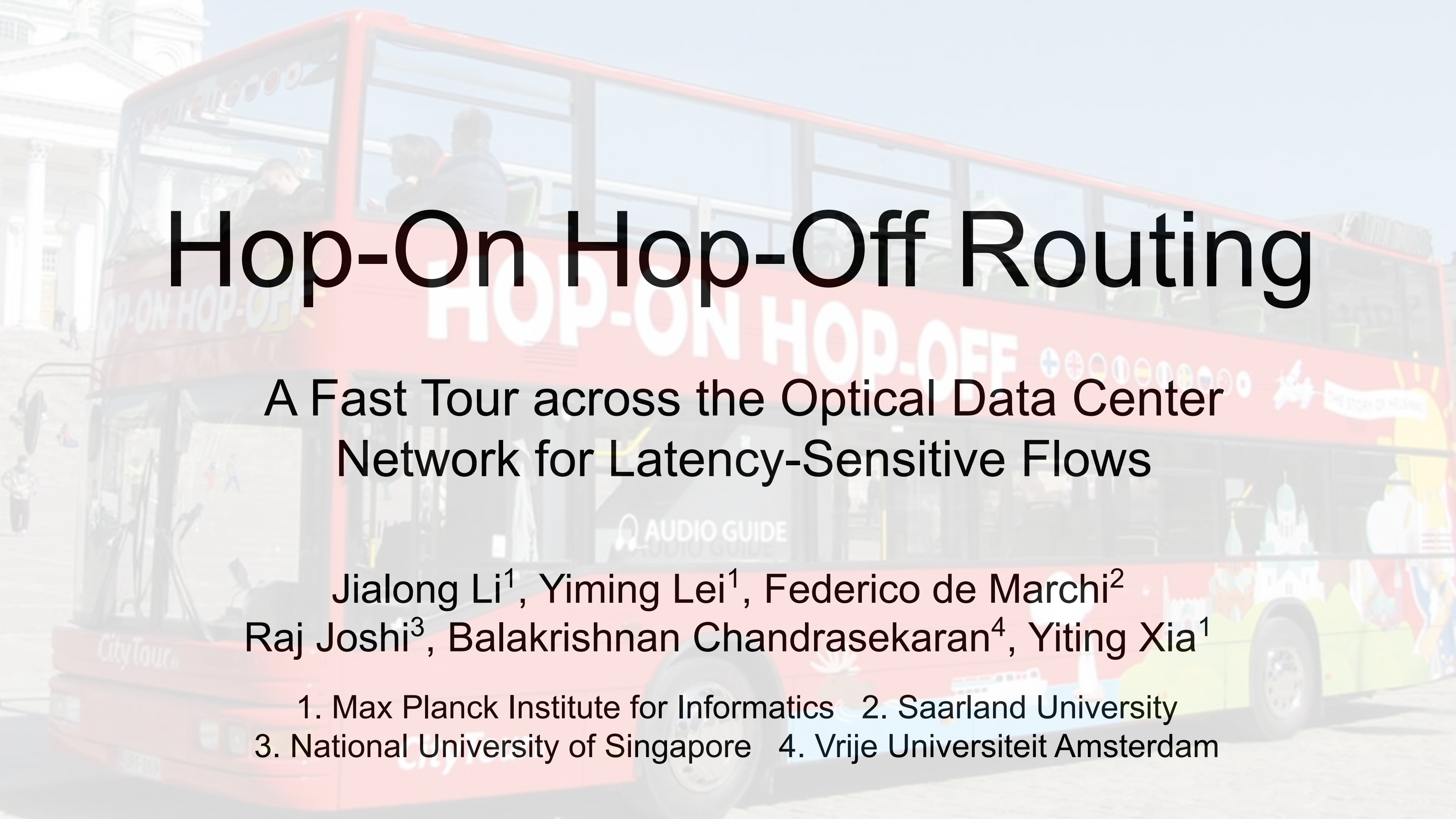


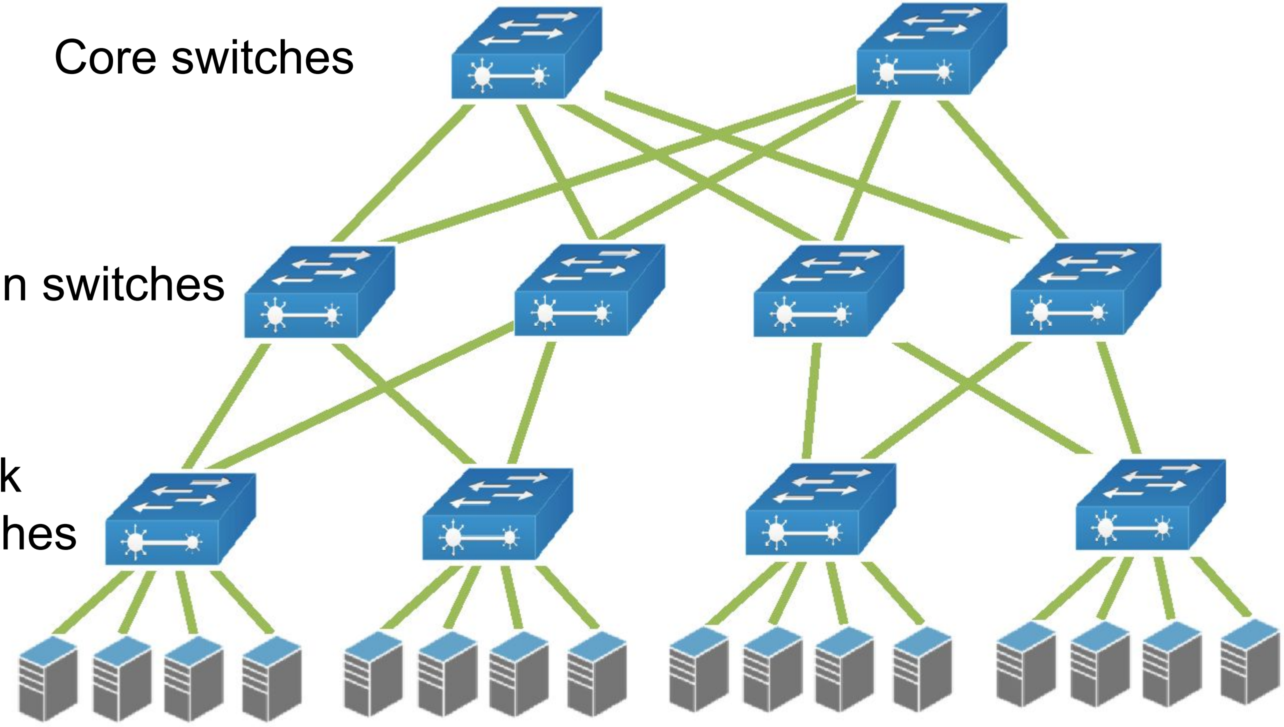
Hop-On Hop-Off Routing

A Fast Tour across the Optical Data Center Network for Latency-Sensitive Flows

Jialong Li[1], Yiming Lei[1], Federico de Marchi[2]
Raj Joshi[3], Balakrishnan Chandrasekaran[4], Yiting Xia[1]

1. Max Planck Institute for Informatics 2. Saarland University
3. National University of Singapore 4. Vrije Universiteit Amsterdam

