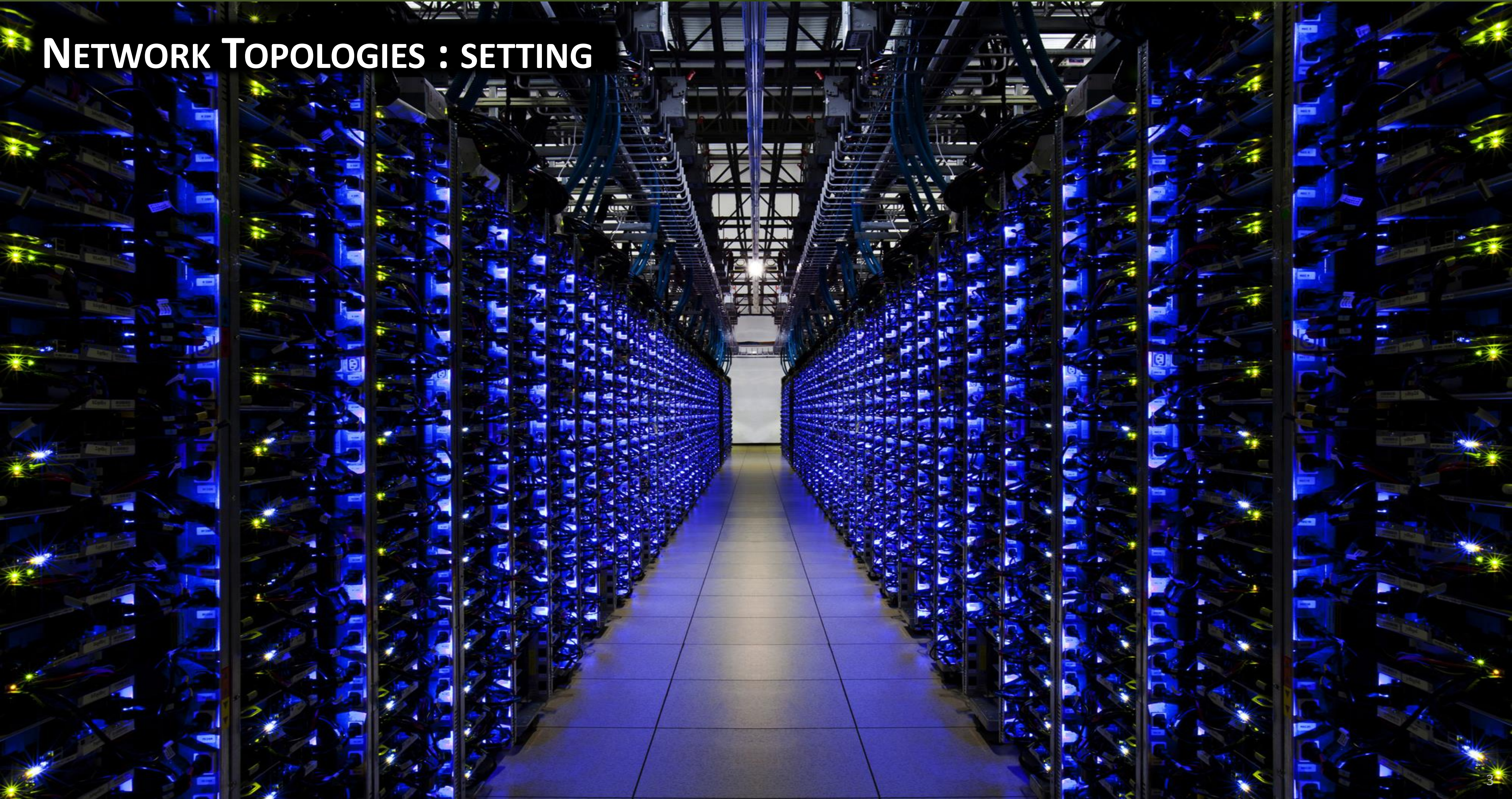
FatPaths: Routing in Supercomputers and Data Centers when Shortest Paths Fall Short

Maciej Besta, Marcel Schneider, Marek Konieczny, Karolina Cynk
Erik Henriksson, Salvatore Di Girolamo, Ankit Singla, Torsten Hoefler


summit
IBM
NASA

The interconnect: a key part of supercomputers and data centers, relevant both for high performance and low cost

