EuroSys 2026

Artifacts Available V1.1 · Artifacts Evaluated Functional V1.1 · Results Reproduced V1.1

LCMP: Distributed Long-Haul Cost-Aware Multi-Path Routing for Inter-Datacenter RDMA Networks

Dong-Yang Yu[1], Yuchao Zhang[1#], Xiaodi Wang[1], Jun Wang[1],

Wenfei Wu[2], Haipeng Yao[1], Wendong Wang[1], Ke Xu[3]

1 北京郵電大學 Beijing University of Posts and Telecommunications

2 北京大學 PEKING UNIVERSITY

3 清華大學 Tsinghua University

Edinburgh

April 27-30, 2026

Background

AI workloads are scaling fast, and infrastructure is moving from one datacenter to multiple datacenters.

Text generation — ChatGPT

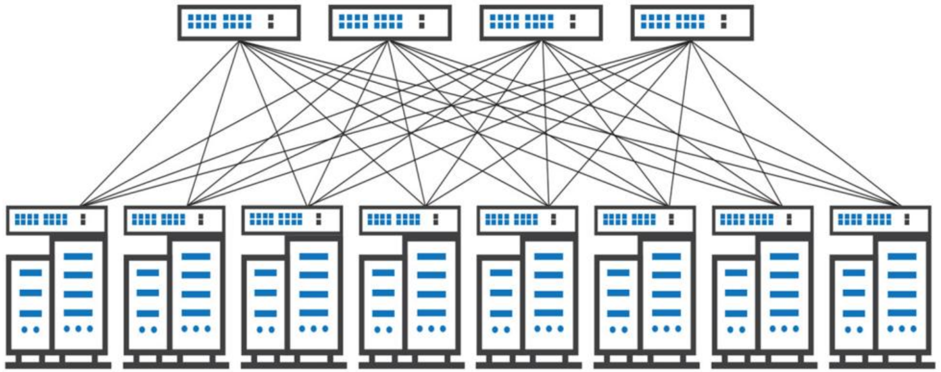
Video generation — ByteDance Seedance 2.0

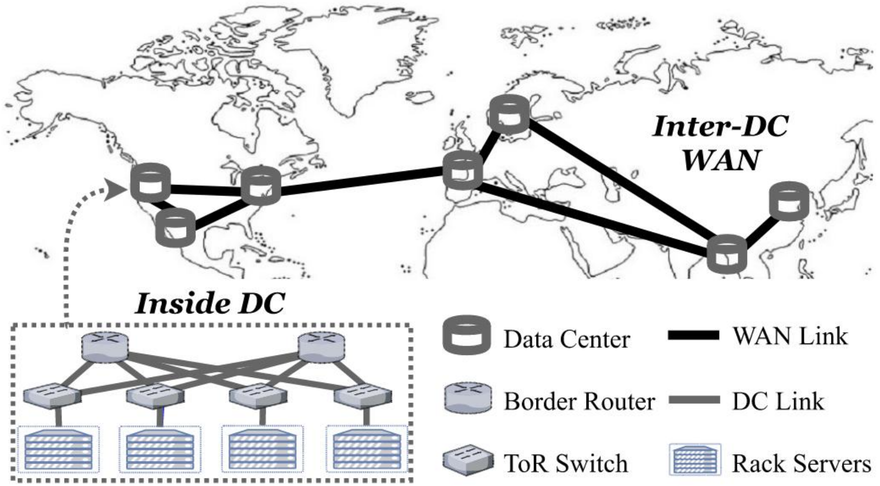
Code generation — Claude

Single-DC deployment → **Multi-DC deployment**

AI demand ↑ Power constraints

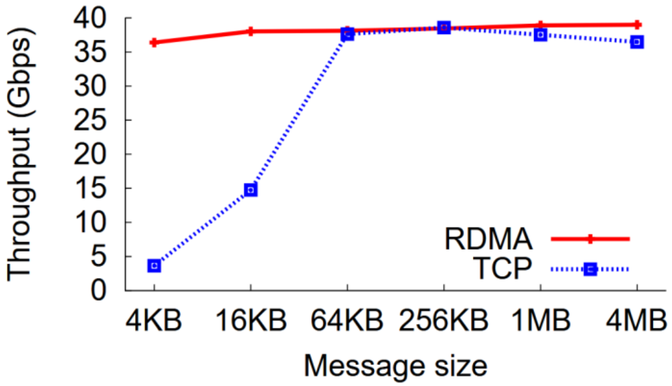
Space, cooling limits



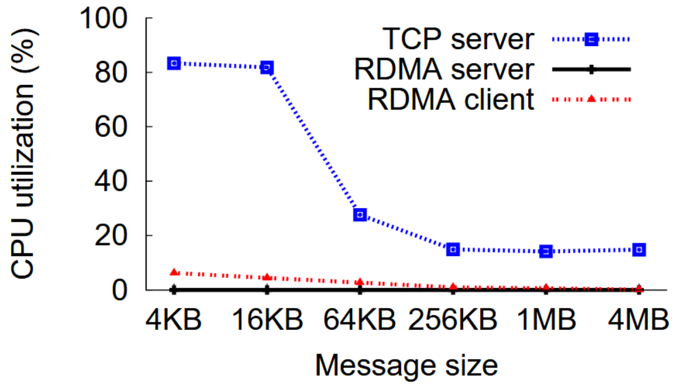


The Golden Standard: RDMA inside Datacenters

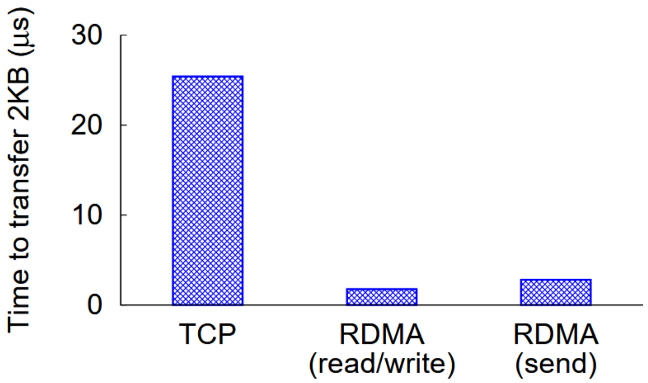
RDMA outperforms TCP (From DCQCN, SIGCOMM '15)



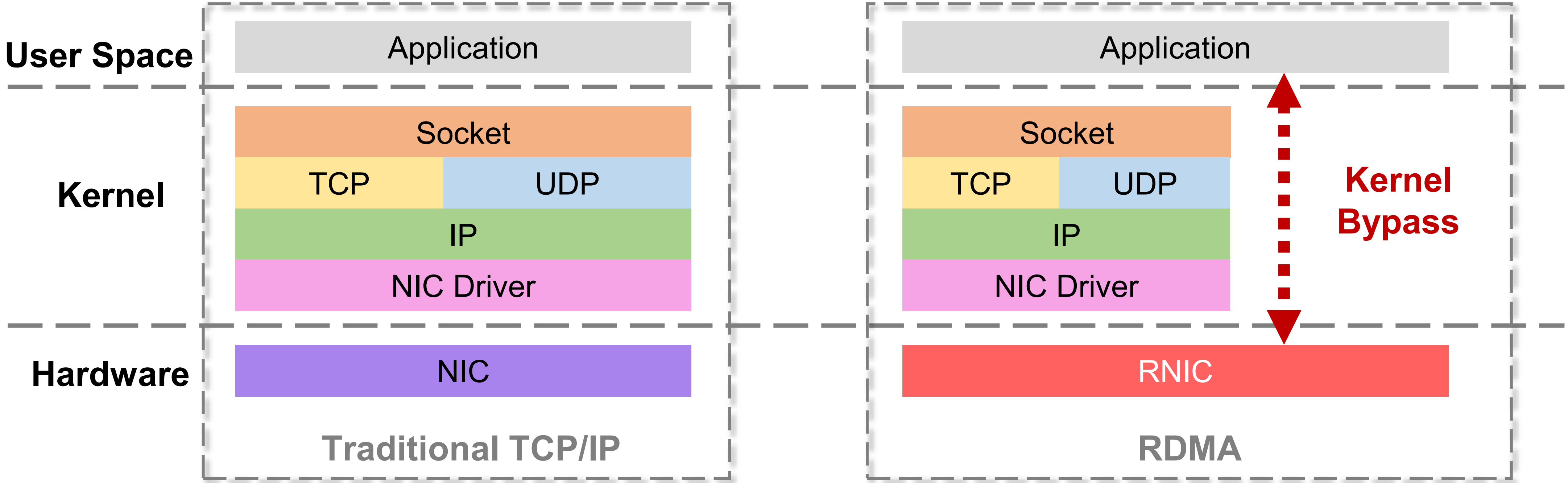
RDMA drives full throughput



RDMA CPU ~0%

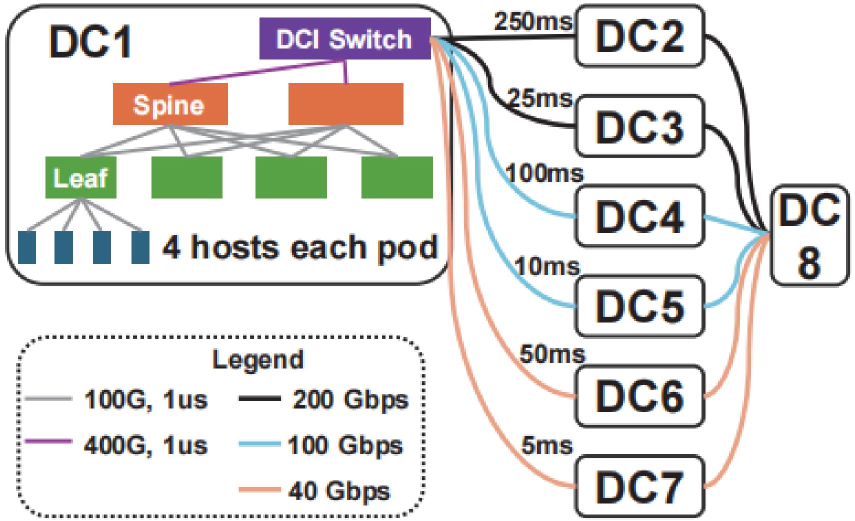


RDMA latency 1~2 us

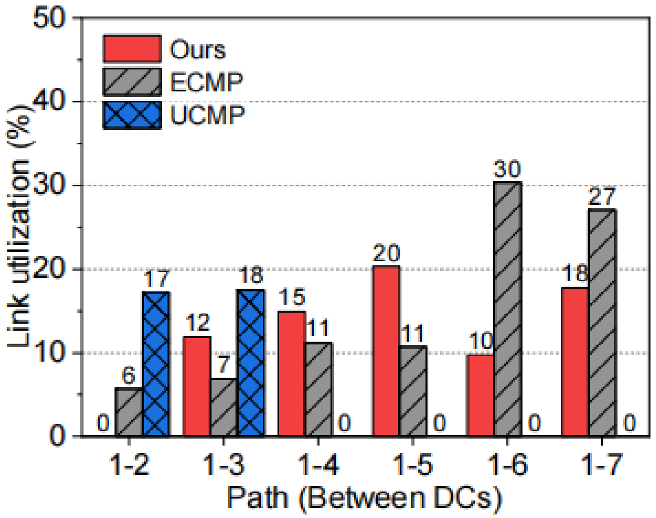


Motivation

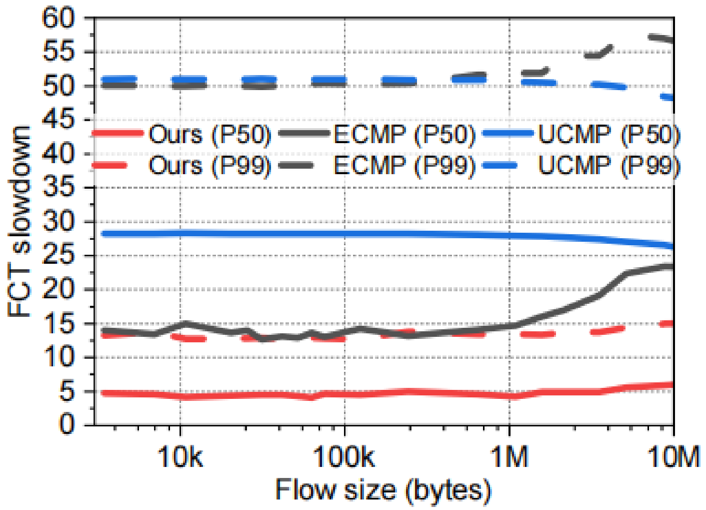
RDMA Fails in Inter-DC



(a) Inter 8-DC topology[1].



(b) Per-link utilization.



(c) Median and tail FCT slowdown Median and tail FCT slowdown for Web Search under 30% load using DCQCN.