Today’s Data Center Networks

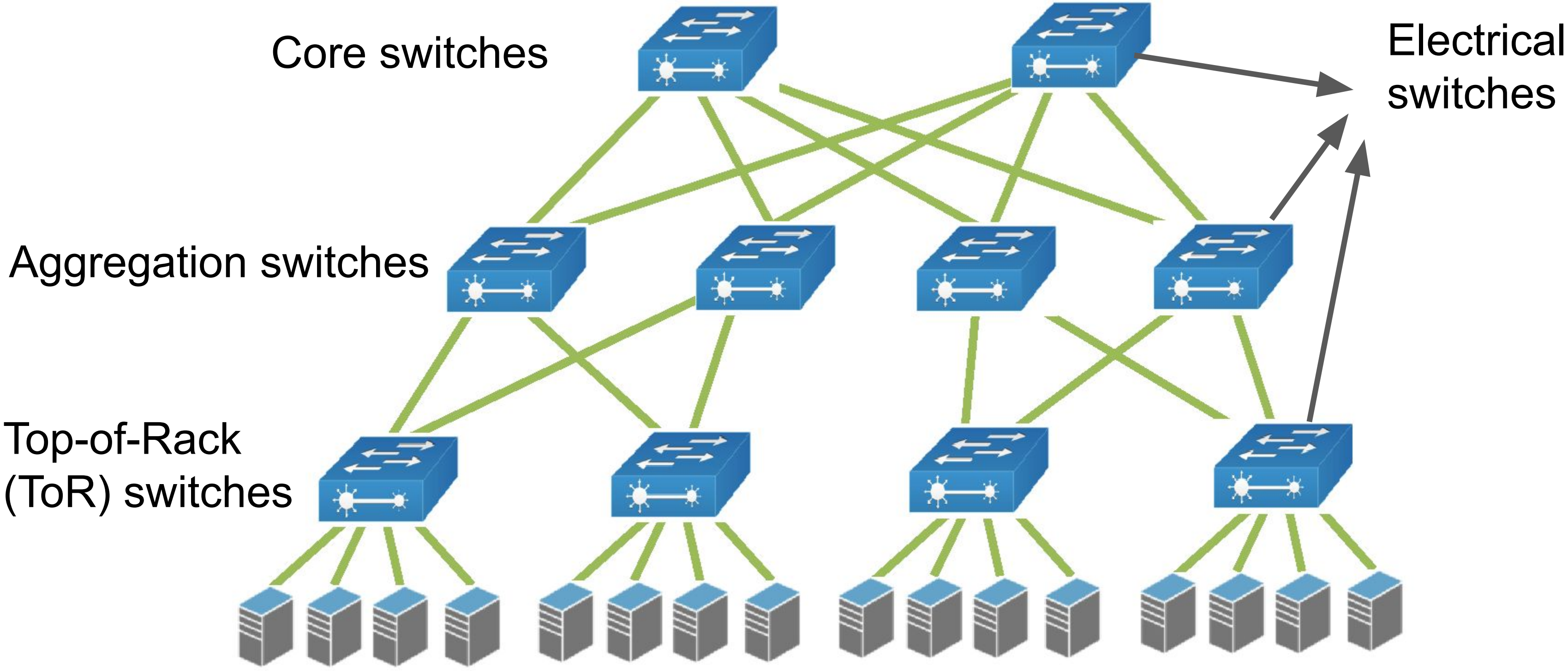


Core switches

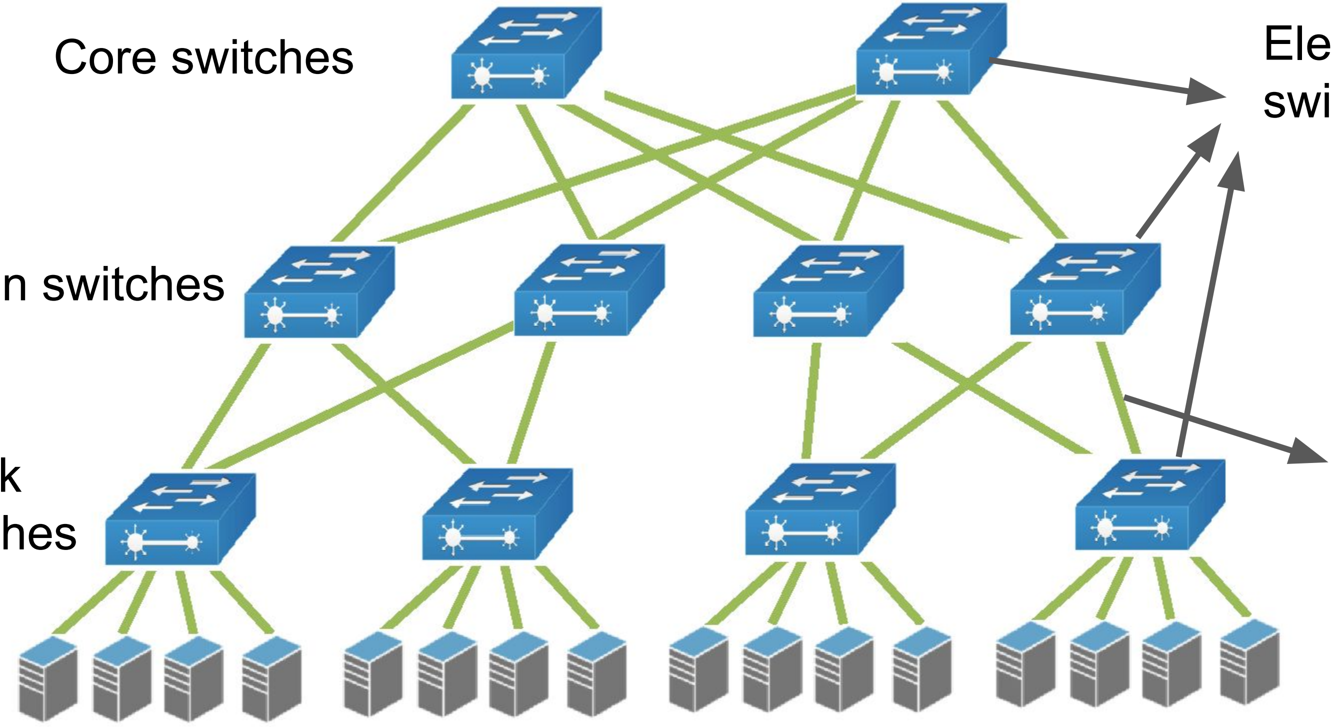
Aggregation switches

Top-of-Rack (ToR) switches

Today’s Data Center Networks



Today’s Data Center Networks



Core switches

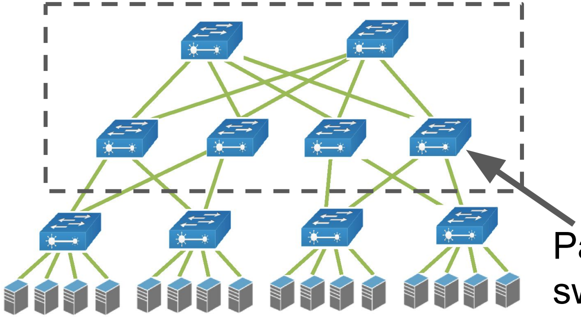
Aggregation switches

Top-of-Rack
(ToR) switches

Electrical
switches

Optical
fibers

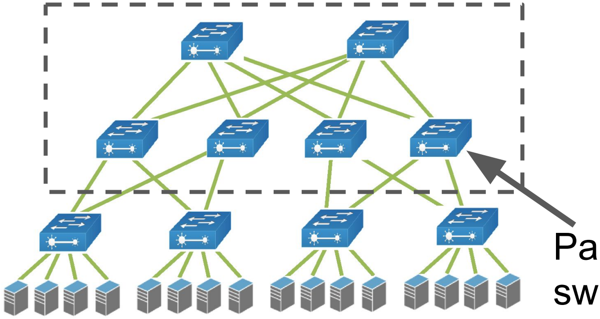
From Electrical Switching to Optical Switching



Packet buffer, switching…

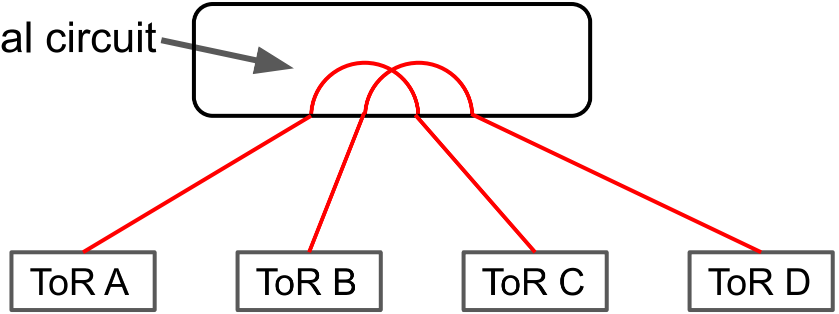
Electrical Switching

From Electrical Switching to Optical Switching



Packet buffer, switching…

Optical switch



Optical circuit

ToR A

ToR B

ToR C

ToR D

Electrical Switching

↓

Optical Switching

Why Optical Switching

- No queuing delay along the circuit

Why Optical Switching

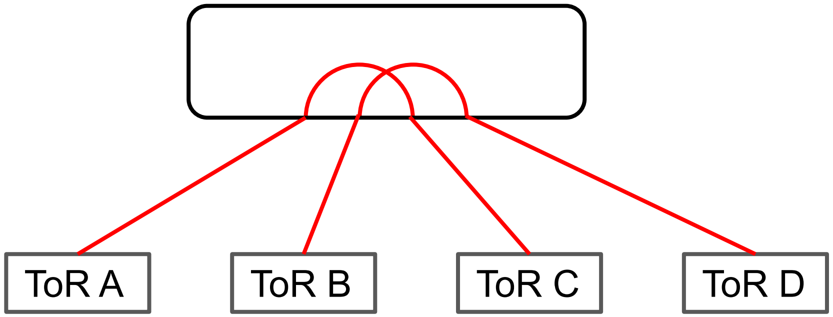
- No queuing delay along the circuit

- Less power consumption

Why Optical Switching

- No queuing delay along the circuit
- Less power consumption
- Independent to link bandwidth

Optical Switching

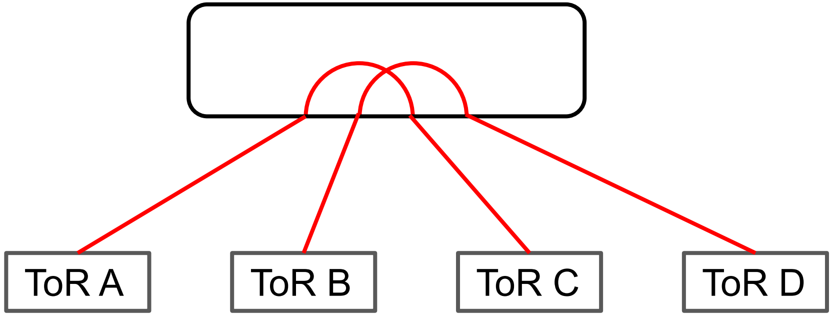


Time slot 0 (time 0 - T)

A ⇔ C

B ⇔ D

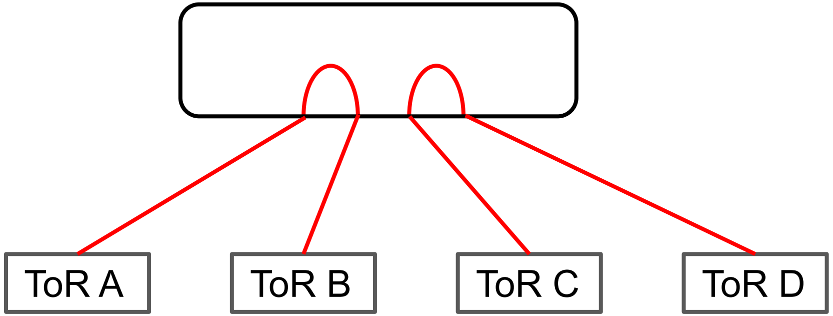
Optical Switching



Time slot 0 (time 0 - T)