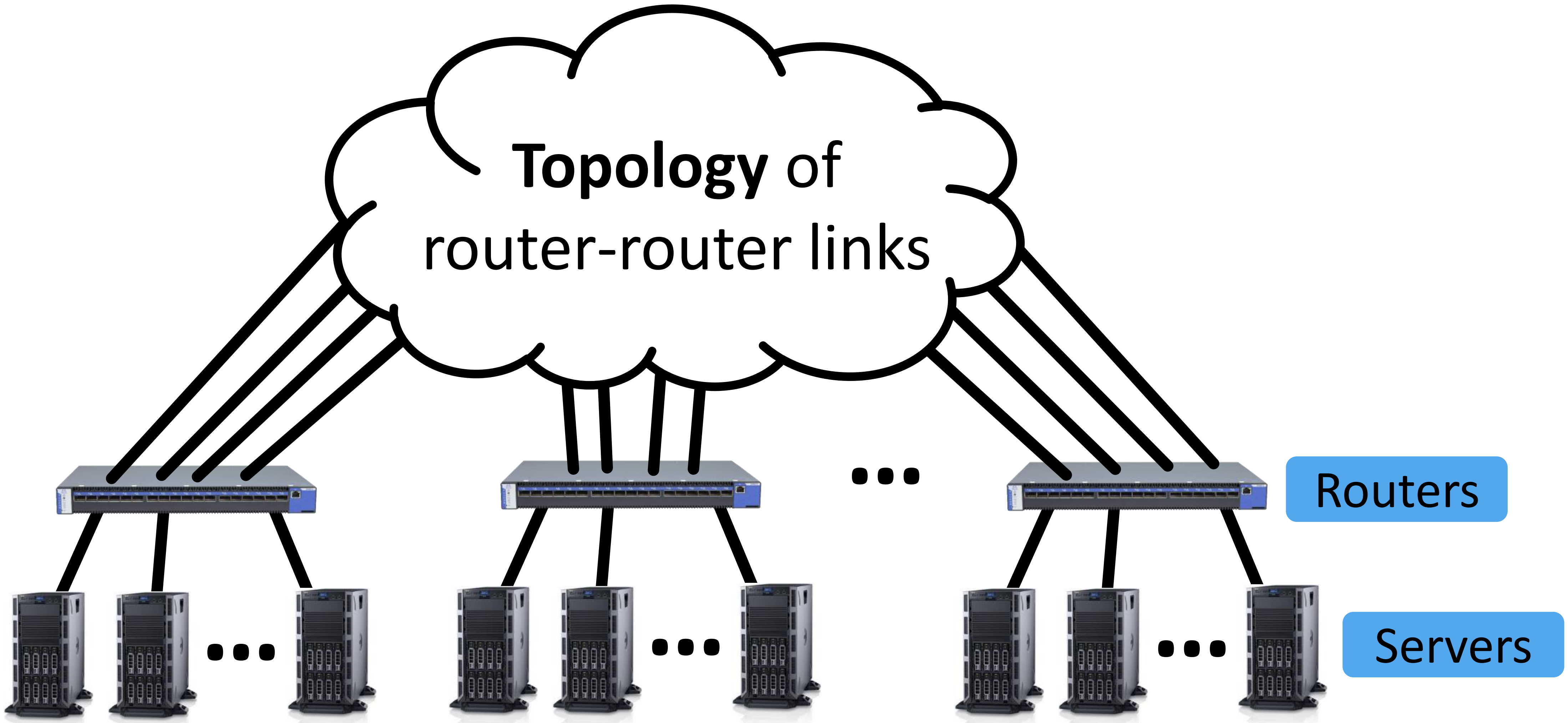
Network Topologies : setting



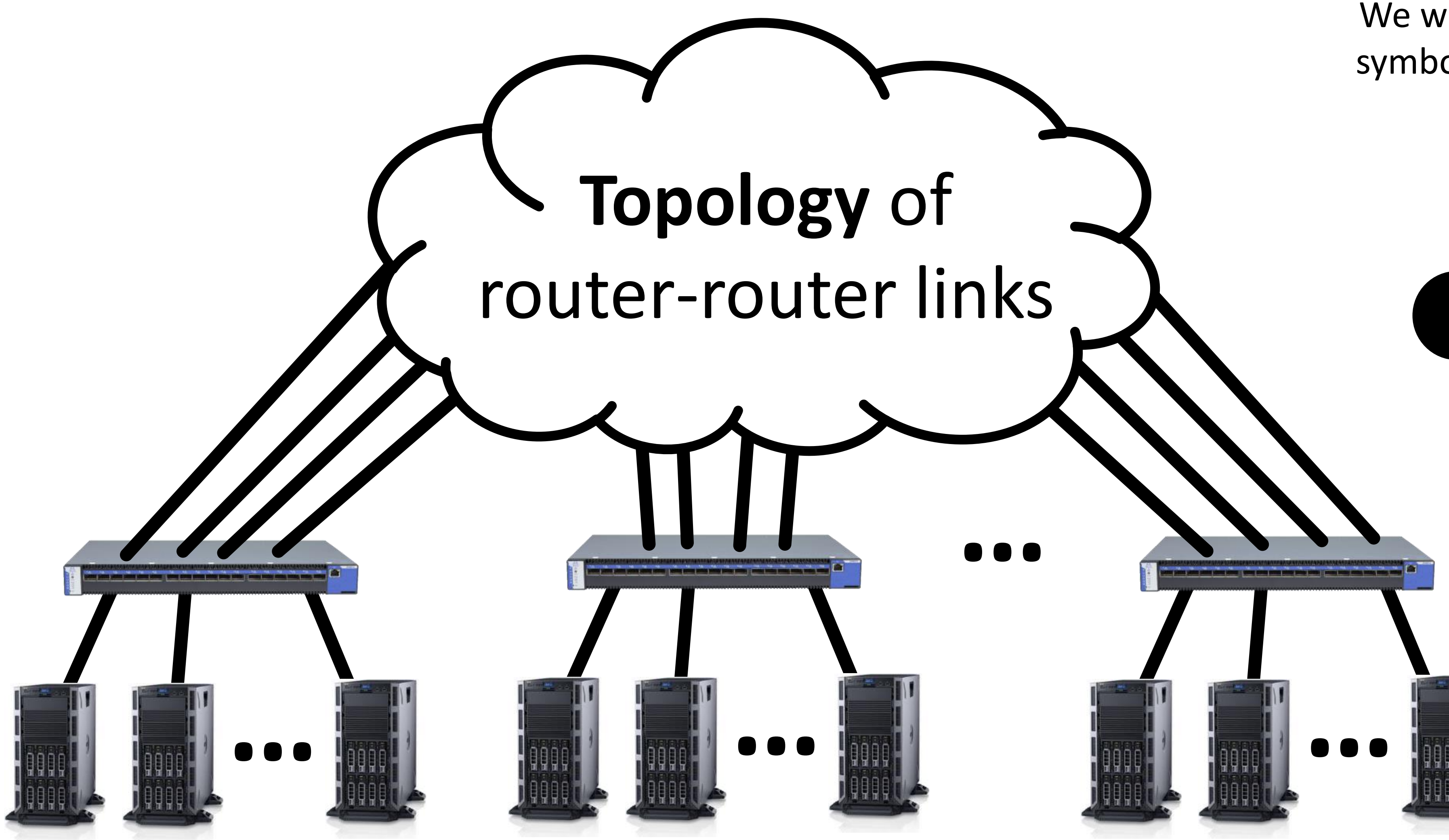
Network Topologies : setting

Network Topologies : setting

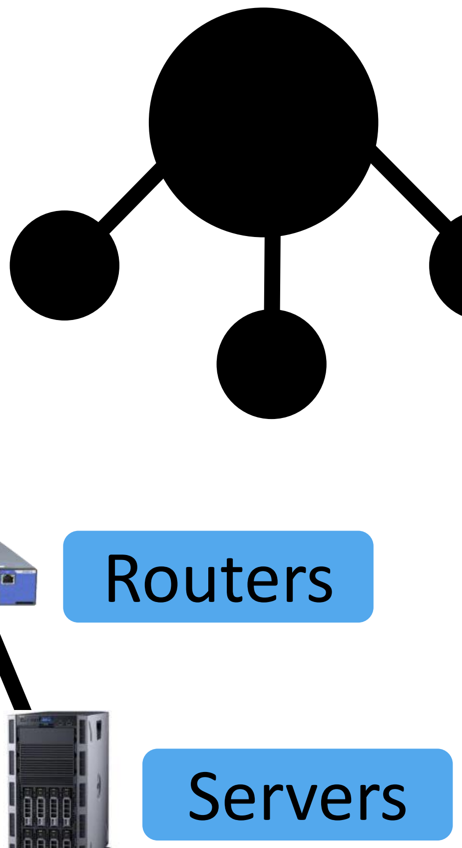
NETWORK TOPOLOGIES : SETTING



Network Topologies : Setting

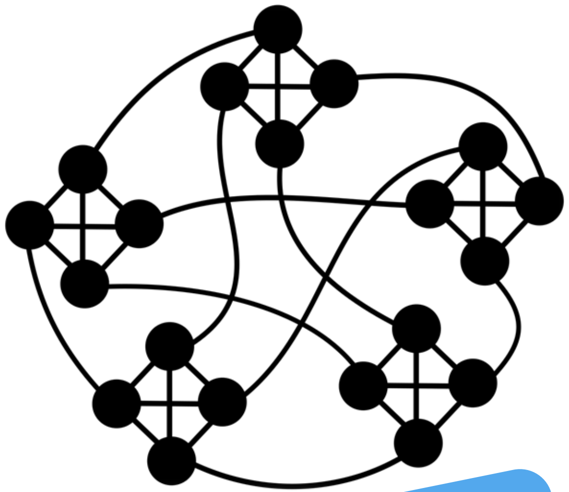