Tradition: ECMP Problem

Random hashing ignores capacity or delay asymmetry

Can choose **low-bw links**:

- DC1→DC6: 30% util
- DC1→DC7: 27% util

SOTA: UCMP Problem [SIGCOMM 24]

Capacity-centric concentrates on high-cap paths

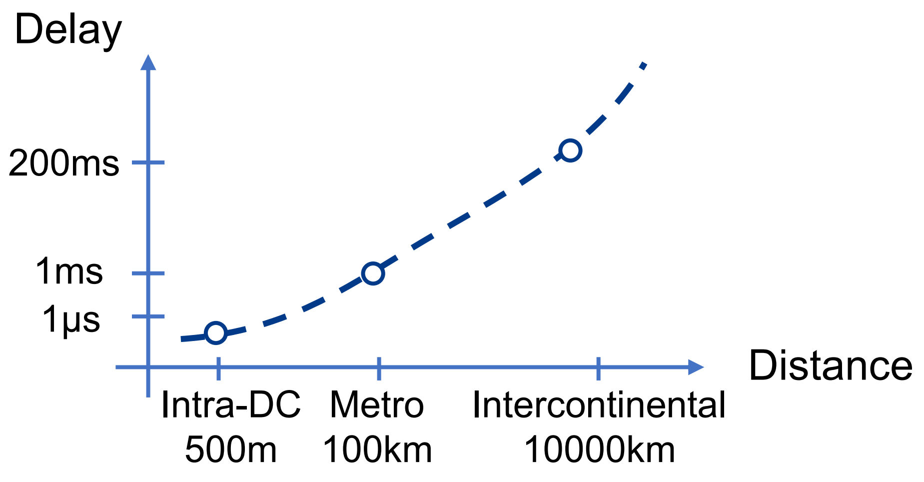
DC1→DC2 shows **17% util** (vs 6% in ECMP), leaving lower-delay capacity **underused (0%)**

Direct Impact on FCT

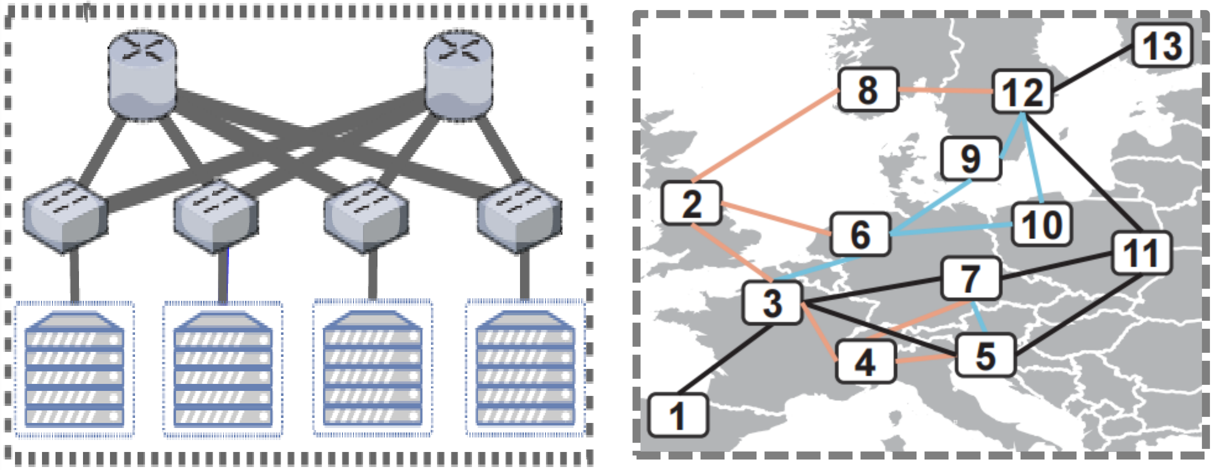
Median FCT Slowdown

Tail FCT Slowdown

Why Existing Routing Methods Fail in Inter-DC Networks



Propagation Delay	**~1µs**	**1-100ms**
One-way latency	Intra-DC	Inter-DC

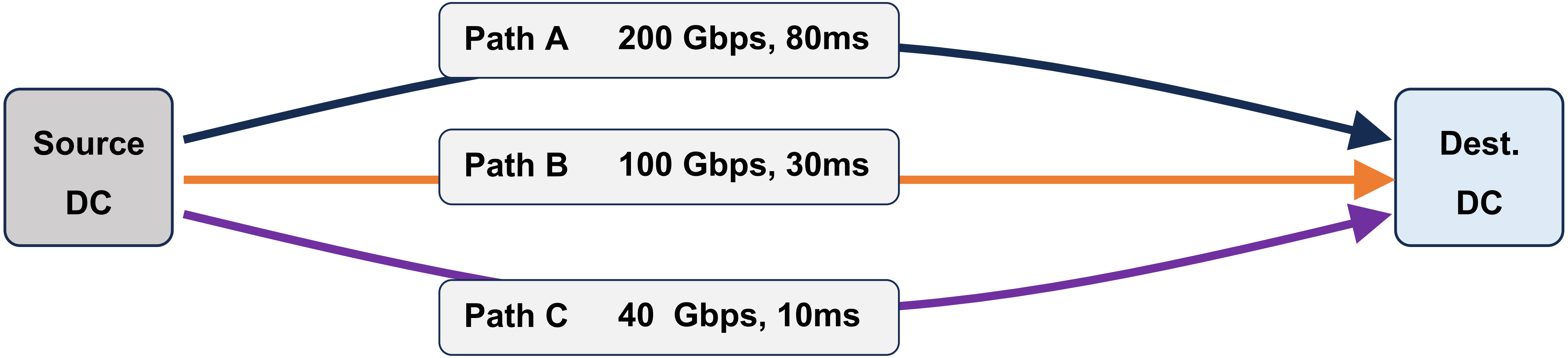


Topology	**Symmetric**	**Sparse**
Network structure	Intra-DC	Inter-DC

Routing **in the inter-DC setting** **cannot** **simply follow the assumptions we used inside a single datacenter.**

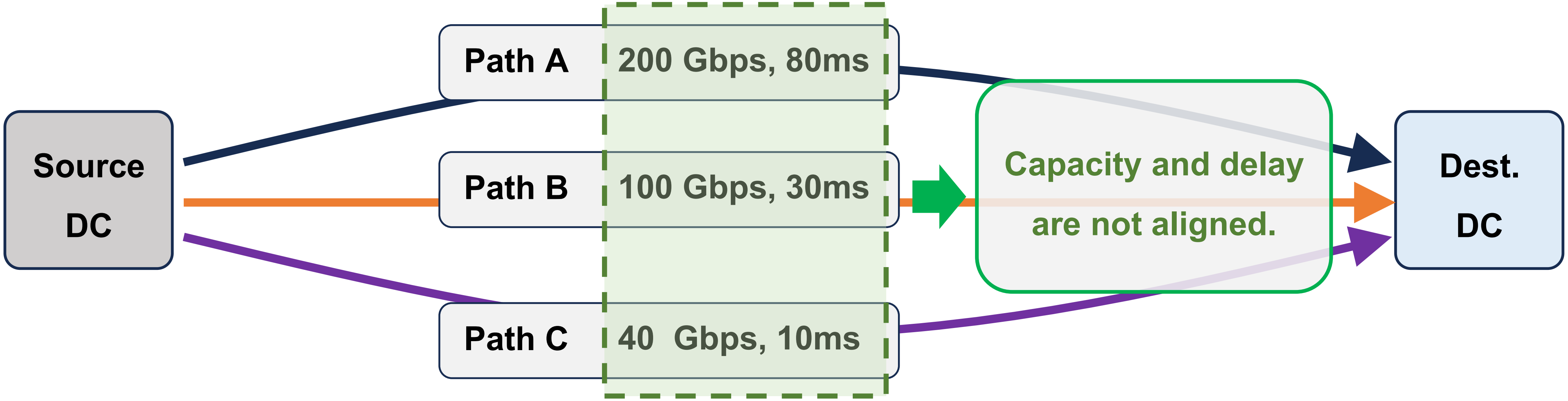
Challenges

➡️C1 Path asymmetry: delay and capacity are misaligned

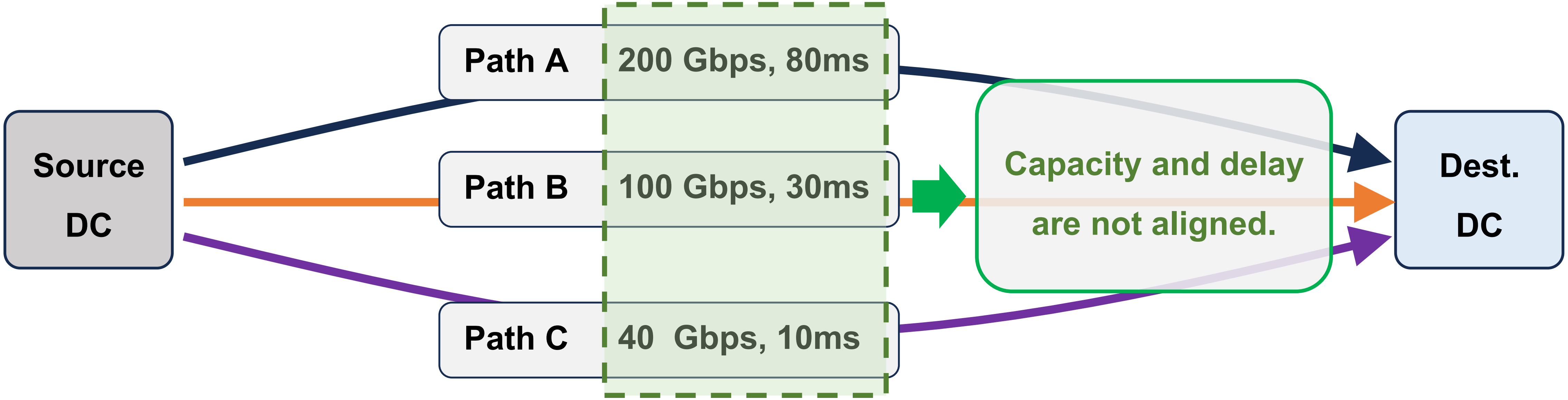


Challenges

C1 Path asymmetry: delay and capacity are misaligned



➡️ C1 Path asymmetry: delay and capacity are misaligned



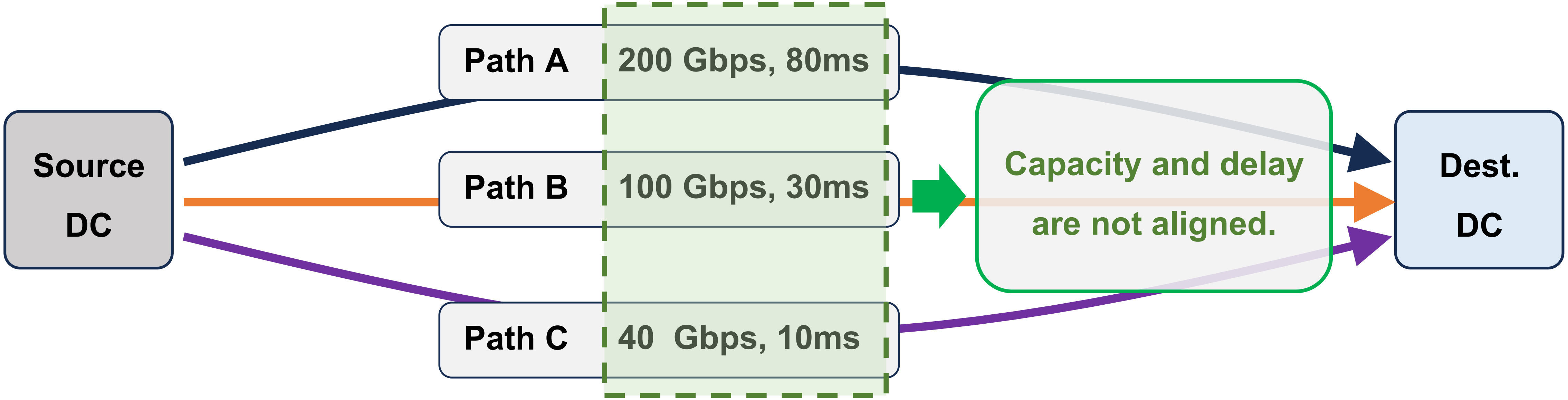
Capacity-only ranking

Picks Path A since 200 Gbps ✅
But the worst propagation delay ❌

Latency-sensitive RDMA flows ❌

Challenges

C1 Path asymmetry: delay and capacity are misaligned



Capacity-only ranking

Picks Path A since 200 Gbps ✅
But the worst propagation delay ❌

Latency-sensitive RDMA flows ❌

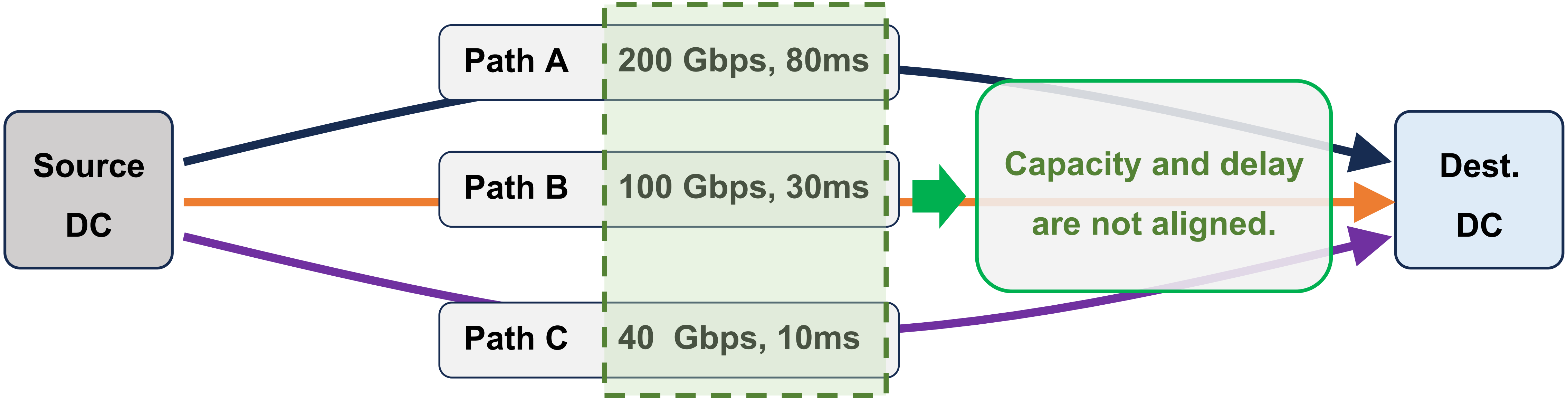
Delay-only ranking

Picks Path C since 10 ms ✅
But the weakest bandwidth ❌

Throughput-sensitive RDMA flows ❌

Challenges

➡️ C1 Path asymmetry: delay and capacity are misaligned



Capacity-only ranking

Picks Path A since 200 Gbps ✅
But the worst propagation delay ❌

Latency-sensitive RDMA flows ❌

Delay-only ranking

Picks Path C since 10 ms ✅
But the weakest bandwidth ❌

Throughput-sensitive RDMA flows ❌

Need a unified path-quality score that jointly captures delay and capacity.