A ⇔ C

B ⇔ D

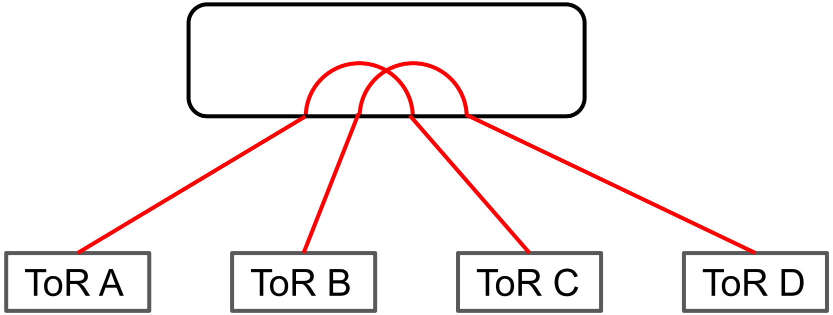


Time slot 1 (time T - 2T)

A ⇔ B

C ⇔ D

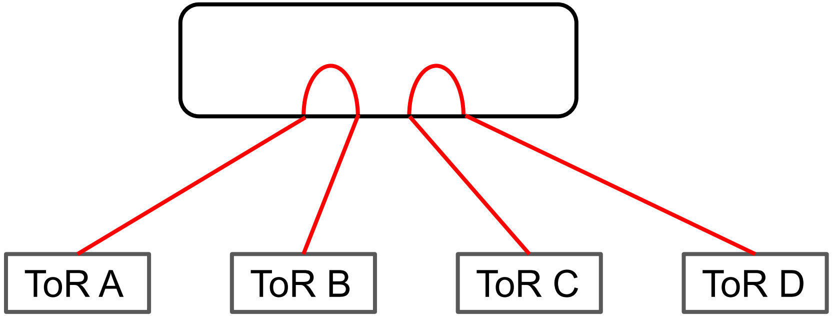
Optical Switching



Time slot 0 (time 0 - T)

A ⇔ C

B ⇔ D



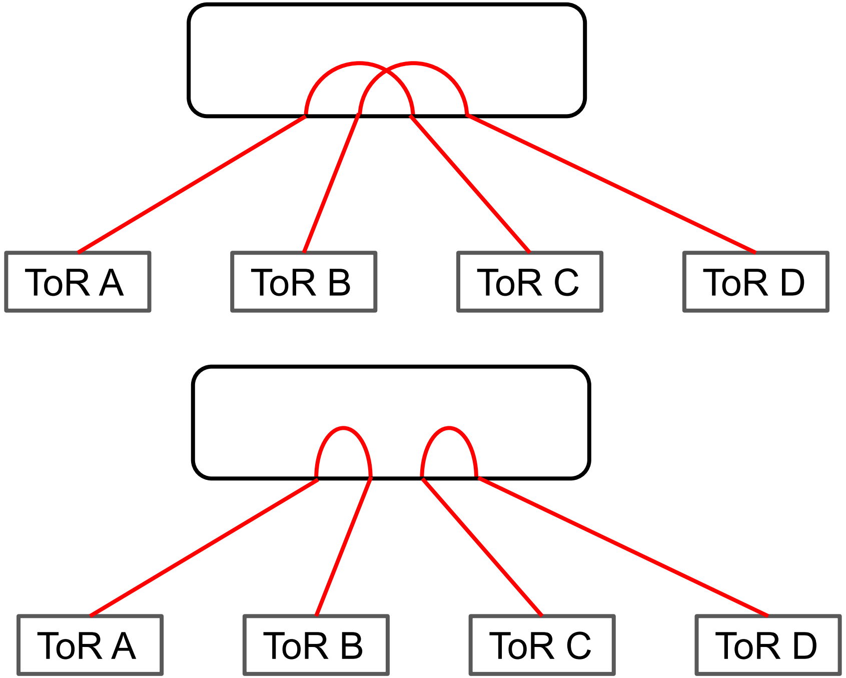
Time slot 1 (time T - 2T)

A ⇔ B

C ⇔ D

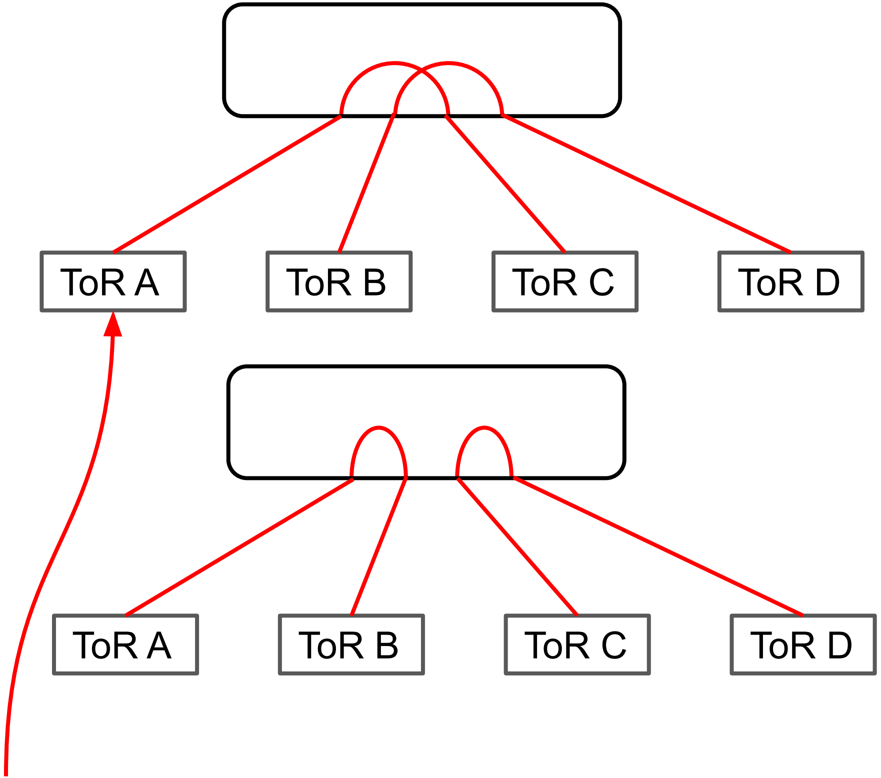
Time slot: When and how long the circuit exists

Optical Switching



- A packet arrives at time 0
- Src = A, dst = B
- Waiting at A for time T

Optical Switching



- A packet arrives at time 0
- Src = A, dst = B
- Waiting at A for time T

Circuit waiting delay: delay at src ToR before required circuit comes

Latency-Sensitive Flows

Latency-Sensitive Flows: Flow Completion Time (FCT) is critical

Latency-Sensitive Flows

Latency-Sensitive Flows: Flow Completion Time (FCT) is critical

FCT: tens of *ns* to hundreds of *μs*[1]

Circuit waiting delay: tens of *μs* to *ms*

[1] Expanding across time to deliver bandwidth efficiency and low latency, NSDI' 20

Latency-Sensitive Flows

Latency-Sensitive Flows: Flow Completion Time (FCT) is critical

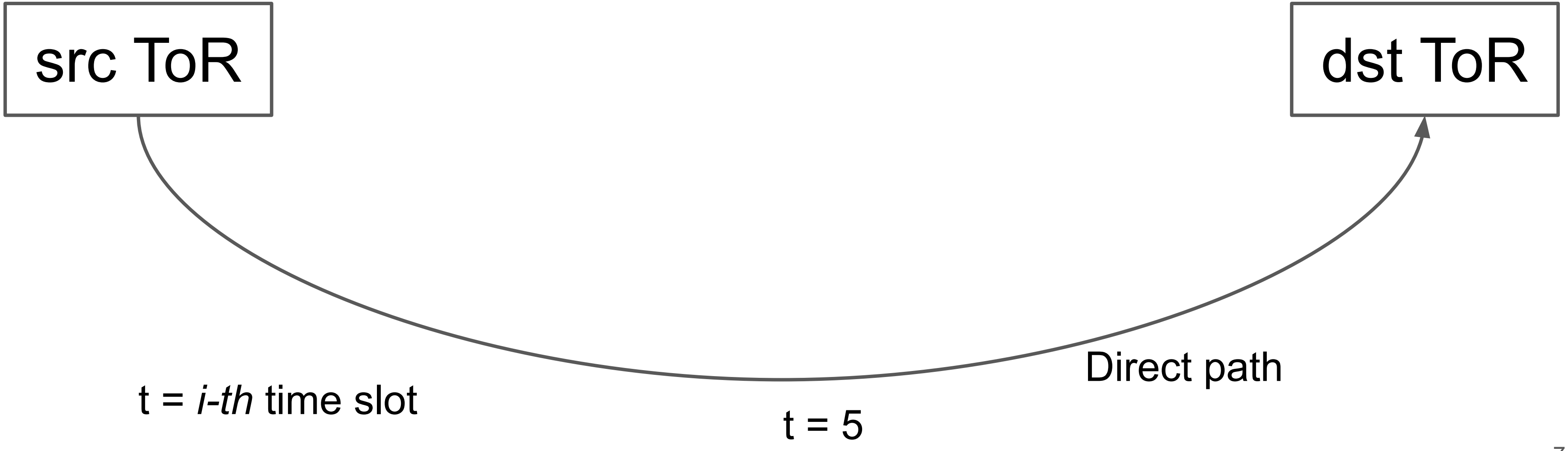
FCT: tens of *ns* to hundreds of *μs*[1]