We will often use the following symbols for routers and servers



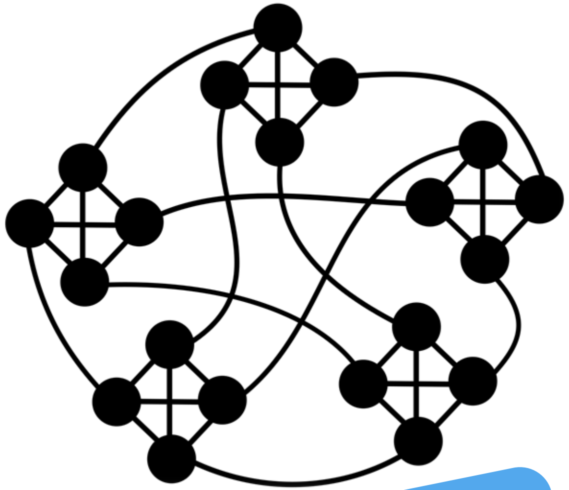
LOW-DIAMETER NETWORK TOPOLOGIES

Low-Diameter Network Topologies



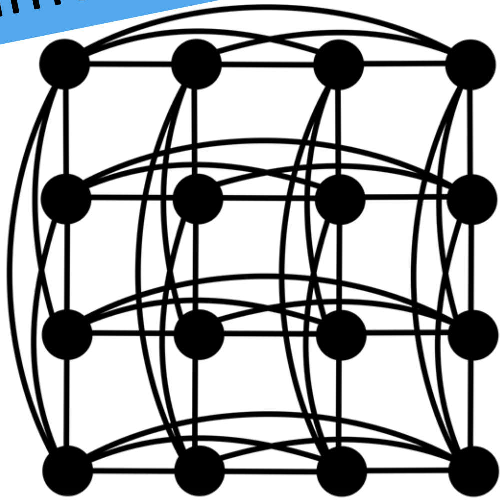
Dragonfly [2008]
diameter = 3

LOW-DIAMETER NETWORK TOPOLOGIES