Challenges

➡️ C2 Slow & Outdated Congestion Signals

Timeline

t0 t1 t2 t3

One DCI switch scenario

DC1
New flow starts

DCI switch
makes per-flow routing decision

Inter-DC path
queue can grow during transmission

Challenges

➡️ C2 Slow & Outdated Congestion Signals

Timeline

new flow

t0 t1 t2 t3

One DCI switch scenario

DC1
New flow starts

Forward traffic

DCI switch
makes per-flow routing decision

Inter-DC path
queue can grow during transmission

C2 Slow & Outdated Congestion Signals

Timeline

- t0 — new flow
- t1 — switch decides
- t2
- t3

One DCI switch scenario

DC1
New flow starts

→ Forward traffic →

DCI switch
makes per-flow routing decision

sees only current state

→ Long-haul propagation →

Inter-DC path
queue can grow during transmission

t1: decision

Switch routes a flow using latest state.
Now, congestion **is not obvious yet**.

Challenges

➡️ C2 Slow & Outdated Congestion Signals

Timeline

new flow — switch decides — queue already grows

t0 — t1 — t2 — t3

One DCI switch scenario

DC1
New flow starts

Forward traffic

DCI switch
makes per-flow routing decision

sees only current state

Long-haul propagation

Inter-DC path
queue can grow during transmission

queue builds up

t1: decision

Switch routes a flow using latest state.
Now, congestion **is not obvious yet**.

t2: queue growth

More packets arrive on the path.
Queue **starts growing.**

Challenges

➡ C2 Slow & Outdated Congestion Signals

Timeline

- t0 — new flow
- t1 — switch decides
- t2 — queue already grows
- t3 — feedback finally returns

One DCI switch scenario

DC1
New flow starts

Forward traffic

DCI switch
makes per-flow routing decision

sees only current state

Long-haul propagation

Inter-DC path
queue can grow during transmission

queue builds up

delayed ECN, telemetry, congestion signal

t1: decision

Switch routes a flow using latest state.
Now, congestion **is not obvious yet**.

t2: queue growth

More packets arrive on the path.
Queue **starts growing.**

t3: outdated feedback

Signal finally arrives back.
But it reflects **the past**, not the current

Challenges

C2 Slow & Outdated Congestion Signals

Timeline

- t0 — new flow
- t1 — switch decides
- t2 — queue already grows
- t3 — feedback finally returns

One DCI switch scenario

DC1
New flow starts

Forward traffic →

DCI switch
makes per-flow routing decision
sees only current state

Long-haul propagation →

Inter-DC path
queue can grow during transmission
queue builds up

delayed ECN, telemetry, congestion signal

t1: decision	t2: queue growth	t3: outdated feedback
Switch routes a flow using latest state. Now, congestion is not obvious yet.	More packets arrive on the path. Queue starts growing.	Signal finally arrives back. But it reflects the past, not the current
Looks correct, but only from an old link state	**The network state has already changed**	**Feedback-driven routing reacts too late**

Challenges

➡ C2 Slow & Outdated Congestion Signals

Timeline
- t0 — new flow
- t1 — switch decides
- t2 — queue already grows
- t3 — feedback finally returns

One DCI switch scenario

DC1
New flow starts

Forward traffic

DCI switch
makes per-flow routing decision
sees only current state

Long-haul propagation

Inter-DC path
queue can grow during transmission
queue builds up

delayed ECN, telemetry, congestion signal

t1: decision
Looks correct, but only from an old link state
Switch routes a flow using latest state.
Now, congestion is not obvious yet.

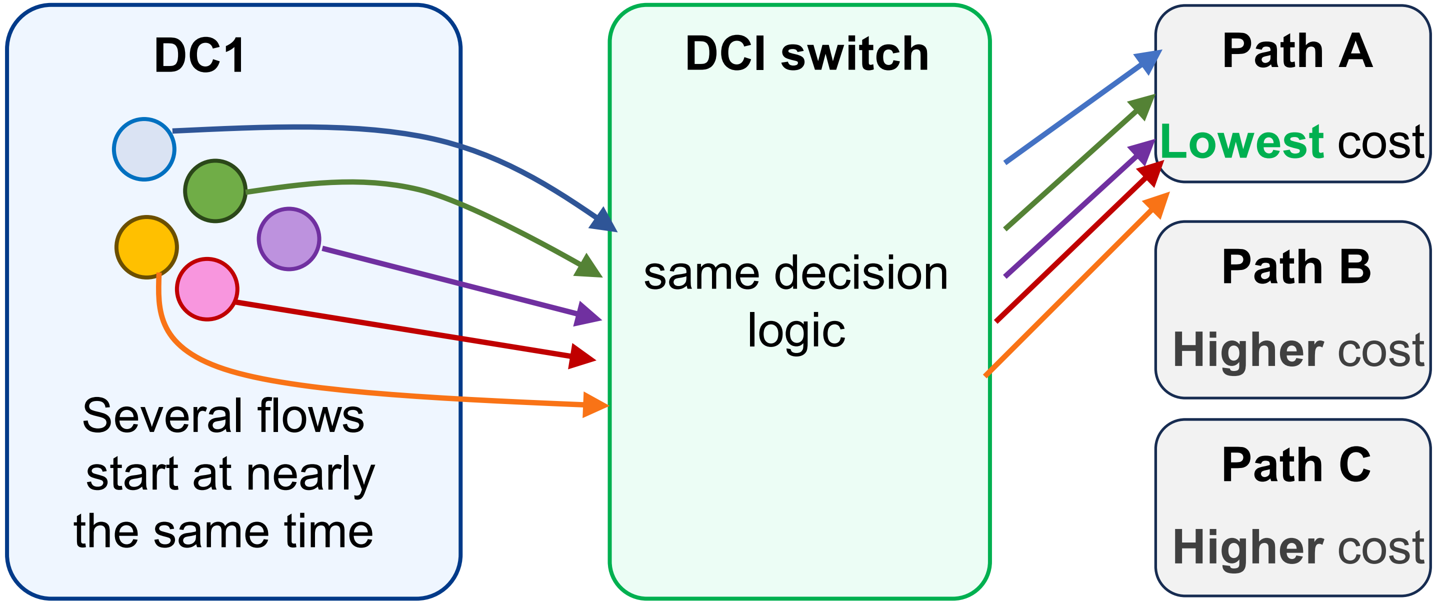
t2: queue growth
The network state has already changed
More packets arrive on the path.
Queue starts growing.

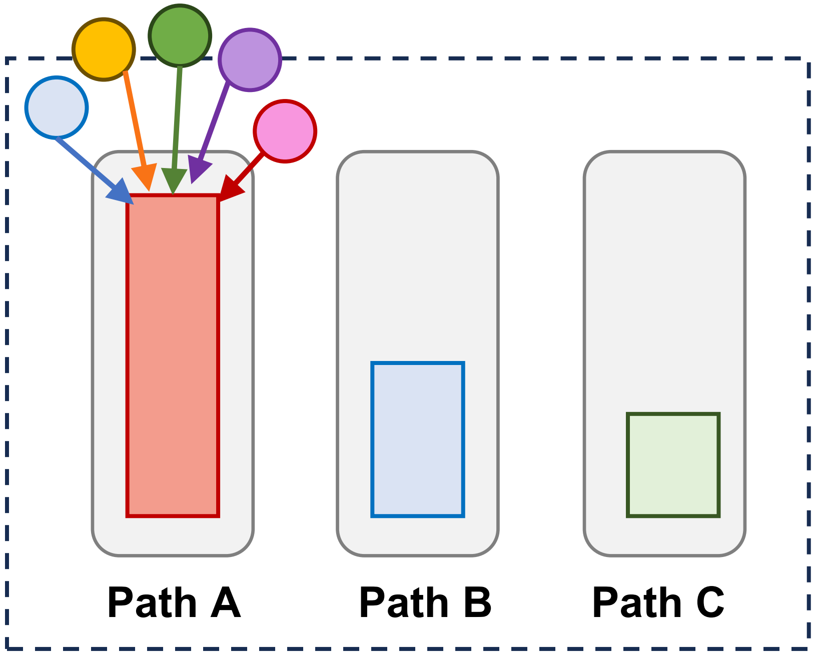
t3: outdated feedback
Feedback-driven routing reacts too late
Signal finally arrives back.
But it reflects the past, not the current

Need timely on-switch signals, instead of relying on delayed end-to-end feedback.

Challenges

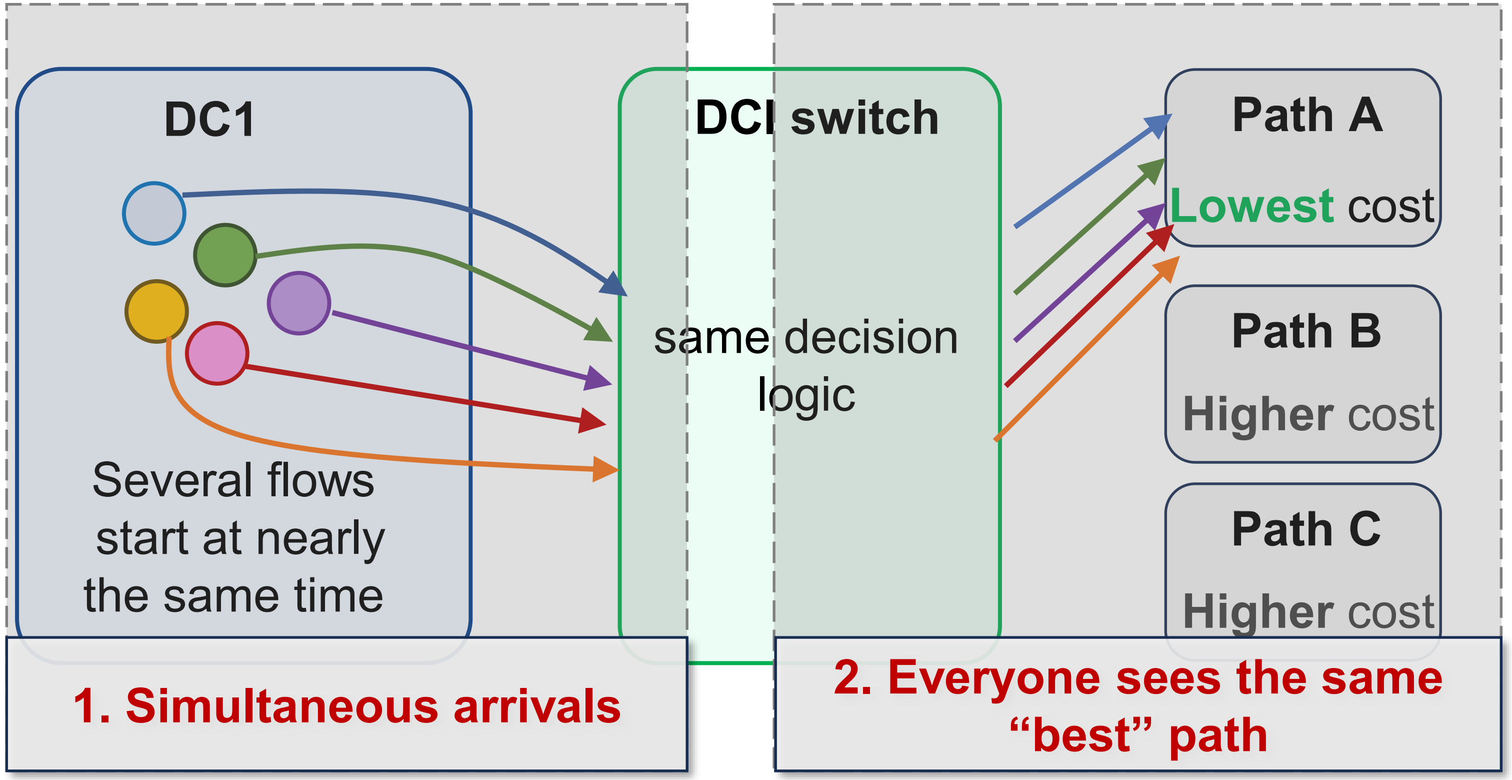
➡ C3 Simultaneous flow arrivals can create a herd effect