Circuit waiting delay: tens of *μs* to *ms*

Reduce circuit waiting delay: Using multi-hop paths

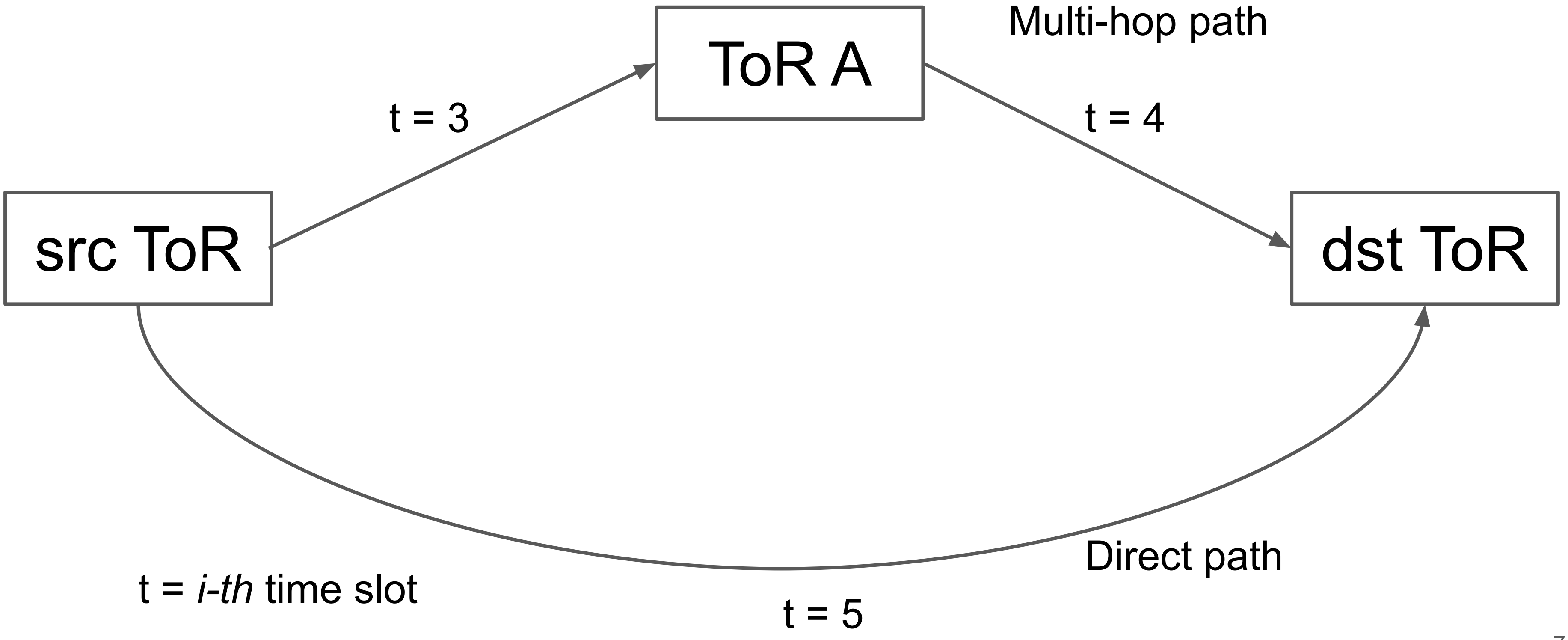
[1] Expanding across time to deliver bandwidth efficiency and low latency, NSDI' 20

Multi-Hop Path



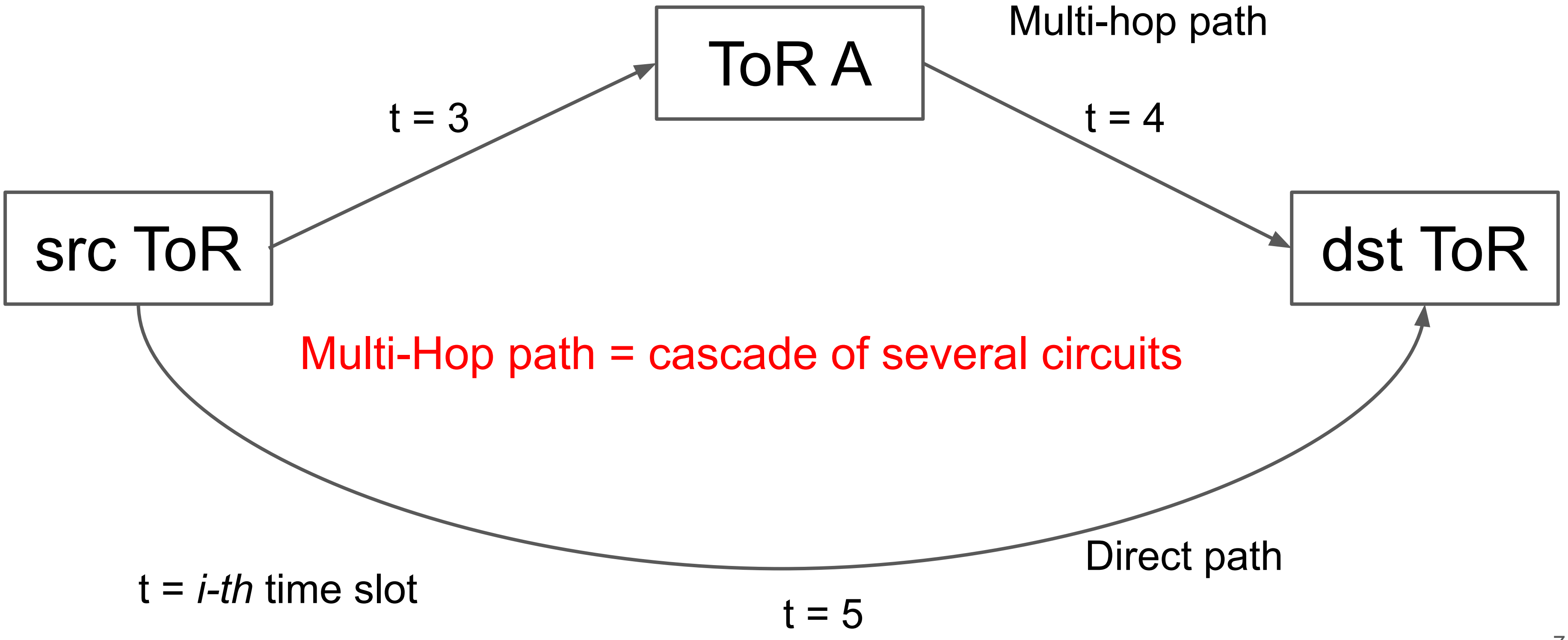
t = *i-th* time slot

Multi-Hop Path



t = *i-th* time slot

Multi-Hop Path



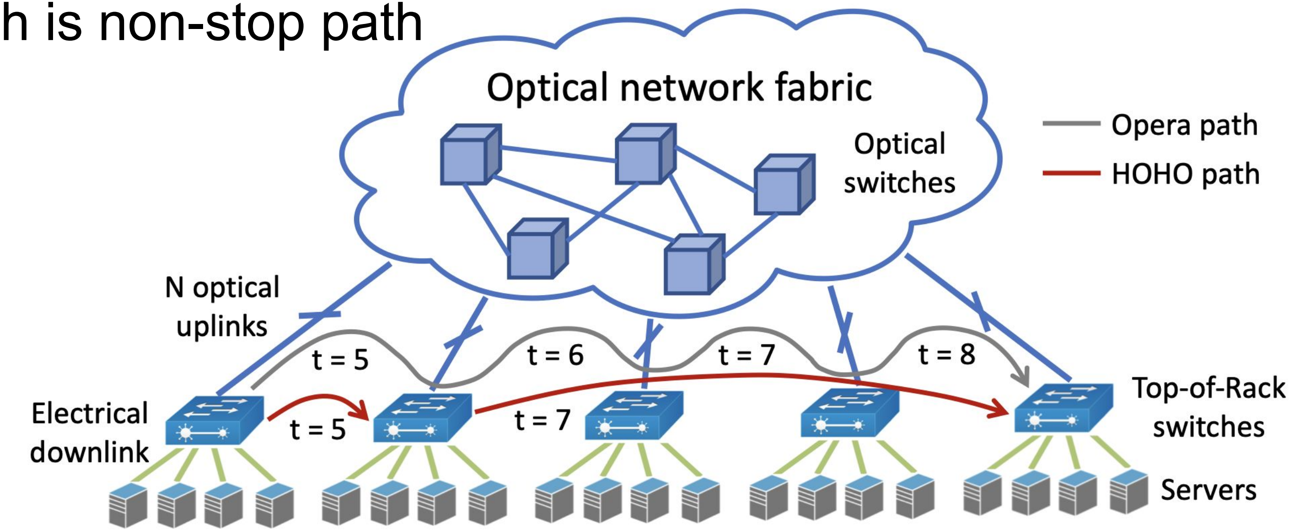
Multi-Hop path = cascade of several circuits

t = *i-th* time slot

State-of-the-art: Opera (NSDI’ 2020)

Opera

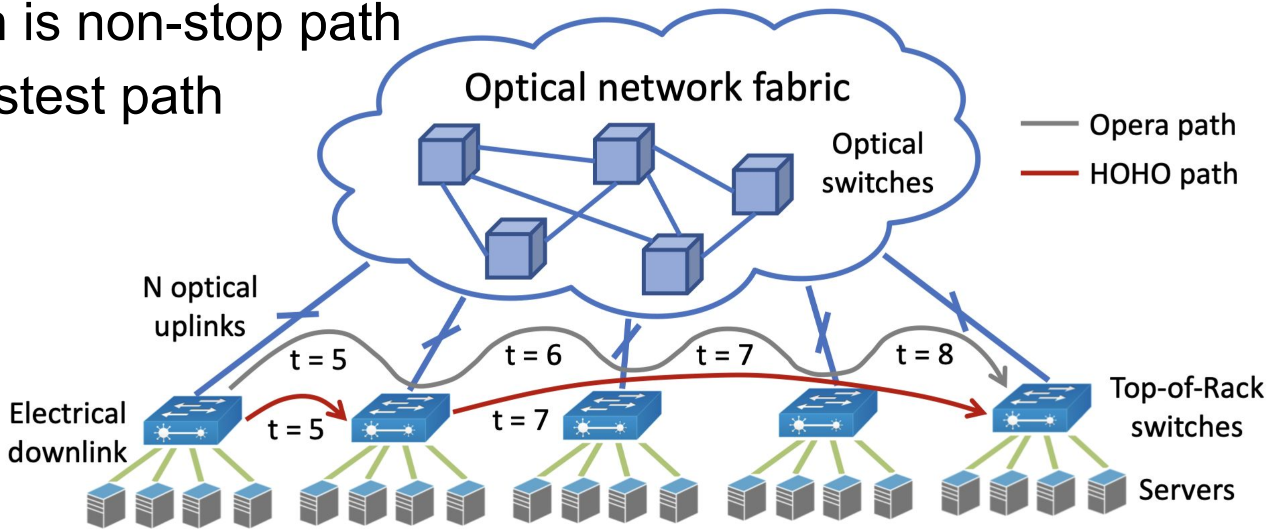
- A multi-hop path is available at any moment between any ToR pairs
- This multi-hop path is non-stop path