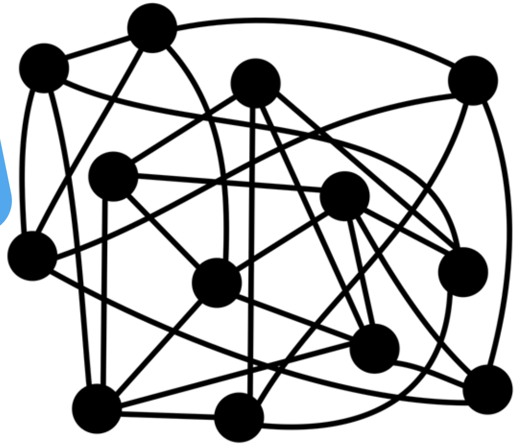
Dragonfly [2008]
diameter = 3

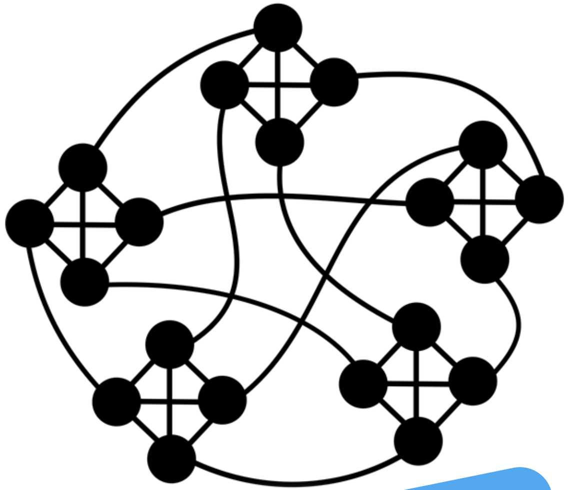
HyperX [2009]
diameter = 3



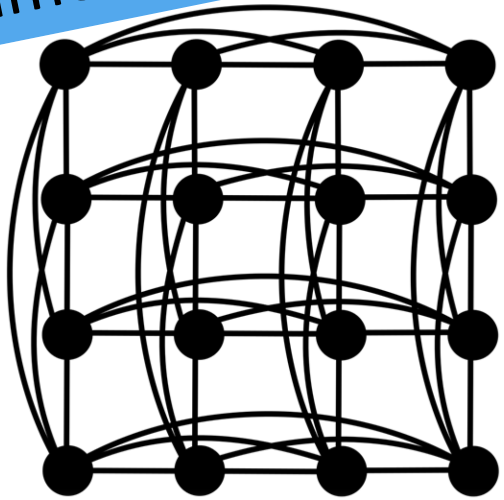
LOW-DIAMETER NETWORK TOPOLOGIES

Jellyfish [2012]
diameter ≤ 3





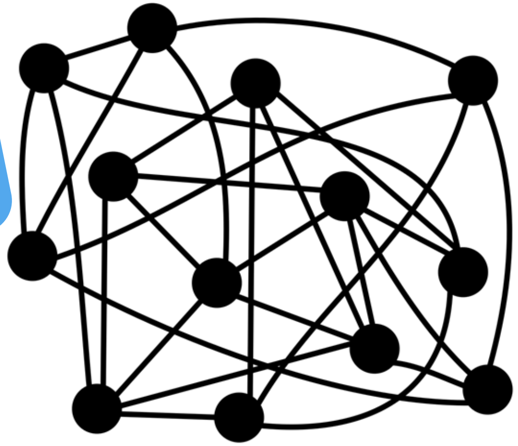
HyperX [2009]
diameter = 3