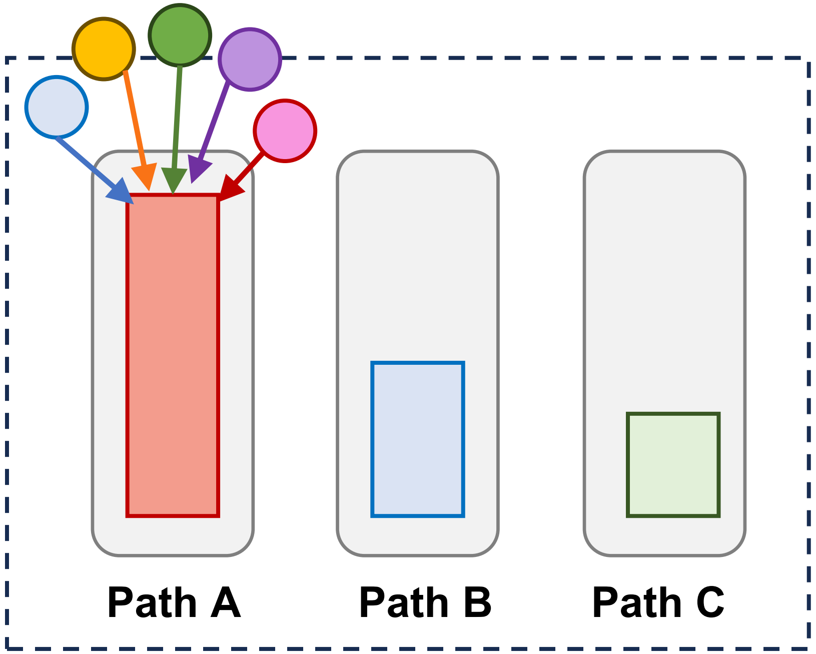




Challenges

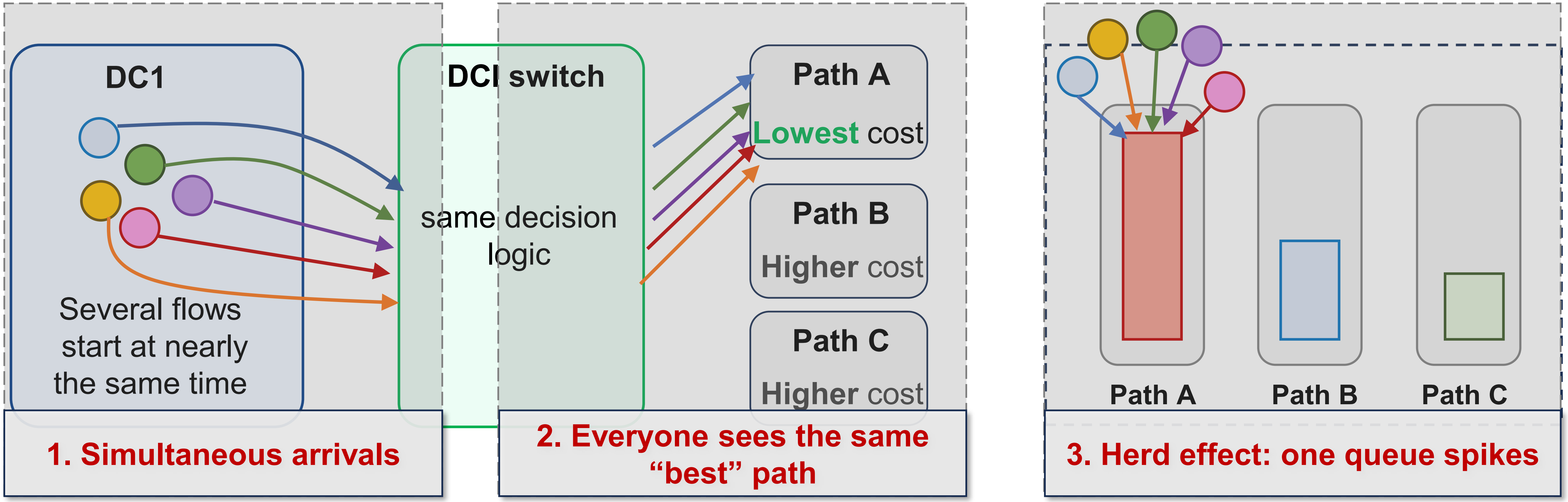
C3 Simultaneous flow arrivals can create a herd effect





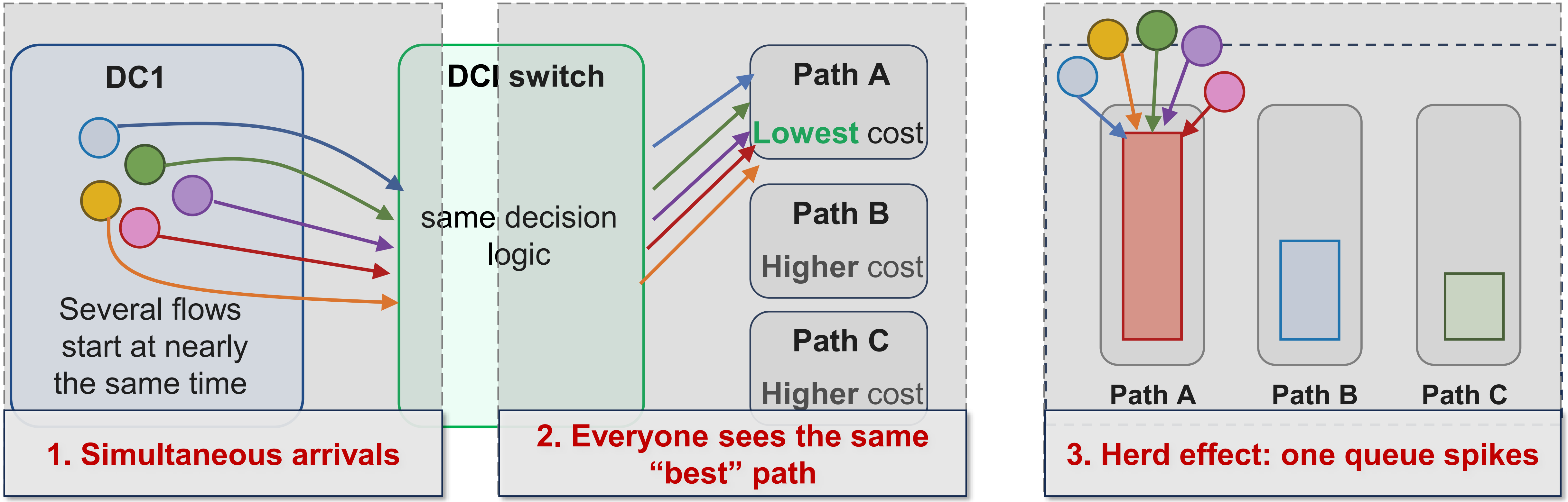
Challenges

➡️ C3 Simultaneous flow arrivals can create a herd effect



Challenges

➡ C3 Simultaneous flow arrivals can create a herd effect



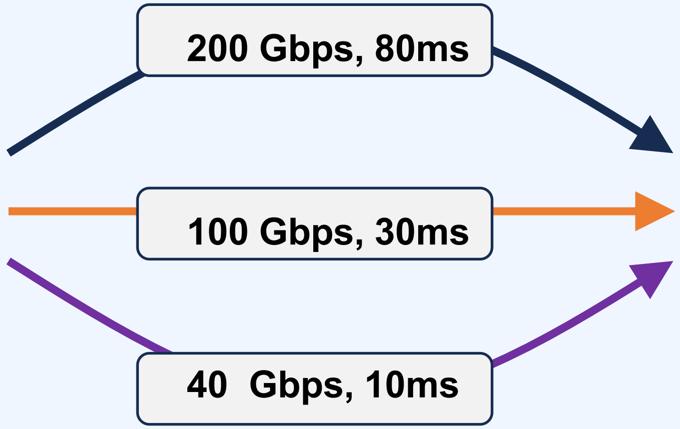
Need diversity-preserving selection:

keep several low-cost paths, then randomize within that reduced set.

Key Takeaways from the Challenges

Inter-DC routing is hard: path heterogeneity, feedback delay, and synchronized flow decisions interact together.

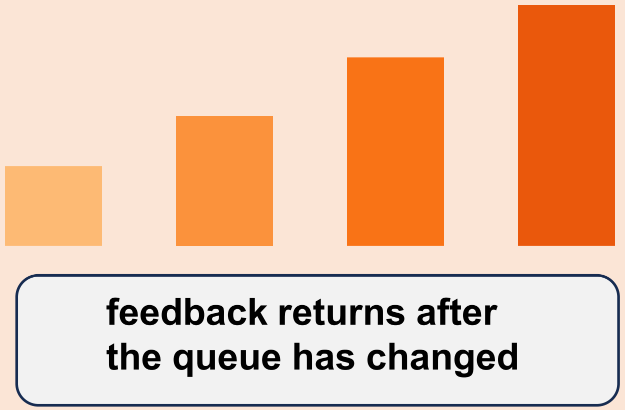
C1 Path asymmetry



Not all "good" paths are good

Delay and capacity are misaligned.
Routing cannot rely on one metric only.

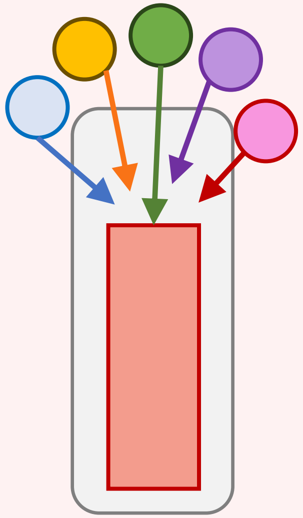
C2 Delayed feedback



Reactive routing sees the past

By the time feedback arrives,
the network state has already evolved.

C3 Herd effect



Local best can become global collision

Many new flows choose the same path,
and one queue spikes while others stay idle.

Key Takeaways from the Challenges

Inter-DC routing is hard: path heterogeneity, feedback delay, and synchronized flow decisions interact together.

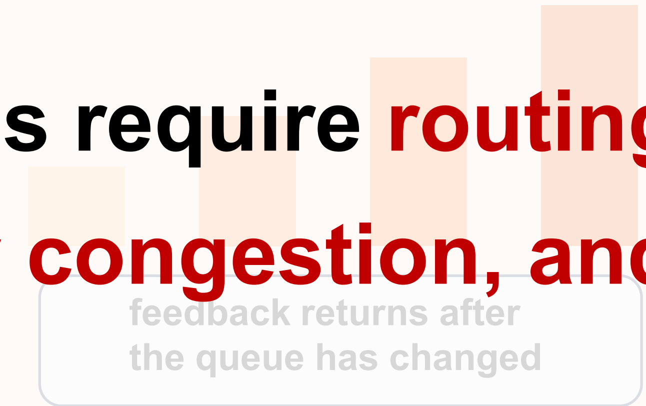
These challenges require routing to consider path quality, timely congestion, and flow diversity.

C1 Path asymmetry

Not all "good" paths are good

Delay and capacity are misaligned.
Routing cannot rely on one metric only.

C2 Delayed feedback



Reactive routing sees the past

By the time feedback arrives,
the network state has already evolved.

C3 Herd effect

Local best can become global collision

Many new flows choose the same path,
and one queue spikes while others stay idle.

LCMP Design

Architecture overview

Core Abstraction: Cost Fusion

$$C(p) = \alpha \cdot C_{path}(p) + \beta \cdot C_{cong}(p)$$

Control Plane	Data Plane
C_{path} ➡ C1 Delay + Capacity	C_{cong} ➡ C2 Queue + Trend + Duration

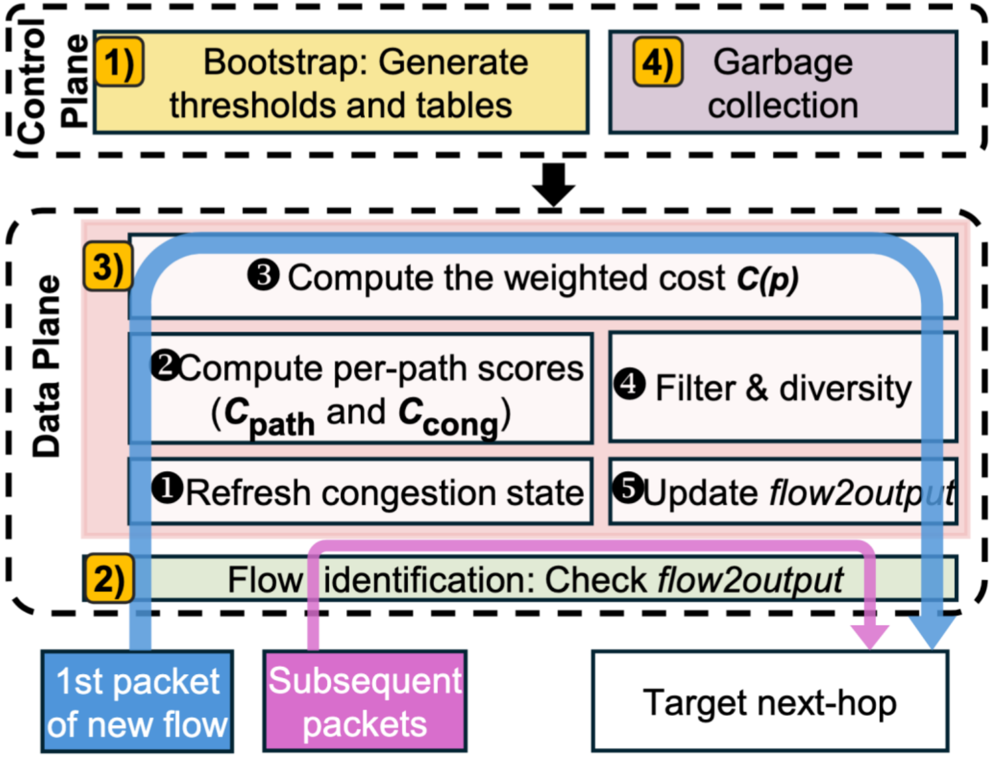
1. **DCI Switch Bootstrap**
   Install thresholds, level-score tables, C_{path} scores

2. **Flow Identification**
   Check flow2output mapping; forward if established

3. **Flow Routing (First Packet)** ➡ C3
   Refresh congestion → Compute scores → Filter & select