State-of-the-art: Opera (NSDI’ 2020)

Opera

- A multi-hop path is available at any moment between any ToR pairs
- This multi-hop path is non-stop path
- Non-stop path ≠ fastest path



Hop-On Hop-Off (HOHO) Routing

- Search for the fastest path, instead of non-stop paths
- Packets could wait at ToRs, so they can "hop on" a circuit, "hop off", and "hop on" another circuit

HOHO Routing: Intuition

- The latency of a path is only determined by the time slot of the last-intermediate ToR and dst ToR

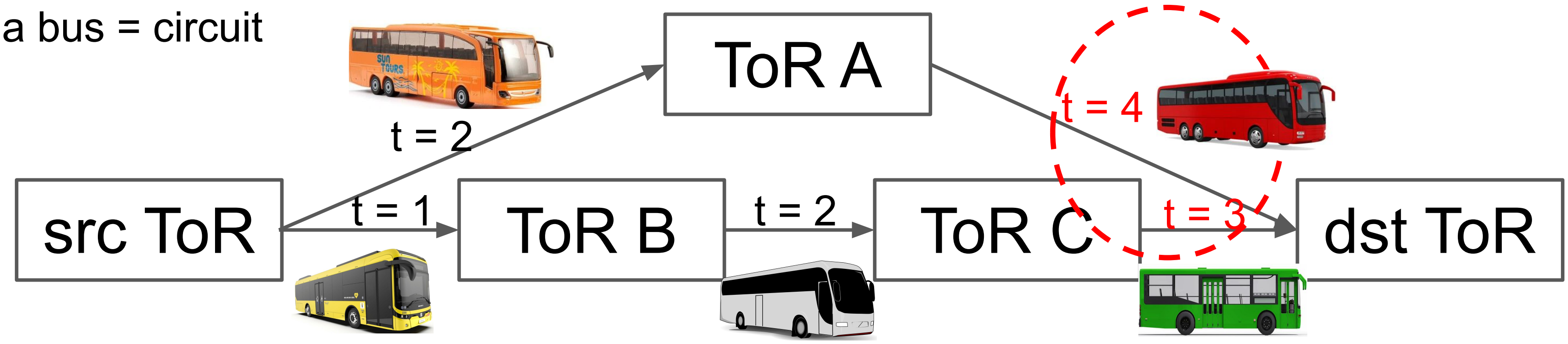
HOHO Routing: Intuition

- The latency of a path is only determined by the time slot of the last-intermediate ToR and dst ToR

t = *i-th* time slot
a bus = circuit

Path 1: src => A => dst

Path 2: src => B => C => dst



HOHO Routing: Intuition

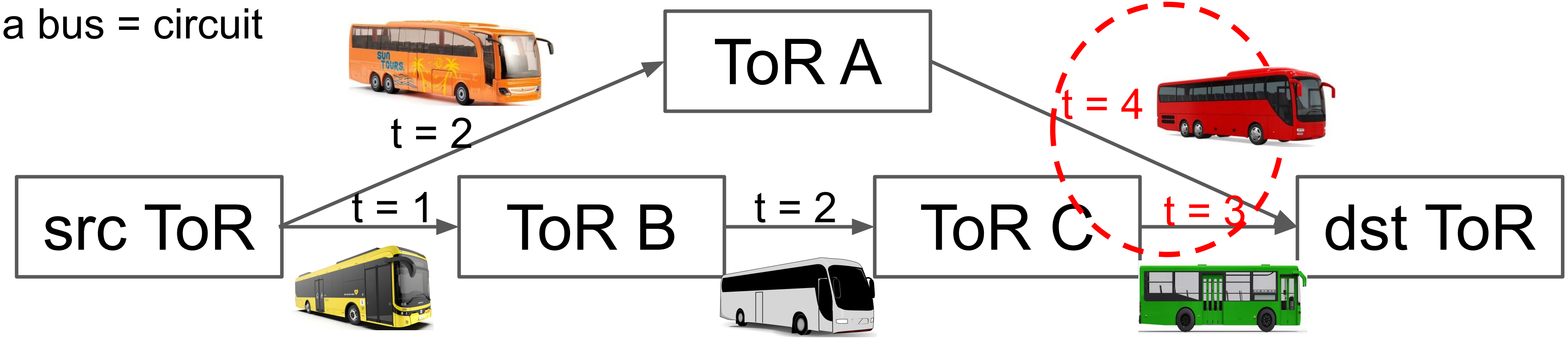
- The latency of a path is only determined by the time slot of the last-intermediate ToR and dst ToR
- Backtracking: Search from dst ToR to src ToR

Path 1: src => A => dst

Path 2: src => B => C => dst

t = *i-th* time slot

a bus = circuit



HOHO Routing

Time Slot	Circuits
1	S – H
2	A – F
3	H – E, F – A, S – B
4	S – G, A – D
5	B – D, G – B
6	S – D
7	S – A
8	C – D

Fixed, periodic optical schedule

- Time slot index indicating when two ToRs have a circuit

Input

- A fixed, periodic optical schedule
- Src and dst ToR
- Packet arrival time slot
- Maximum hops (optional)

HOHO Routing

Time Slot	Circuits
1	S – H
2	A – F
3	H – E, F – A, S – B
4	S – G, A – D
5	B – D, G – B
6	S – D
7	S – A
8	C – D

Fixed, periodic optical schedule

- Time slot index indicating when two ToRs have a circuit

Input

- A fixed, periodic optical schedule
- Src and dst ToR
- Packet arrival time slot
- Maximum hops (optional)

Output

- A fastest path between src and dst ToR

Rerun per ToR pair, per packet arrival time slot

HOHO Routing

Send a packet from S to D in the minimal time within max hops, max = 3 here

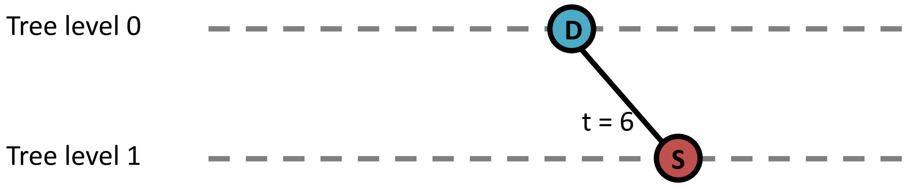
Time Slot	Circuits
1	S – H
2	A – F
3	H – E, F – A, S – B
4	S – G, A – D
5	B – D, G – B
6	S – D
7	S – A
8	C – D

Tree level 0
D

HOHO Routing

Send a packet from S to D in the minimal time within max hops, max = 3 here

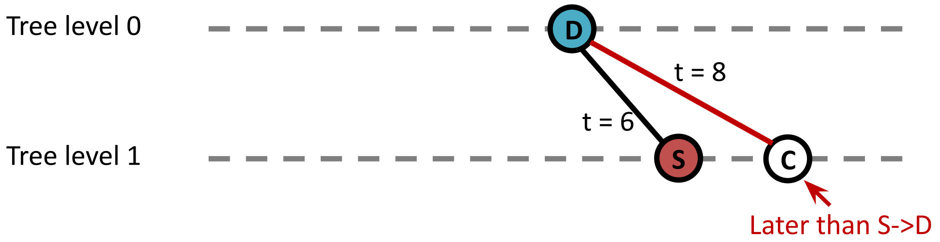
Time Slot	Circuits
1	S – H
2	A – F
3	H – E, F – A, S – B
4	S – G, A – D
5	B – D, G – B
6	S – D
7	S – A
8	C – D



HOHO Routing

Send a packet from S to D in the minimal time within max hops, max = 3 here

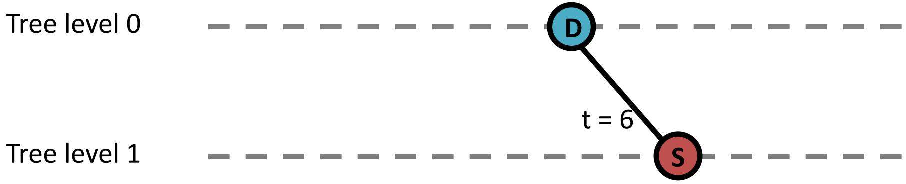
Time Slot	Circuits
1	S – H
2	A – F
3	H – E, F – A, S – B
4	S – G, A – D
5	B – D, G – B
6	S – D
7	S – A
8	C – D



HOHO Routing

Send a packet from S to D in the minimal time within max hops, max = 3 here

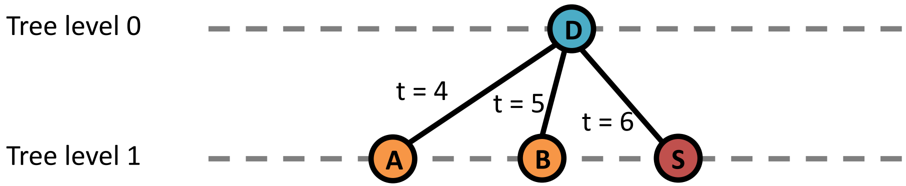
Time Slot	Circuits
1	S – H
2	A – F
3	H – E, F – A, S – B
4	S – G, A – D
5	B – D, G – B
6	S – D
7	S – A
8	C – D



