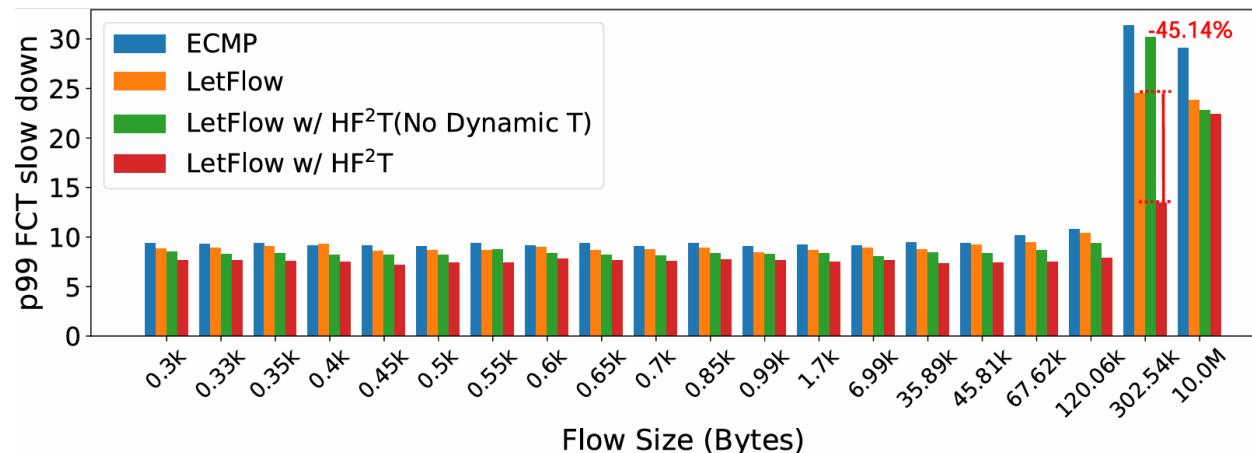
NS3 Simulation

- Workloads: Web Search and Meta Hadoop
- Topology: a Fat-Tree topology with k=4
- Congestion Control: DCQCN
- Comparisons: ECMP, LetFlow, LetFlow with HF²T(No Dynamic T) and LetFlow with HF²T
- NIC: 100Gbps; Links: 100Gbps with a latency of 1us.

Performance of HF²T



When compared with LetFlow, LetFlow with HF²T reduces the Medium FCT by **22%** and the 99-percentile FCT by **16%** on average.



Other workloads in Paper...

Internal Metrics

Scheme	# of flowlets	# of PFC
LetFlow	33583	3568
LetFlow w/ HF ² T(No Dynamic T)	65865	2574
LetFlow w/ HF ² T	138104	844

(PFC: Priority-based Flow Control)



The utilization of HF²T

- increases the number of flowlets by **311%**
- reduces the occurrences of PFC by **76%**

Conclusion

Dilemma of RDMA load balancing:

- Flowlet-level load balancing is powerful in TCP networks but it fails in RDMA networks.
- Time gaps between packets larger than the flowlet timeout is notably scarce in RDMA.

Our Solution:

- HF²T: Host-Based Flowlet Fine-Tuning for RDMA load balancing
 - postpone a minimal number of specific packets to greatly promote flowlet generation
 - a modest cost for a substantial gain in rerouting opportunities
 - deployment-friendly: only require modification at the end host

chench23@m.fudan.edu.cn

Q&A

chench23@m.fudan.edu.cn