4. **Garbage Collection**
   Periodic eviction of idle flow entries

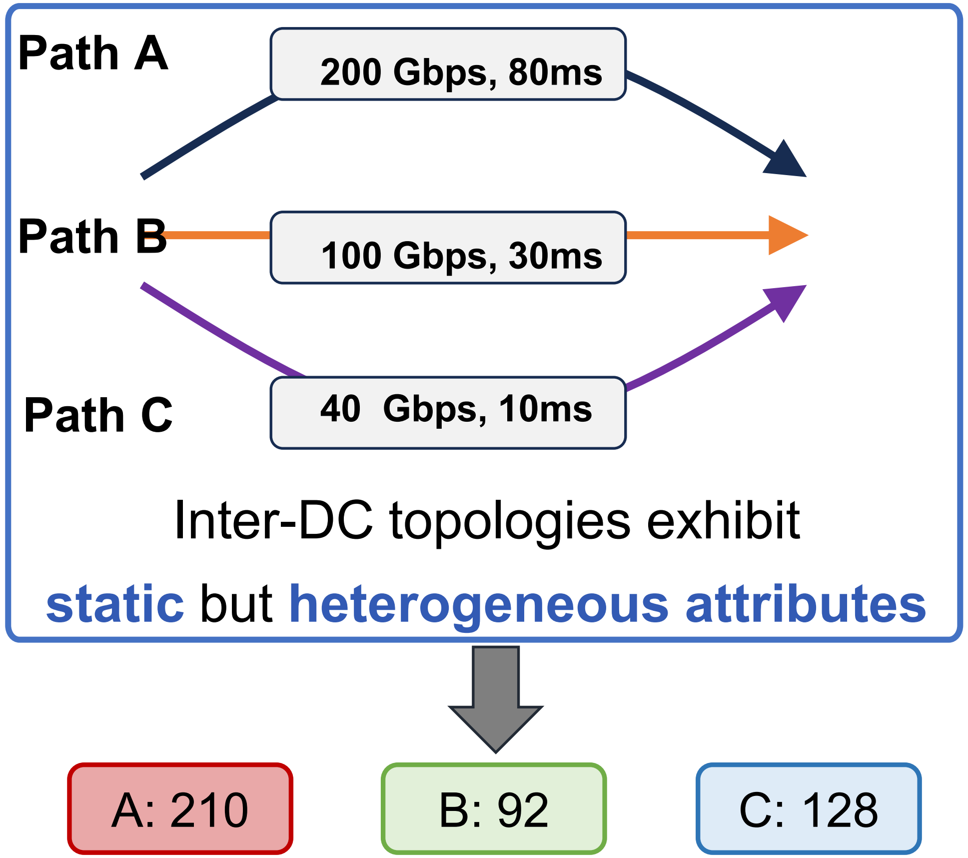
Runtime Workflow



Key Properties

- ✓ Orthogonal to end-host CC
- ✓ Line-rate decisions
- ✓ Only DCI switches upgraded
- ✓ Integer-only operations

Build one compact path-quality score ➡️ C1



Control plane normalization

Step 1: Calculate Path Score

$$\text{pathScore} = w_{dl} \cdot \text{delayScore}(p) + w_{lc} \cdot \text{linkCapScore}(p)$$

Step 2: Normalize to [0, 255]

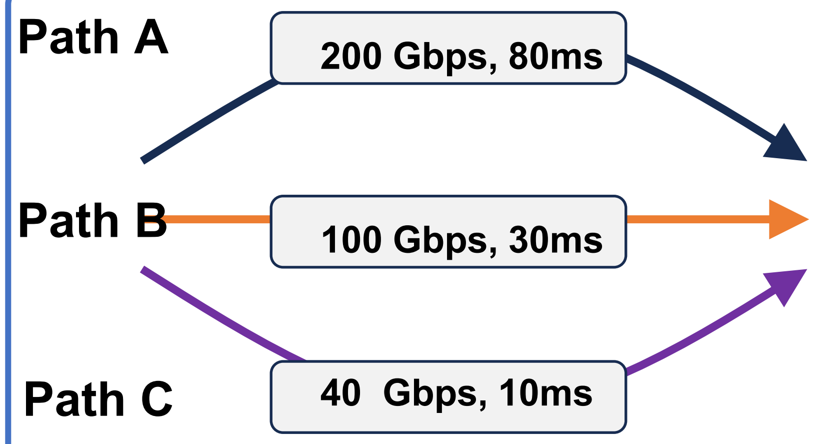
$$C_{path}(p) = \min(\text{pathScore} \gg S_{path}, 255)$$

w_{dl}: delay weight
w_{lc}: link capacity weight

S_{path}: right-shift bits
$\gg$: bit-shift division

Solution: Separate slowly-varying attributes from fast on-switch signals by **precomputing compact per-path scores** in control plane

Build one compact path-quality score ➡ C1



Inter-DC topologies exhibit **static** but **heterogeneous attributes**

A: 210 | B: 92 | C: 128

Control plane normalization

Step 1: Calculate Path Score

$$\text{pathScore} = \boxed{w_{dl} \cdot \text{delayScore}(p)} + \boxed{w_{lc} \cdot \text{linkCapScore}(p)}$$

Step 2: Normalize to [0, 255]

$$C_{path}(p) = \min(\ \text{pathScore} \gg S_{path},\ 255\ )$$

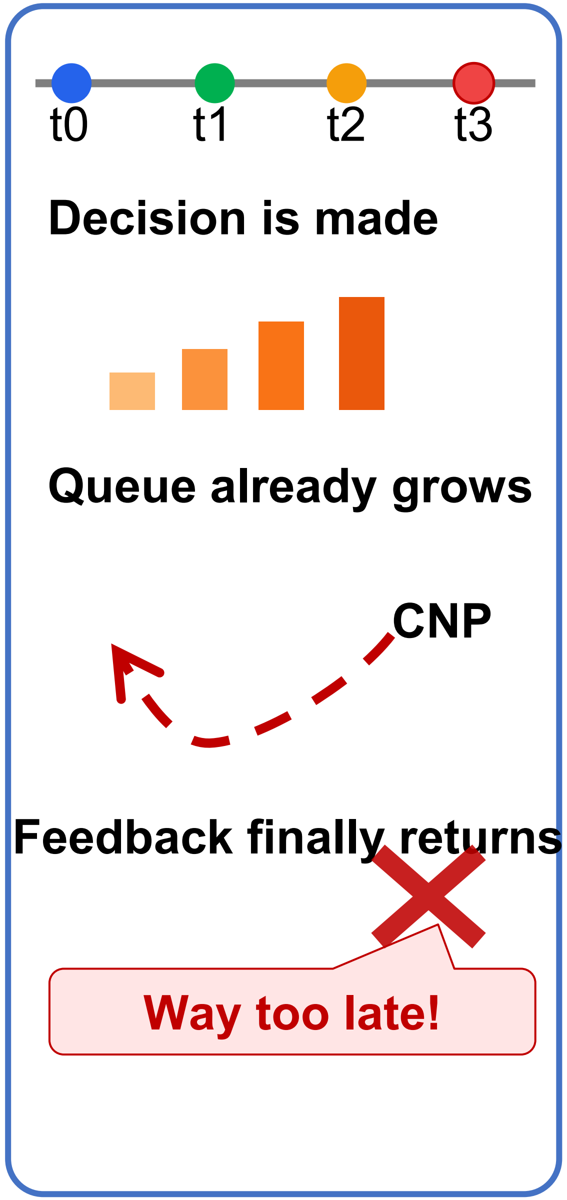
w_{dl}: delay weight
w_{lc}: link capacity weight

S_{path}: right-shift bits
$\gg$: bit-shift division

Solution: Separate slowly-varying attributes from fast on-switch signals by **precomputing compact per-path scores** in control plane

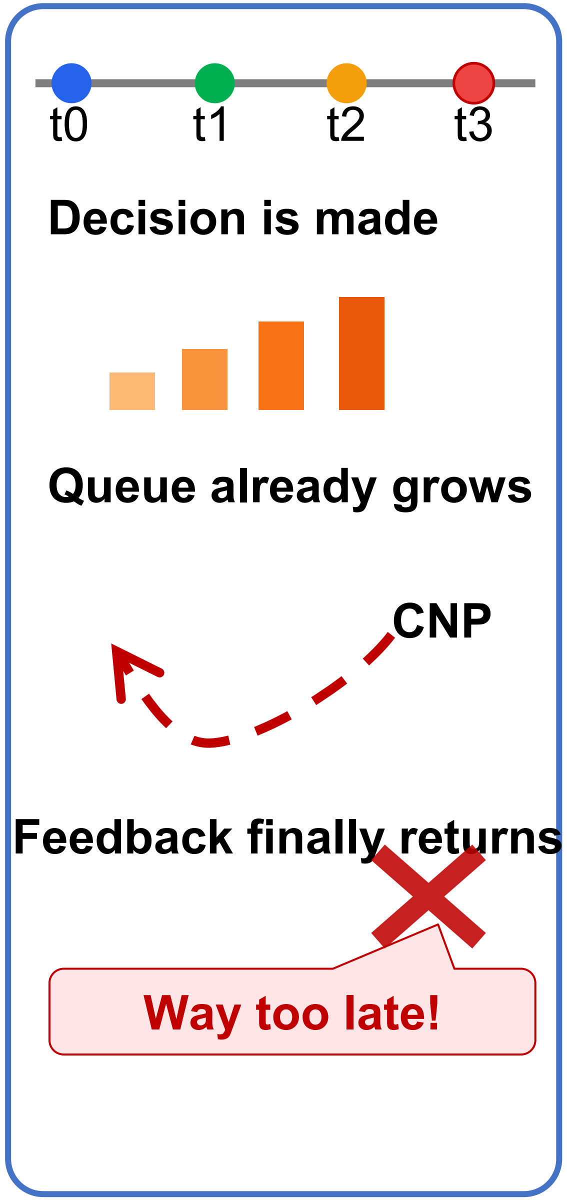
Estimate congestion directly inside the switch

C2



Estimate congestion directly inside the switch

C2



LCMP's local congestion estimate

Queue level
How many queue length of egress exists now?

Trend level
Is the queue growing quickly?

Persistence
Has the congestion lasted for a while?

Estimate congestion directly inside the switch

C2

t0 t1 t2 t3

Decision is made

Queue already grows

CNP

Feedback finally returns

Way too late!

LCMP's local congestion estimate

Queue level
How many queue length of egress exists now?

Trend level
Is the queue growing quickly?

Persistence
Has the congestion lasted for a while?