HOHO Routing

Send a packet from S to D in the minimal time within max hops, max = 3 here

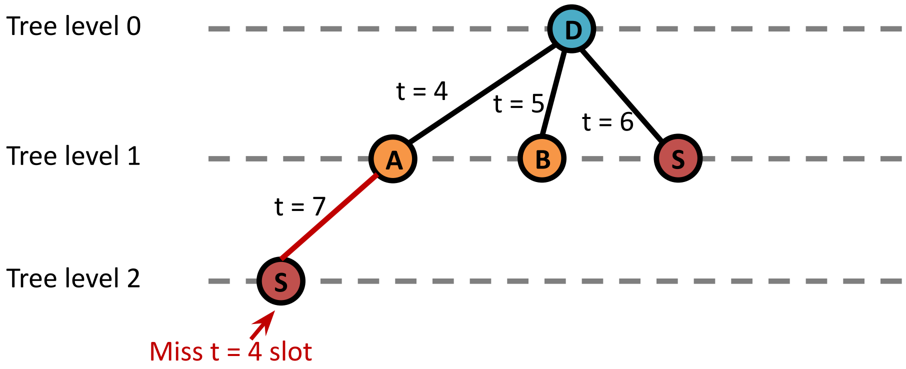
Time Slot	Circuits
1	S – H
2	A – F
3	H – E, F – A, S – B
4	S – G, A – D
5	B – D, G – B
6	S – D
7	S – A
8	C – D



HOHO Routing

Send a packet from S to D in the minimal time within max hops, max = 3 here

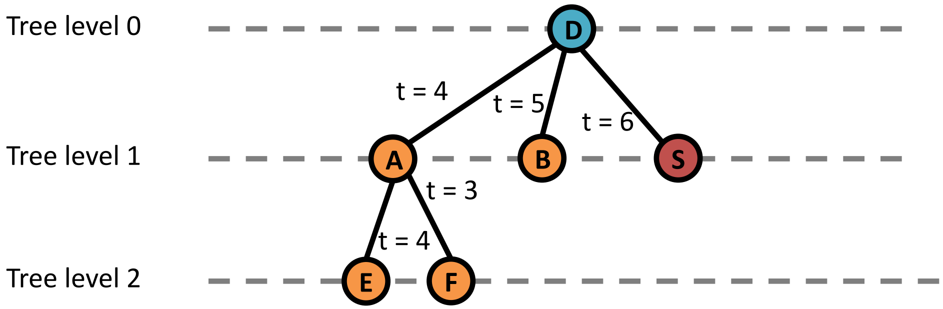
Time Slot	Circuits
1	S – H
2	A – F
3	H – E, F – A, S – B
4	S – G, A – D
5	B – D, G – B
6	S – D
7	S – A
8	C – D



HOHO Routing

Send a packet from S to D in the minimal time within max hops, max = 3 here

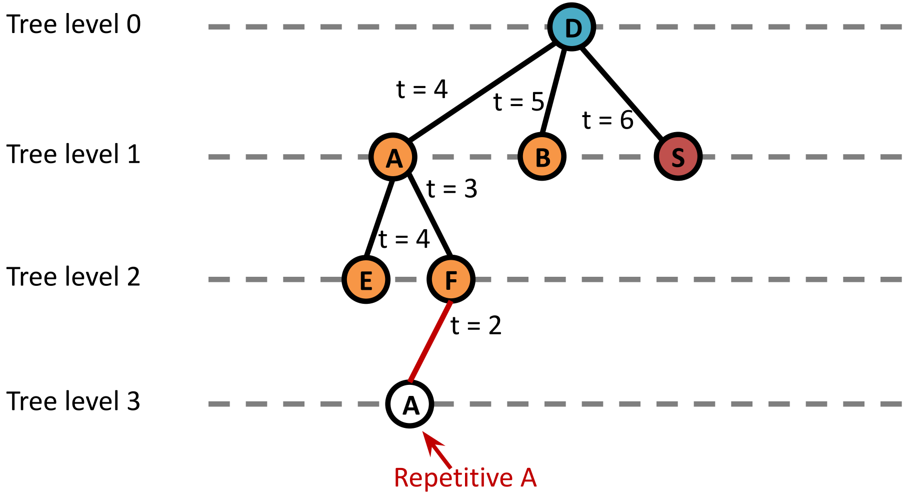
Time Slot	Circuits
1	S – H
2	A – F
3	H – E, F – A, S – B
4	S – G, A – D
5	B – D, G – B
6	S – D
7	S – A
8	C – D



HOHO Routing

Send a packet from S to D in the minimal time within max hops, max = 3 here

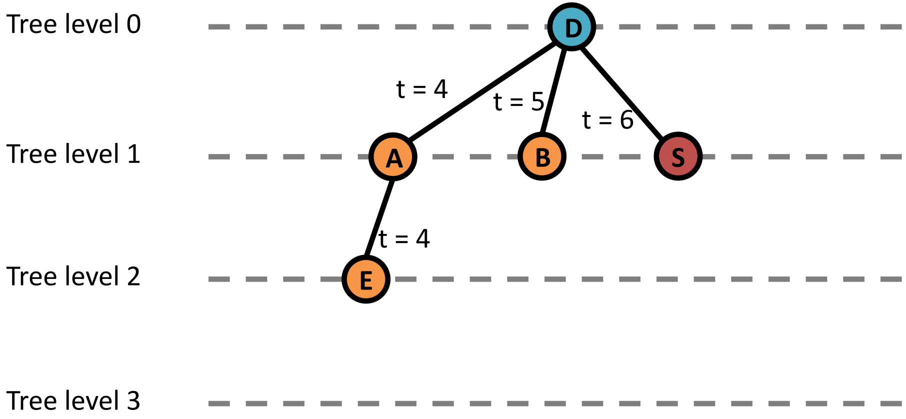
Time Slot	Circuits
1	S – H
2	A – F
3	H – E, F – A, S – B
4	S – G, A – D
5	B – D, G – B
6	S – D
7	S – A
8	C – D



HOHO Routing

Send a packet from S to D in the minimal time within max hops, max = 3 here

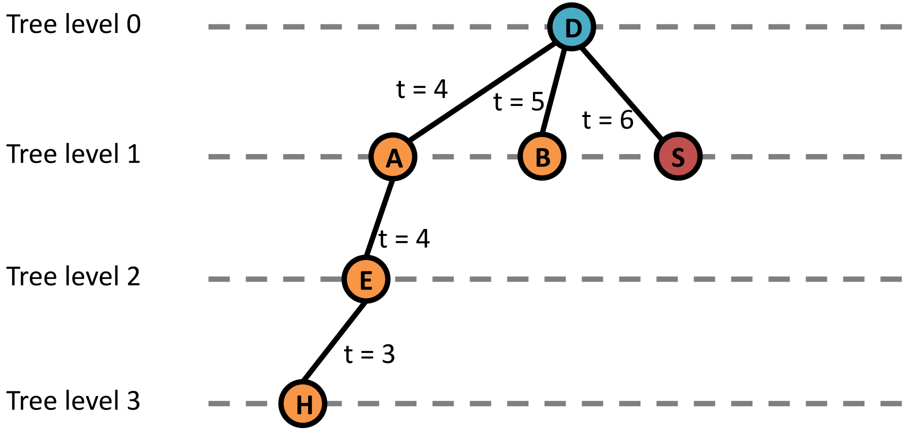
Time Slot	Circuits
1	S – H
2	A – F
3	H – E, F – A, S – B
4	S – G, A – D
5	B – D, G – B
6	S – D
7	S – A
8	C – D



HOHO Routing

Send a packet from S to D in the minimal time within max hops, max = 3 here

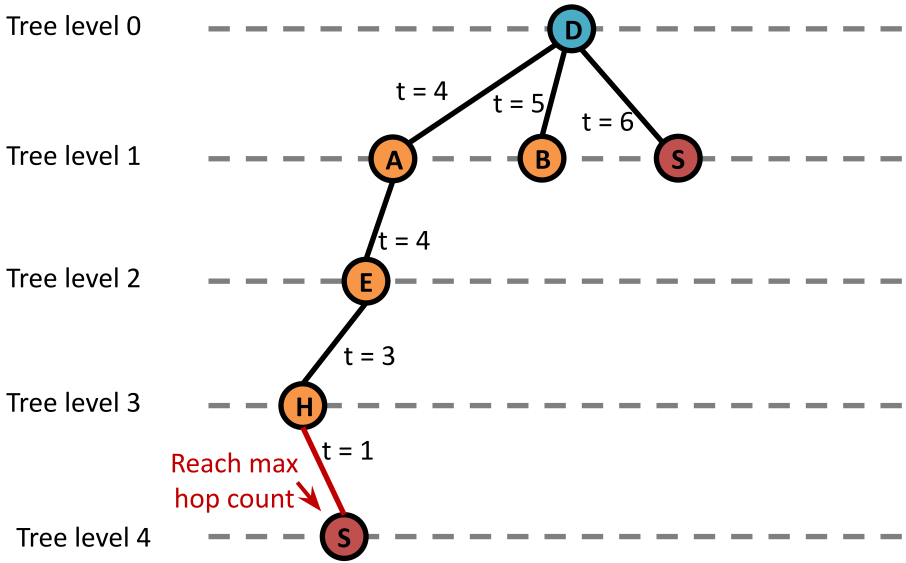
Time Slot	Circuits
1	S – H
2	A – F
3	H – E, F – A, S – B
4	S – G, A – D
5	B – D, G – B
6	S – D
7	S – A
8	C – D