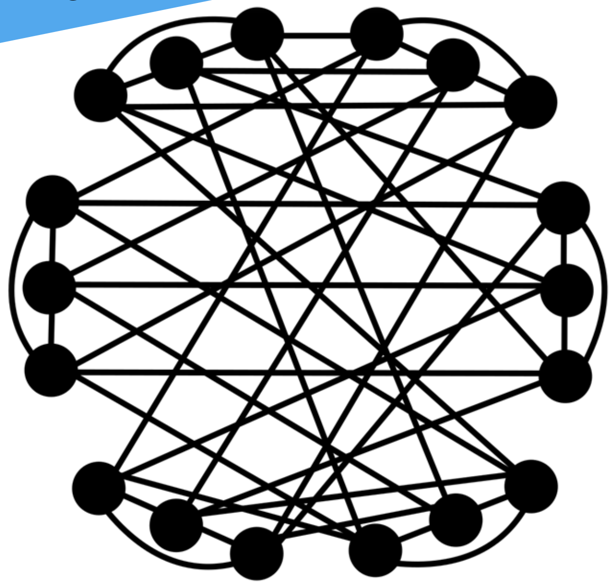
Dragonfly [2008]
diameter = 3

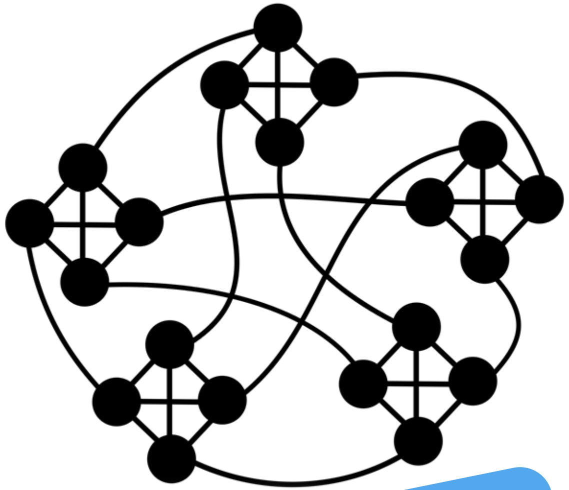
Low-Diameter Network Topologies

Jellyfish [2012]
diameter ≤ 3



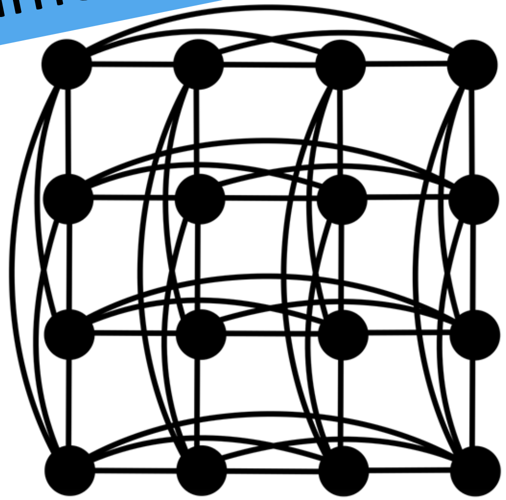
Slim Fly [2014]
diameter = 2



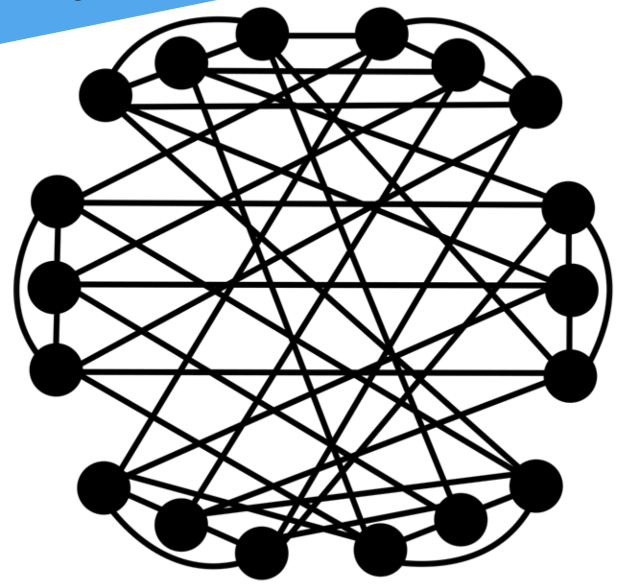