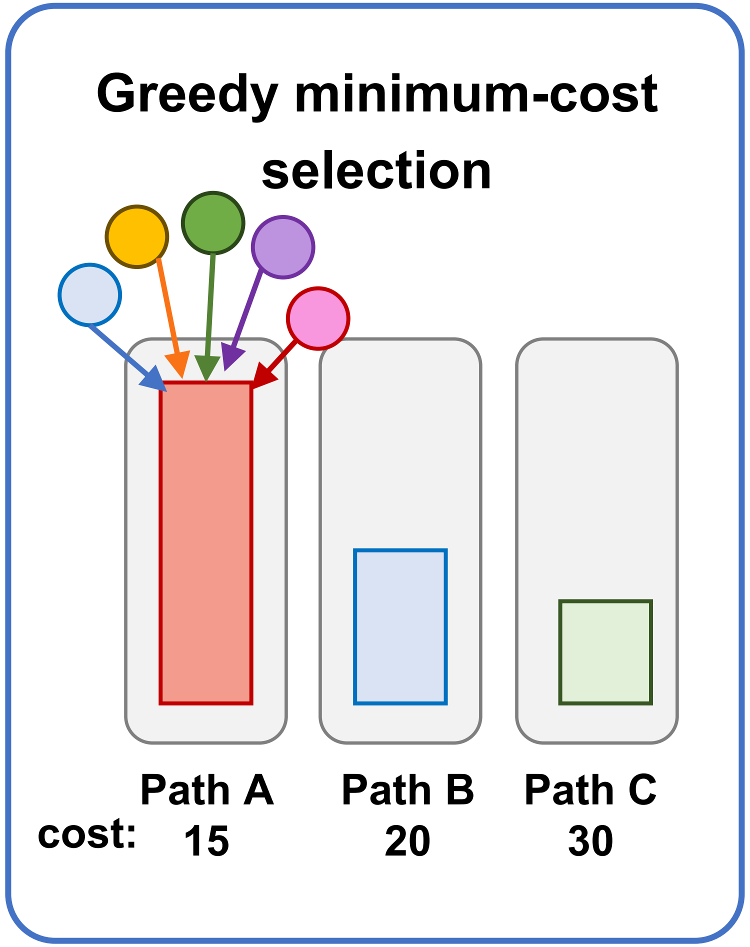
Weighted Combination

$$\text{congScore} = w_{ql} \cdot Q + wtl \cdot T + w_{dp} \cdot D$$

Faster local reaction

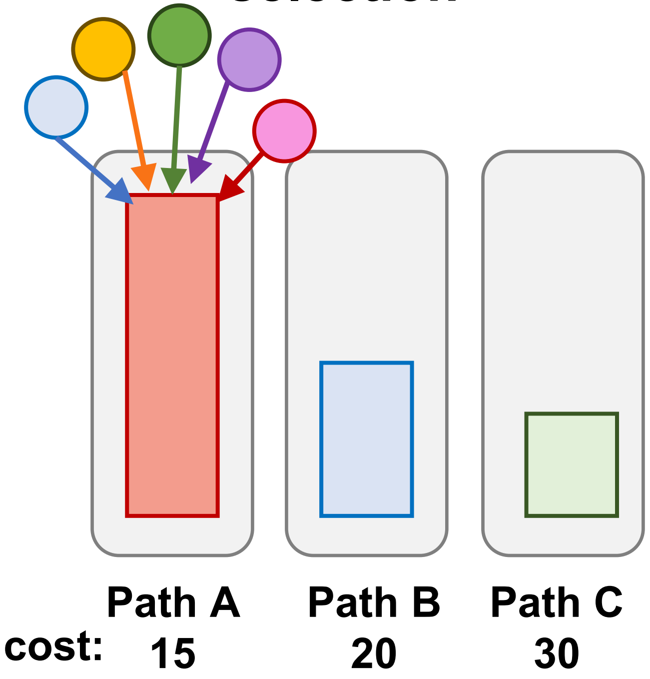
Timely congestion signal

Diversity-Preserving Selectio ➡️ C3



Diversity-Preserving Selectio ➡ C3

Greedy minimum-cost selection



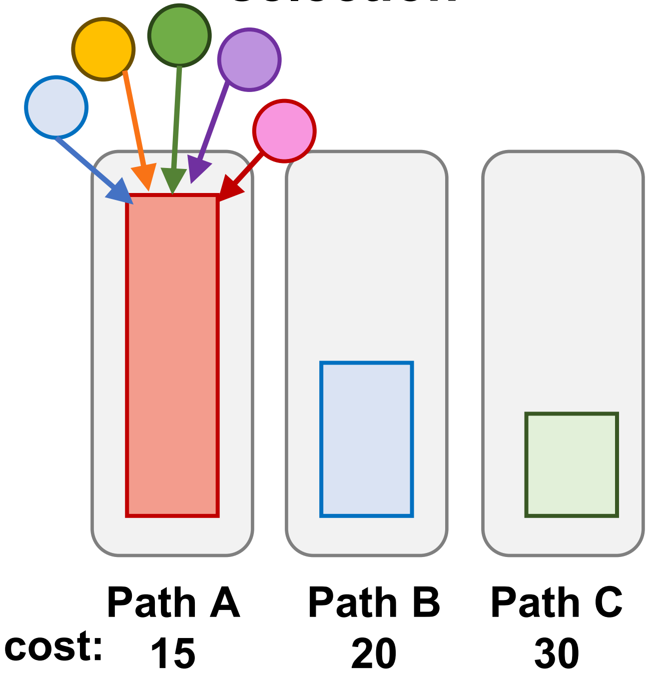
Two-stage decision

1. Rank by fused cost
2. Prune bad paths
3. Hash in reduced set

✓ **Keep quality**
Preserve diversity

Diversity-Preserving Selectio ➡️ C3

Greedy minimum-cost selection

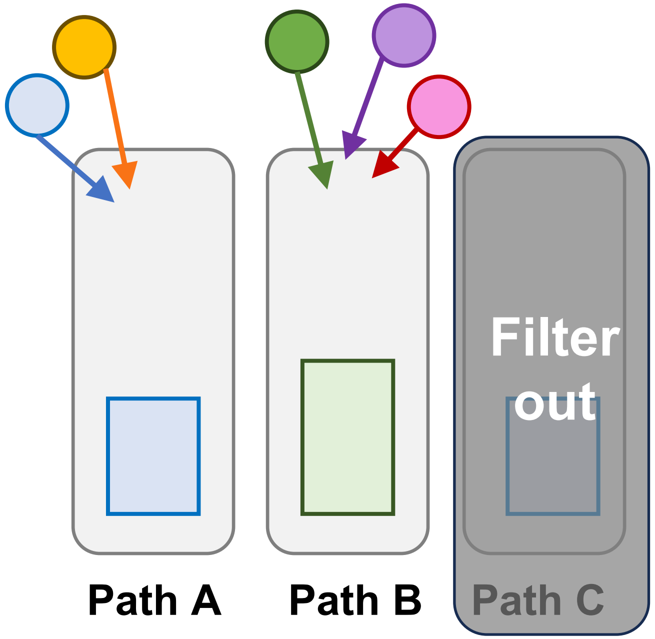


Two-stage decision

1. Rank by fused cost
2. Prune bad paths
3. Hash in reduced set

✓ **Keep quality**
Preserve diversity

What happens instead



All Paths Congested

If all candidates are highly congested, LCMP falls back to ECMP

Path Consistency

Per-flow mapping recorded in local flow cache → Subsequent packets follow same egress → Prevents reordering

Evaluation: Testbed, simulation, and realistic workloads

Implementation Highlights

Platform
Tofino switches using P4 language

No new features or complex operations required

Deployment Scope
Only DCI edge switches require upgrade

End hosts and intra-DC fabric **remain unchanged**

Compatibility
Orthogonal to congestion control

Works with DCQCN, HPCC, TIMELY, DCTCP

Details

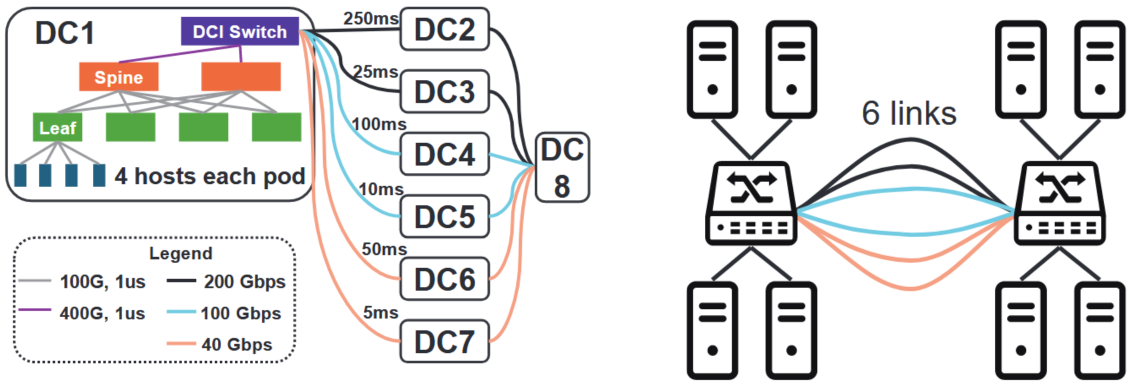
Baselines
Traditional: ECMP

SOTA: UCMP[SIGCOMM ‘24], RedTE [SIGCOMM '24]

Workloads
Microsoft Web Search, Facebook Hadoop, Alibaba Storage

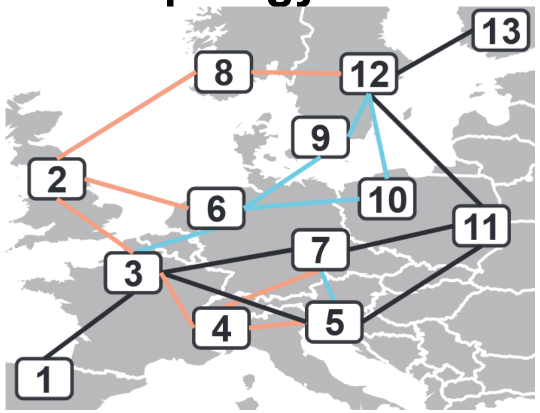
Metric
FCT slowdown (Median, tail),

Testbed simulation topology



Large-scale simulation topology

European topology (BSONetworkSolutions) drawn from the Topology Zoo



Evaluation: Small-scale testbed result

LCMP already improves both median and tail performance in a concrete system setting.

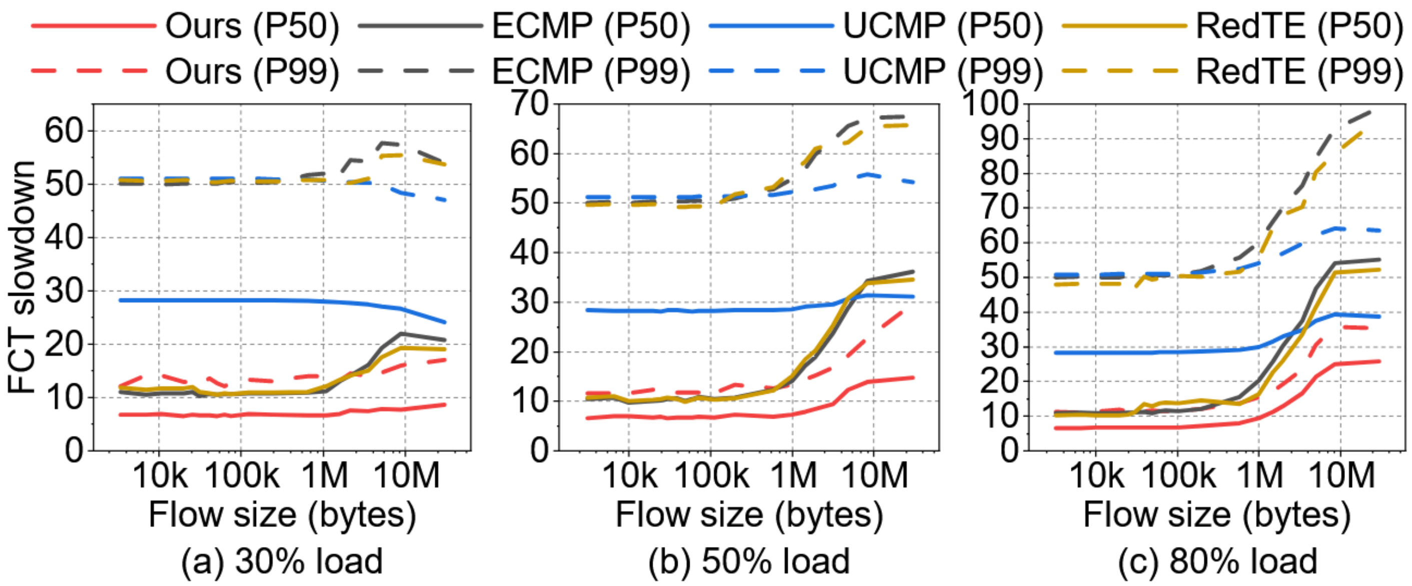


Fig.5 Median and tail FCT slowdown for Web Search on the testbed topology under 30%, 50%, 80% load.

Main result

P50: reduced by
- 36%−41% (vs ECMP)
- 76% (vs UCMP)
- 36%−54% (vs RedTE)

P99: reduced by
- 56% − 68% (vs ECMP)
- 45% − 64% (vs UCMP)
- 73% − 77% (vs RedTE)

Takeaway

The gain is visible before we even move to large-scale simulation.

LCMP is not just a simulator idea — it already works on the testbed.

Evaluation: Large-scale simulation on 13-DC

The overall gain is moderate, because many node pairs do not actually have many path choices.

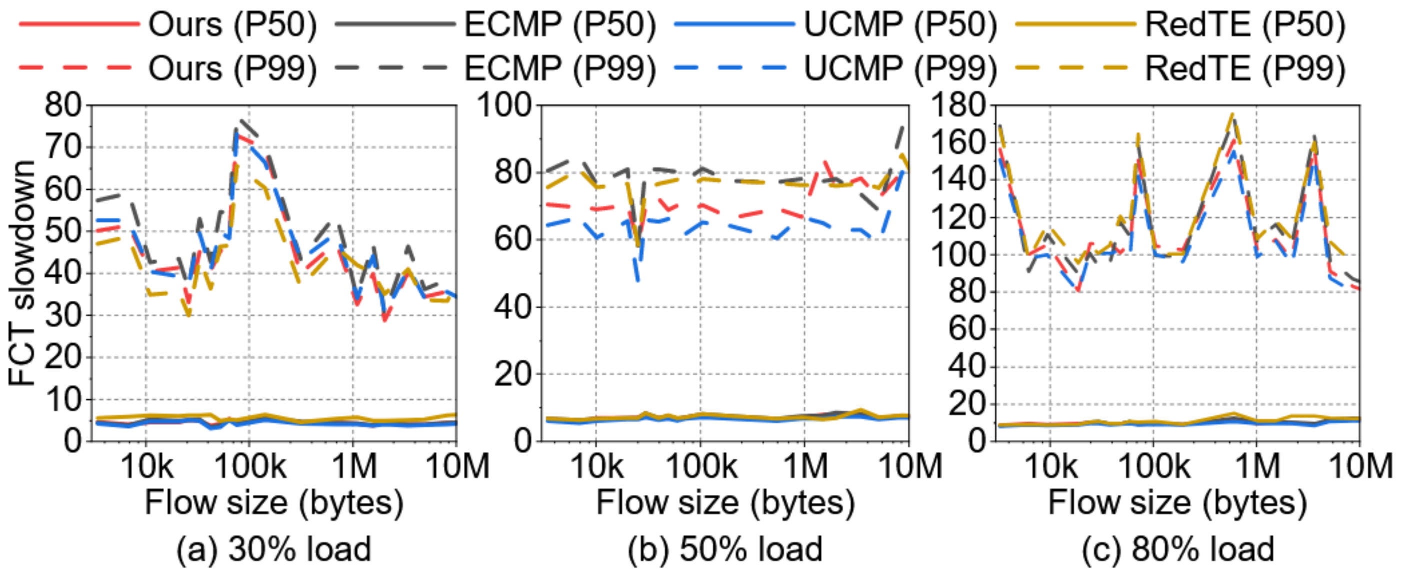


Fig.7 [System-wide validation] Median and tail FCT slowdown across all inter-DC flows at 30%, 50% and 80% loads.