HOHO Routing

Send a packet from S to D in the minimal time within max hops, max = 3 here

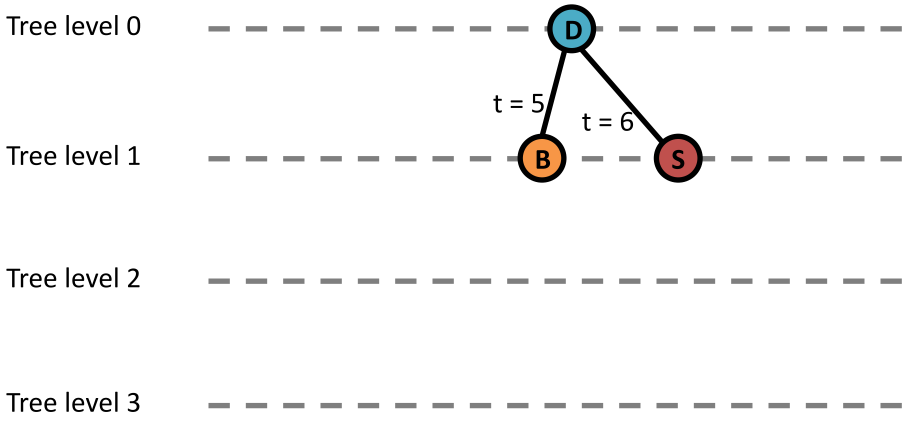
Time Slot	Circuits
1	S – H
2	A – F
3	H – E, F – A, S – B
4	S – G, A – D
5	B – D, G – B
6	S – D
7	S – A
8	C – D



HOHO Routing

Send a packet from S to D in the minimal time within max hops, max = 3 here

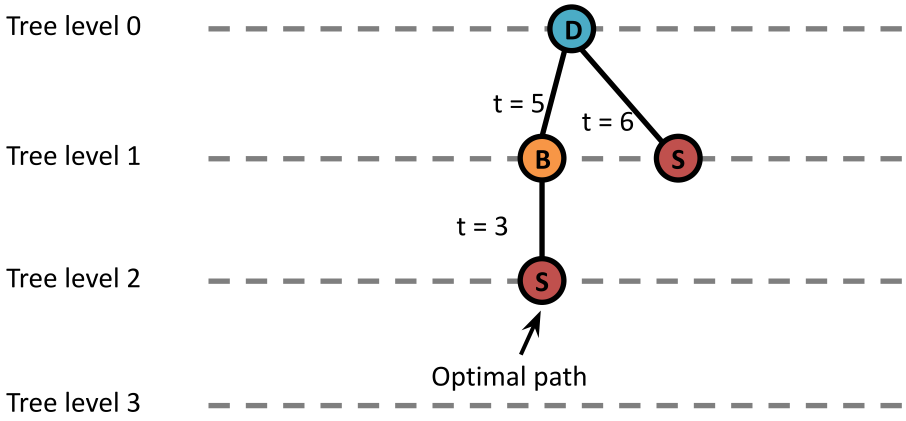
Time Slot	Circuits
1	S – H
2	A – F
3	H – E, F – A, S – B
4	S – G, A – D
5	B – D, G – B
6	S – D
7	S – A
8	C – D



HOHO Routing

Send a packet from S to D in the minimal time within max hops, max = 3 here

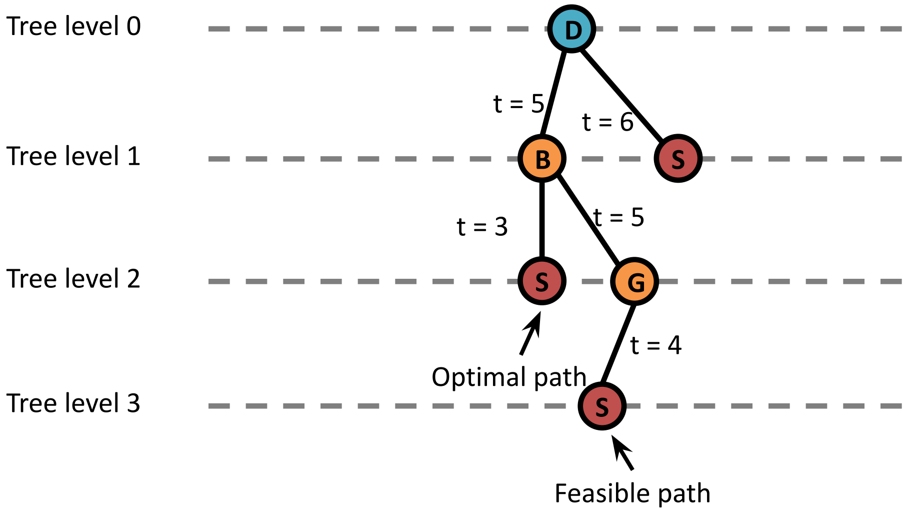
Time Slot	Circuits
1	S – H
2	A – F
3	H – E, F – A, S – B
4	S – G, A – D
5	B – D, G – B
6	S – D
7	S – A
8	C – D



HOHO Routing

Send a packet from S to D in the minimal time within max hops, max = 3 here

Time Slot	Circuits
1	S – H
2	A – F
3	H – E, F – A, S – B
4	S – G, A – D
5	B – D, G – B
6	S – D
7	S – A
8	C – D

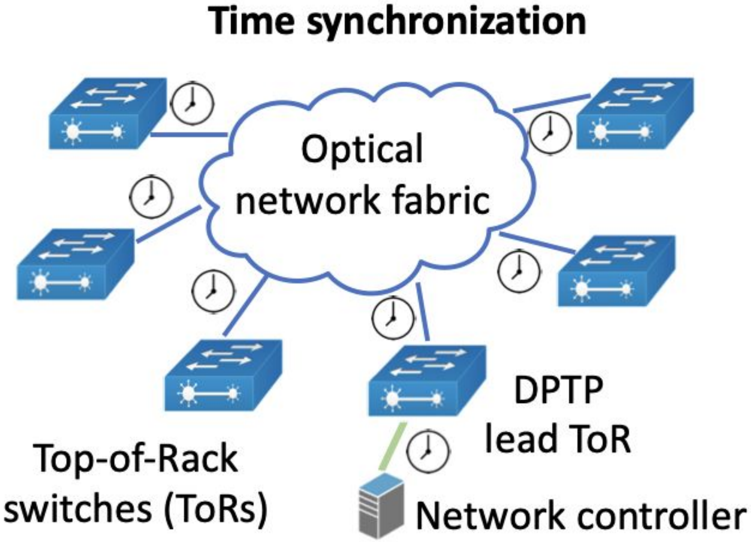


HOHO Routing: Features

- Backtracking
    - Search from dst to src ToR
- Optimal
    - Generate the fastest and shortest path
- Offline
    - Calculate offline and build routing table
- Decouple algorithm and run-time system designs
    - Assume no queuing delay at offline calculation and consider this at run-time system

System Design: Time Synchronization

- Every ToR needs to know when to send packet
- Leverage existing protocols
  - DPTP[1], nanosecond-level synchronization precision



[1] Precise time-synchronization in the data-plane using programmable switching ASICs, SOSR' 19

System Design: Routing Lookup

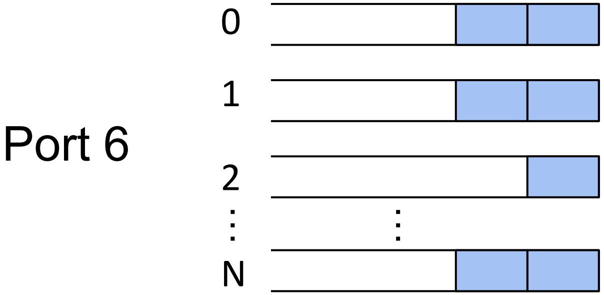
Current Time Slot	Dst ToR	Egress Port	Send Time Slot
0	1	6	0
0	2	23	4
1	1	6	2
…			
7	0	12	10
…			

- HOHO generates a routing table
  - Current time slot: when the packet arrives
  - Dst ToR: final destination ToR
  - Egress port
  - Send time slot: when to send out the packet

System Design: Routing Lookup

Current Time Slot	Dst ToR	Egress Port	Send Time Slot
0	1	6	0
0	2	23	4
1	1	6	2
…			
7	0	12	10
…			

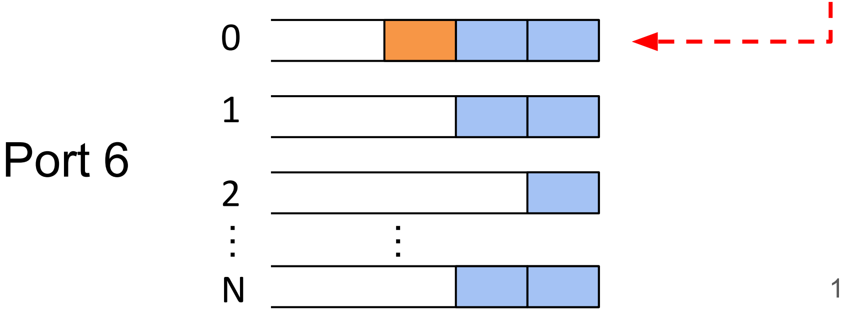
- Look up current time slot and dst ToR to get egress port and send time slot
    - Packet with current time slot 0 and dst ToR 1



System Design: Routing Lookup

Current Time Slot	Dst ToR	Egress Port	Send Time Slot
0	1	6	0
0	2	23	4
1	1	6	2
…			
7	0	12	10
…			

- Look up current time slot and dst ToR to get egress port and send time slot
    - Packet with current time slot 0 and dst ToR 1
    - Buffer at queue 0

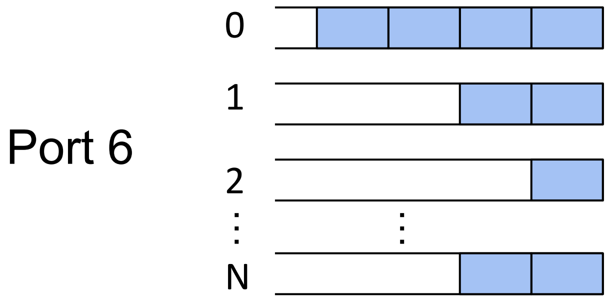


System Design: Routing Lookup

- If missing the planned send time slot due to the queue occupancy, look up from next current time slot