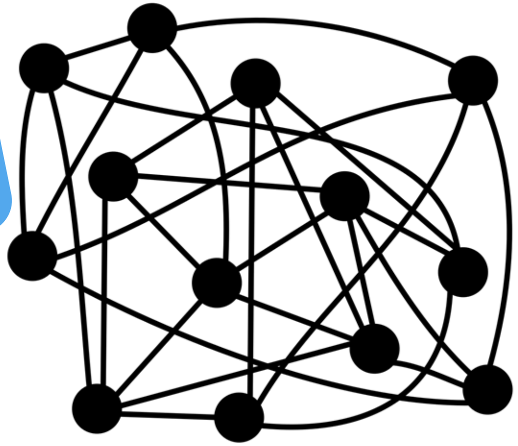
Dragonfly [2008]
diameter = 3

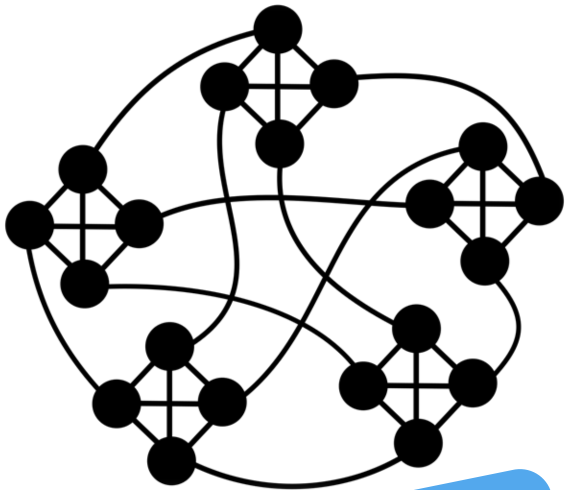
HyperX [2009]
diameter = 3

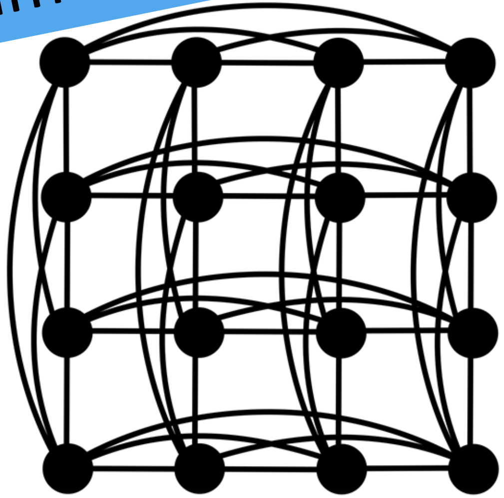


Low-Diameter Network Topologies





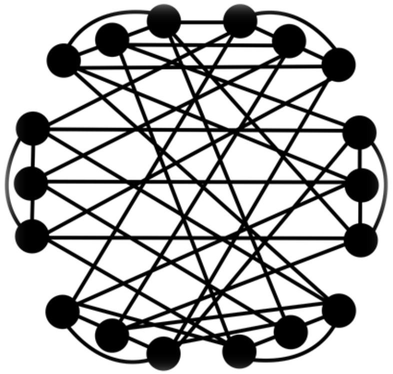


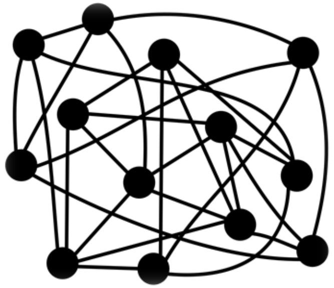


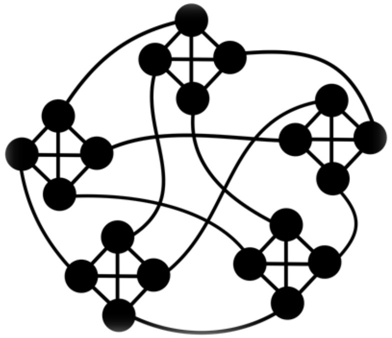