Main result

P50: reduced by

- almost the same (vs ECMP, ECMP, RedTE)

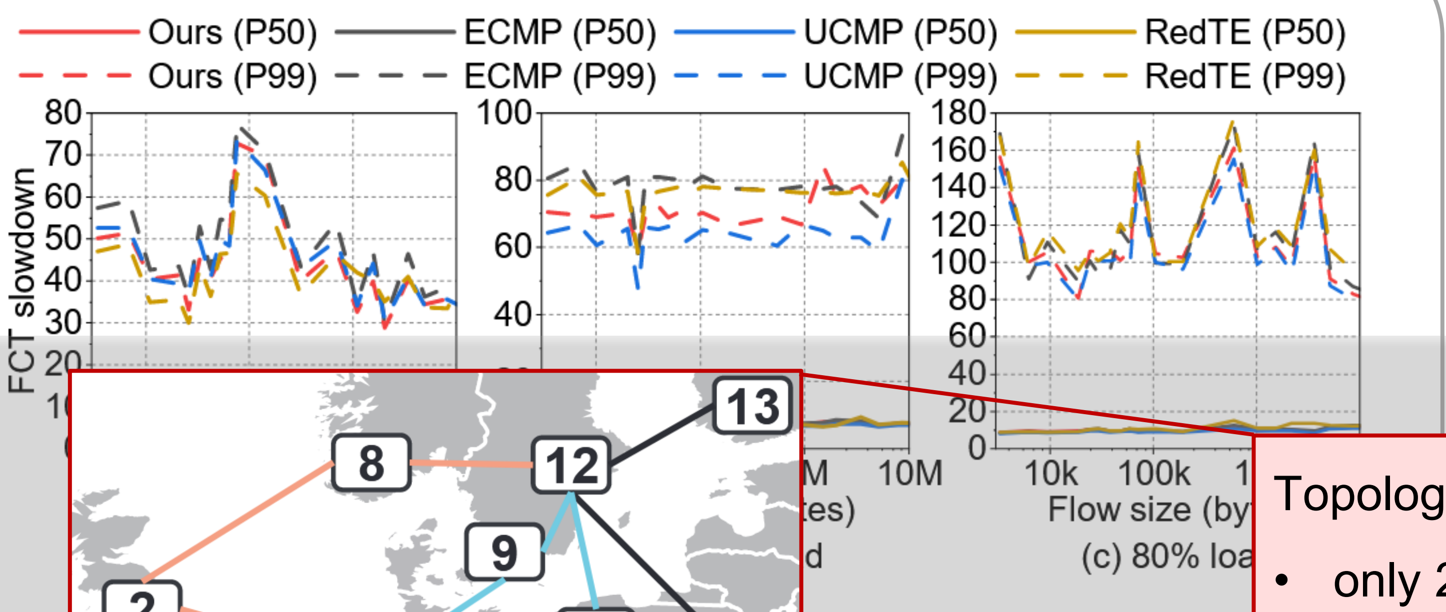
P99: reduced by

- 2% − 9% (vs ECMP)
- almost the same (vs UCMP, RedTE)

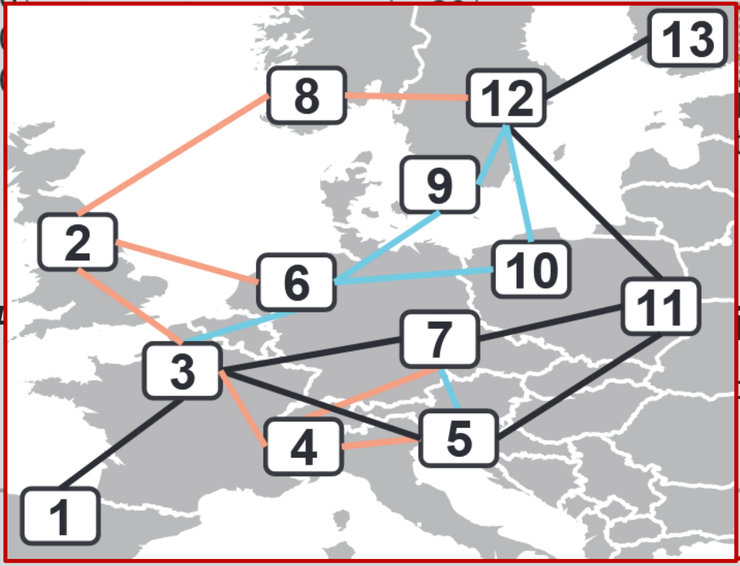
System-wide gains are real, but the strongest gains appear on multi-path cases.

Evaluation: Large-scale simulation on 13-DC

The overall gain is moderate, because many node pairs do not actually have many path choices.



F... ian and tail FCT slow...
..., 50% and 80% loads



Main result

P50: reduced by
- almost the same (vs ECMP, ECMP, RedTE)

P99: reduced by
- 2% − 9% (vs ECMP)
- almost the same (vs UCMP, RedTE)

Topology is **much sparser**

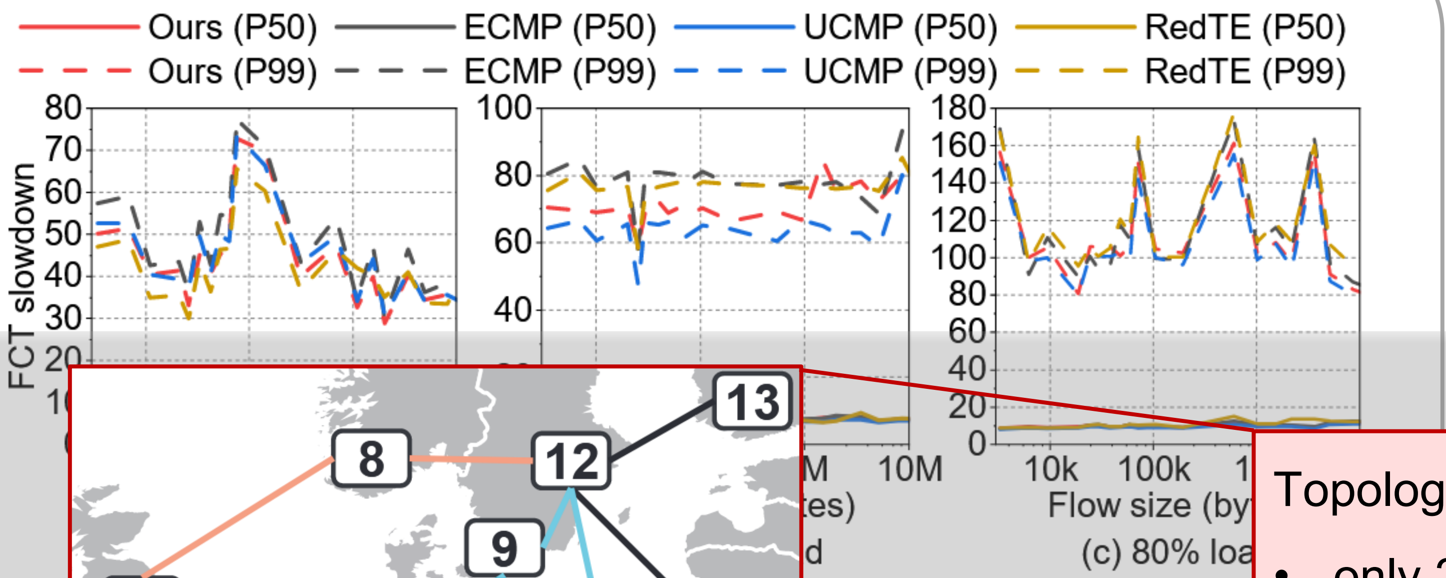
- only 25.6% node pairs have multiple candidate paths(vs. 57.1% in 8-DC)

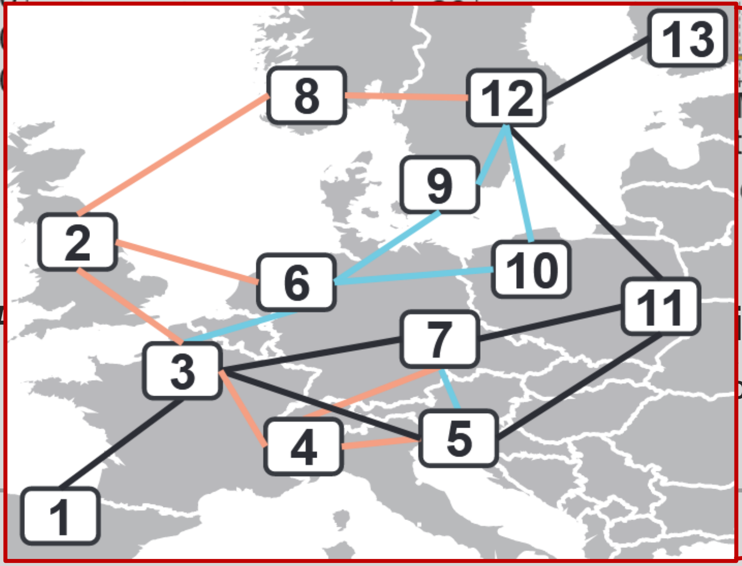
Thus, system-wide view **dilute LCMP's gains**

System-wide gains are real, but the strongest gains appear on multi-path cases.

Evaluation: Large-scale simulation on 13-DC

The overall gain is moderate, because many node pairs do not actually have many path choices.





Main result

P50: reduced by

- almost the same (vs ECMP, ECMP, RedTE)

P99: reduced by

- 2% − 9% (vs ECMP)
- almost the same (vs UCMP, RedTE)

Topology is **much sparser**

- only 25.6% node pairs have multiple candidate paths(vs. 57.1% in 8-DC)

Thus, system-wide view **dilute LCMP's gains**

Let's take a look what will happen between nodes with multiple paths.

Evaluation: Multi-path DC pair: where LCMP should matter most

Once multiple candidate routes really exist, LCMP shows clear median and tail gains.

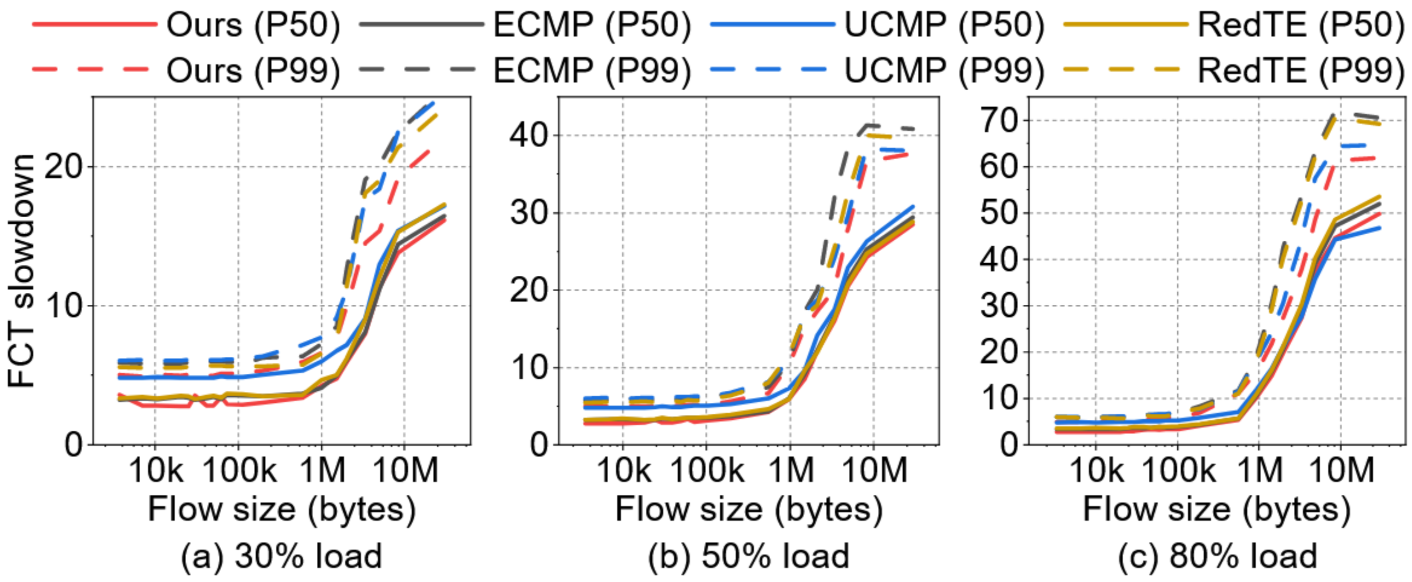


Fig.8 [DC-pair case study] Median and tail FCT slowdown for flows between DC pair (DC1, DC13) at 30%, 50% and 80% loads.

Main result

P50: reduced by
- 7%-11% (vs ECMP, RedTE)
- 25%−30% (vs UCMP)

P99: reduced by
- 15%-18% (vs ECMP, RedTE)
- 13%−16% (vs UCMP)

Takeaway

LCMP helps most where routing actually has multiple choices.

Where routing choice is meaningful, LCMP produces the strongest gains.

Conclusion

What is LCMP?

A **distributed cost-fusion routing framework** for inter-datacenter RDMA networks that combines:

- C_{path} **Control-Plane Path Quality**
  Precomputed scores encoding delay and capacity
- C_{cong} **On-Switch Congestion Signals**
  Real-time queue, trend, and duration fusion
- DP **Diversity-Preserving Selection**
  Two-stage filtering to prevent herd effects

Key Results

76% Median FCT Slowdown↓ vs SOTA baselines	**64%** Tail FCT Slowdown↓ vs SOTA baselines
1.2MB Memory Footprint 48-port, 50K flows	**~105** Integer Ops/Decision Per new flow (m=6)

Future Work

Fine-Grained Steering with OoO Tolerance

Combine selective per-flowlet routing with lightweight in-network reordering or RNIC-side OoO tracking

Cross-Layer Co-Design

Align routing decisions with transport-layer signals to avoid conflicts with senders' recovery logic

Code: https://github.com/dyyuCS/LCMP

EuroSys 2026

LCMP: Distributed Long-Haul Cost-Aware Multi-Path Routing for Inter-Datacenter RDMA Networks

Thank you for your attention.
Any questions?

dyyu@bupt.edu.cn

https://github.com/dyyuCS/LCMP

Deep Dive: Robust across workloads and congestion controls

The benefit is not tied to one traffic mix or one end-host congestion-control algorithm.