Current Time Slot	Dst ToR	Egress Port	Send Time Slot
0	1	6	0
0	2	23	4
1	1	6	2
…			
7	0	12	10
…			

Queue 0 is full

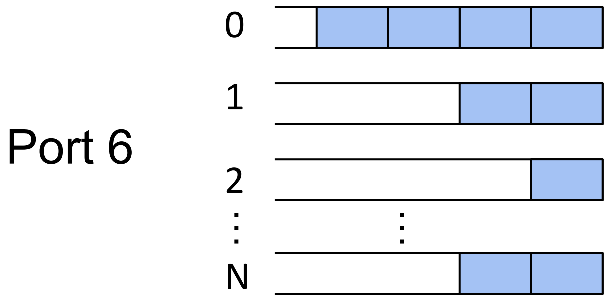


System Design: Routing Lookup

Current Time Slot	Dst ToR	Egress Port	Send Time Slot
0	1	6	0
0	2	23	4
1	1	6	2
…			
7	0	12	10
…			

- If missing the planned send time slot due to the queue occupancy, look up from next current time slot
    - Move to current time slot 1

Queue 0 is full

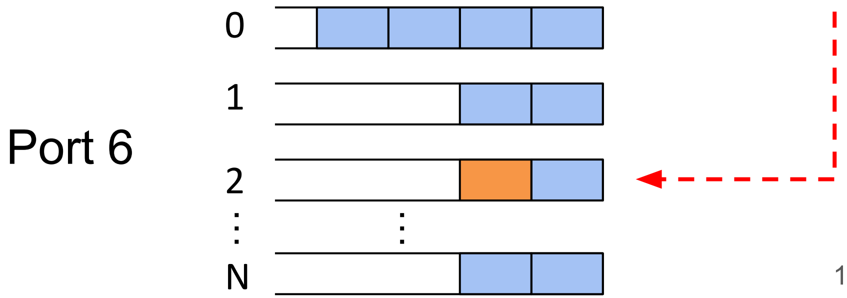


System Design: Routing Lookup

- If missing the planned send time slot due to the queue occupancy, look up from next current time slot
  - Move to current time slot 1
  - Buffer at queue 2

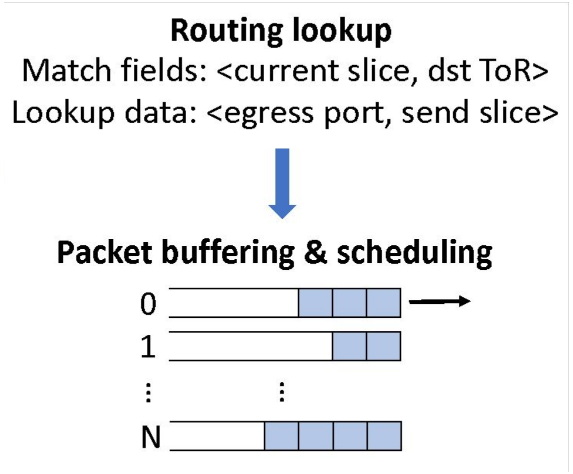
Current Time Slot	Dst ToR	Egress Port	Send Time Slot
0	1	6	0
0	2	23	4
1	1	6	2
...			
7	0	12	10
...			

Queue 0 is full



System Design: Buffering

- Packet is buffered at the queue before its sending time slot
- Could be realized by queue pause


Routing lookup
Match fields: <current slice, dst ToR>
Lookup data: <egress port, send slice>
Packet buffering & scheduling
0
1
⋮
N

Simulation

Setup

- Reused the setup in Opera paper[1]
    - Topology: 108 ToRs and 648 servers, each ToR with six 10G downlinks to servers and six 10G uplinks to optical fabric
    - Workload: 1%~40% data-mining traffic from Microsoft

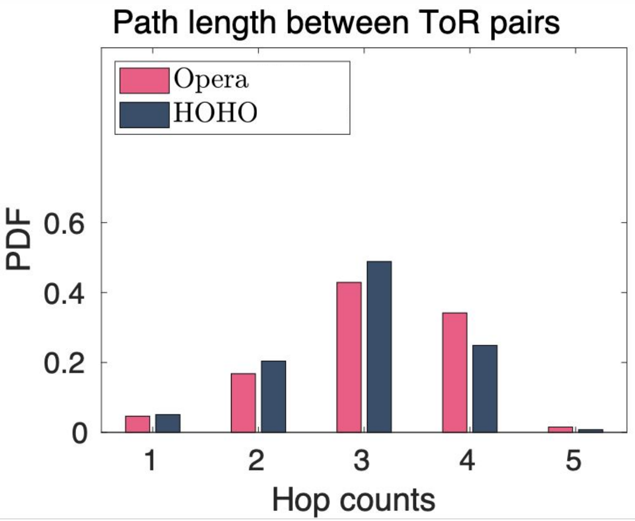
[1] Expanding across time to deliver bandwidth efficiency and low latency, NSDI' 20

Simulation

Setup

- Reused the setup in Opera paper[1]
- Topology: 108 ToRs and 648 servers, each ToR with six 10G downlinks to servers and six 10G uplinks to optical fabric
- Workload: 1%~40% data-mining traffic from Microsoft

Shorter paths

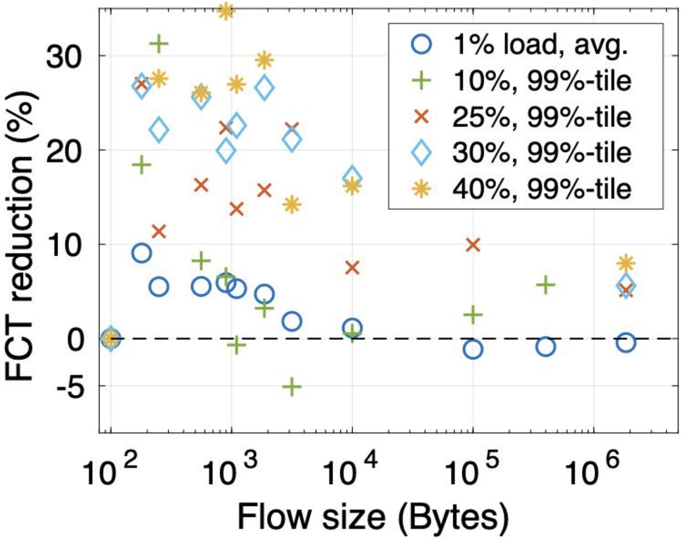


- Avg. hops: 3.11 => 2.80
- >= 4 hops: 37% => 25%

[1] Expanding across time to deliver bandwidth efficiency and low latency, NSDI' 20

Simulation

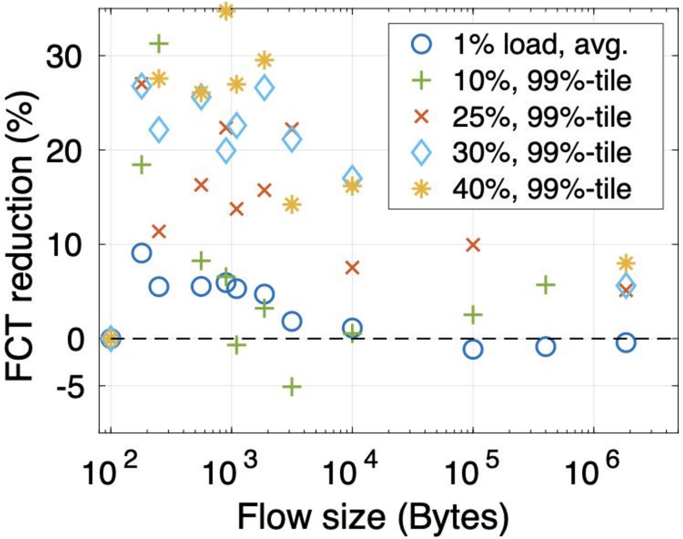
Lower latency



- FCT reduction: up to 35%

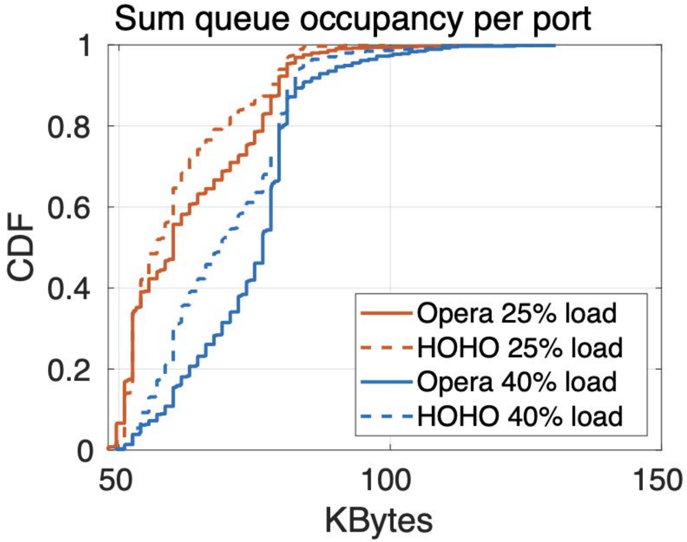
Simulation

Lower latency



- FCT reduction: up to 35%

Lower queue occupancy



- Queue occupancy reduction: 5% (25% load) and 10% (40% load)

Summary

- Hop-On Hop-Off optical circuits
    - Allow packets to “wait” at ToRs
- HOHO routing algorithm
    - Works on any optical schedule
    - Optimal (fastest and shortest path)
- A system sketch
    - Time synchronization + routing lookup + buffer management
- Simulation
    - Shorter paths, lower FCT, lower queue occupancy

THANKS!

city circle
Hop on hop off
Gray Line
SIGHTSEEING EVERYWHERE
BUY BERLIN TICKETS
MULTI-LANGUAGE