Workload sensitivity

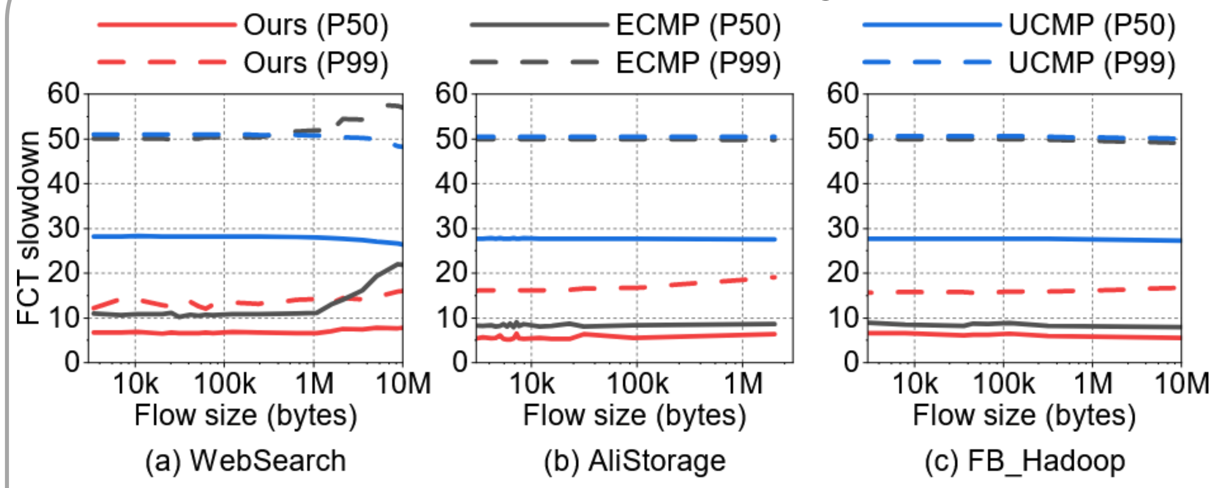


Fig.9 Workload sensitivity: median and tail FCT slowdown different three workloads.

Congestion-Control Orthogonality

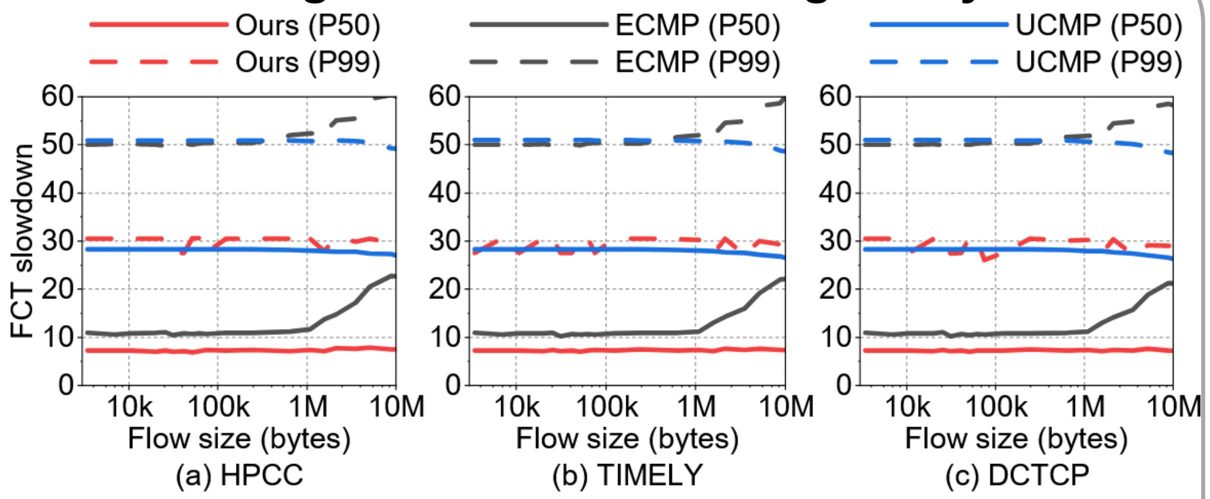


Fig.10 Congestion-control orthogonality: median and tail FCT slowdown under different CCs.

Main result

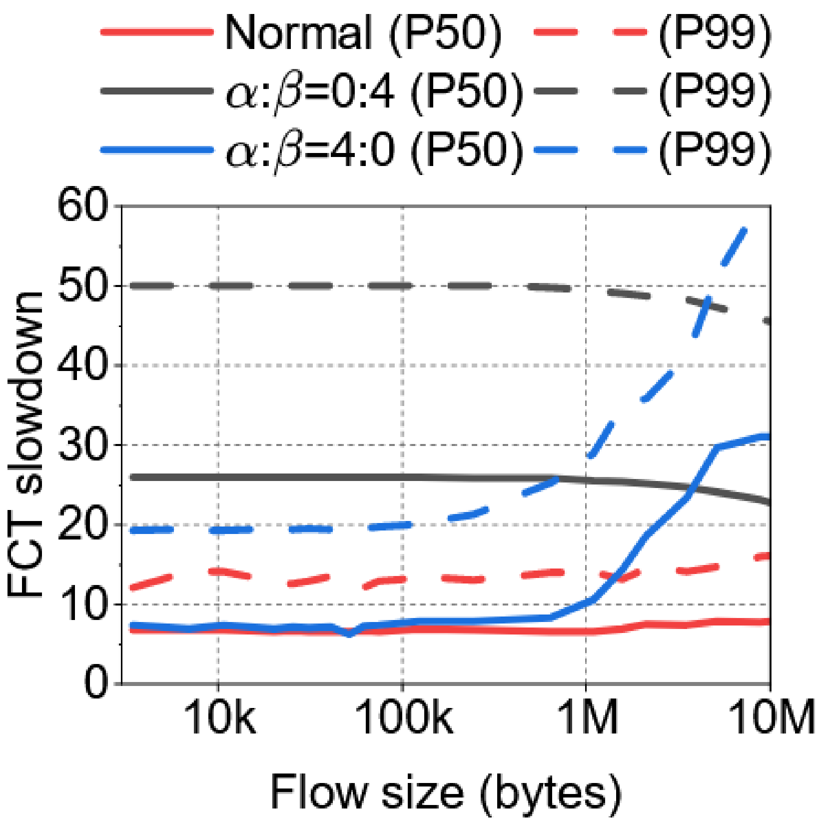
- Web Search: 36% / 58% vs ECMP; 76% / 82% vs UCMP
- AliStorage: 32% / 68% vs ECMP; 80% / 68% vs UCMP
- FB Hadoop: 26% / 69% vs ECMP; 78% / 69% vs UCMP

Insight

- LCMP is largely orthogonal to CC design
- Existing intra-DC CC assumptions **weaken** over large RTTs
- Inter-DC performance needs faster feedback and cross-layer coordination

Sensitivity Analysis: Ablation-both terms are necessary

Removing either path-quality or congestion creates a clear and interpretable failure mode.



(a) [Ablation analysis]

Remove path-quality (α = 0)

P50: 6.8 → 26.0 (+280%)

P99: 12.1 → 50.0 (+312%)

Remove congestion term (β = 0)

P50 : 8.7 → 31.2 (+260%)

P99: 17.1 → 58.4 (+240%)

Takeaway

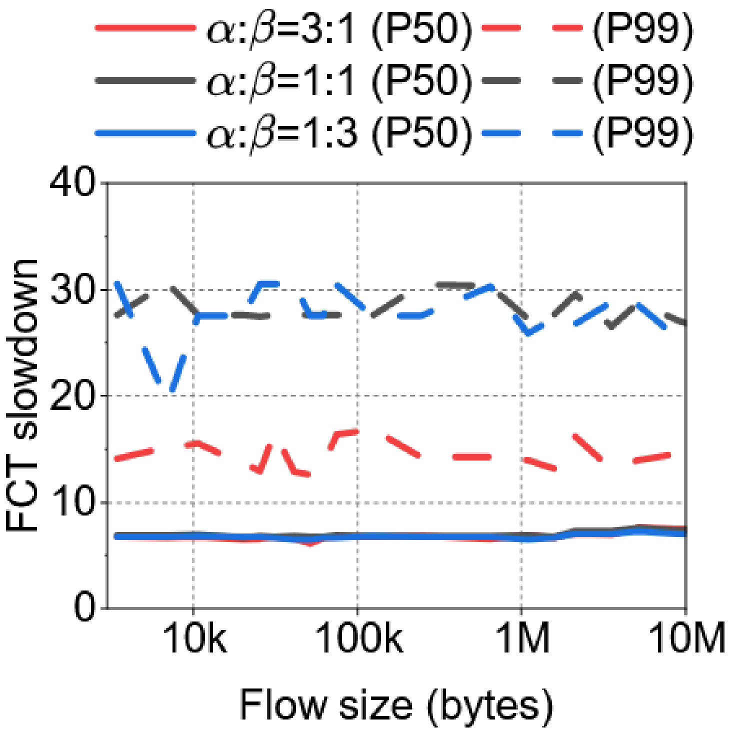
Path-quality keeps medians low.

Congestion sensing keeps tails under control.

Full LCMP works because it combines the two signals; neither one is sufficient alone.

Sensitivity Analysis: Delay-biased fusion gives the most stable tails

Both the global fusion weights and the path-quality weights should favour delay over capacity.



(b) [Global weight analysis] Weight tuples $(\alpha, \beta) = (3,1), (1,1), (1,3)$

Observation

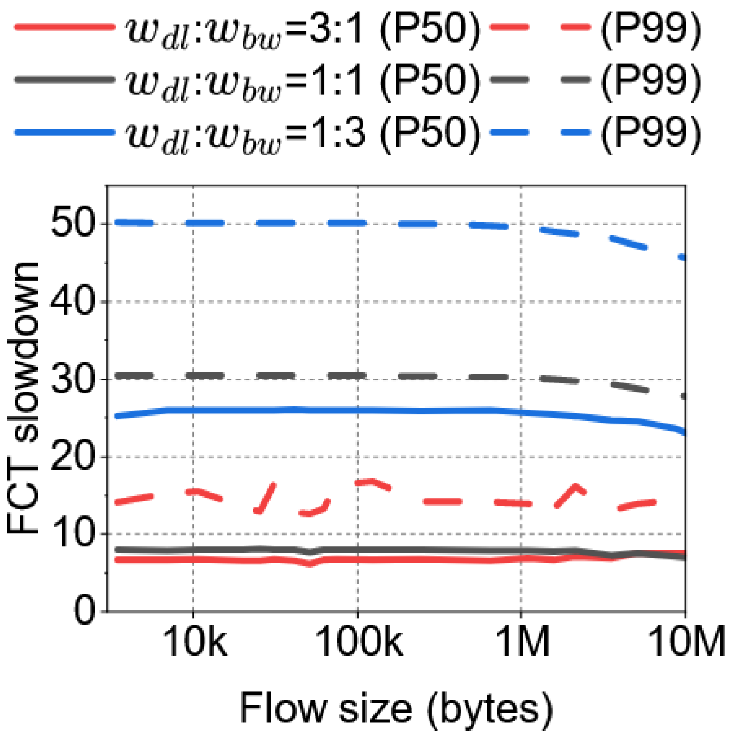
(α,β)=(3,1): P99 is typically around 12–16

(1,1) or (1,3): P99 is often around 24–30

Observation

(w_{dl}, w_{bw})=(3,1): best median and tail

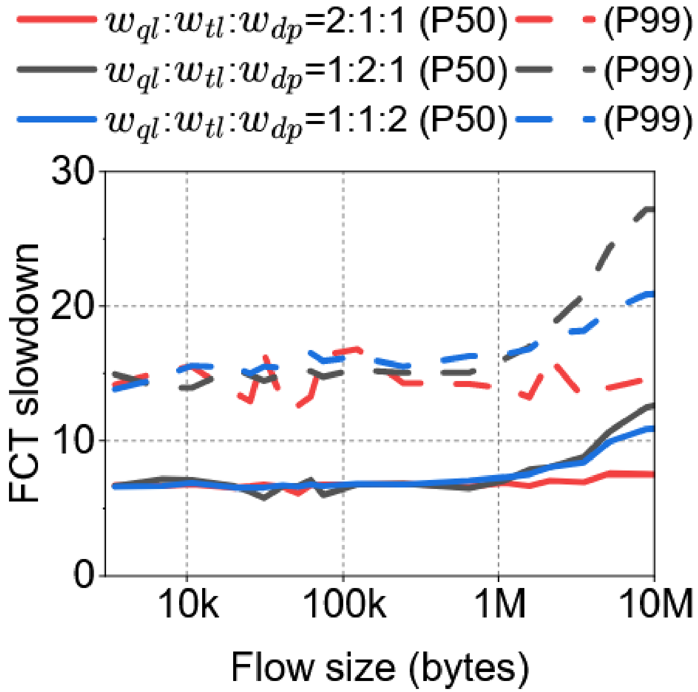
(1,3): P99 often reaches 43–50



(c) [Path-quality weights analysis] Weights tuples $(w_{dl}, w_{lc}) = (3,1), (1,1), (1,3)$

Sensitivity Analysis: Queue-first congestion weight is the safest

Emphasizing instantaneous queue level works better than over-emphasizing trend or persistence.



(d) [Congestion-cost weights analysis] Weight tuples (w_{ql}, w_{tl}, w_{dp}) = $(2, 1, 1), (1, 2, 1), (1, 1, 2)$.

Best default: (2,1,1)

Median stays around 6.1–7.6

P99 stays around 12–17

Trend-heavy or duration-heavy

P99 grows for the largest flows

p50 can rise from about 6–7 to about 8–14

Takeaway

Queue level is the safest signal.

Trend and persistence help, but should not dominate

Default recommendation: use a conservative queue-first congestion score, i.e., (2,1,1).