Low-Diameter Network Topologies

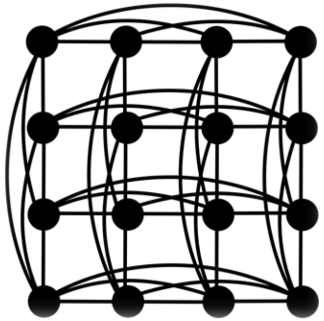
Low-Diameter Network Topologies

Why care? (there is already Clos, Fat tree, etc.).



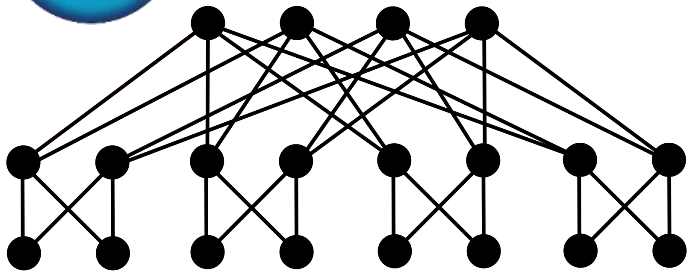


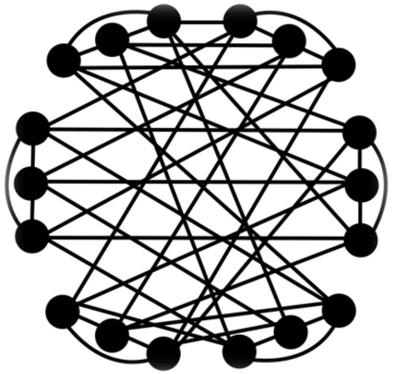


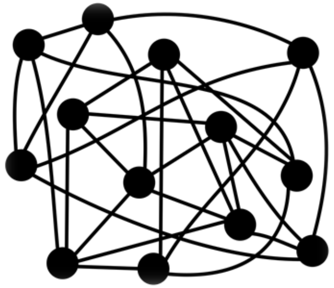


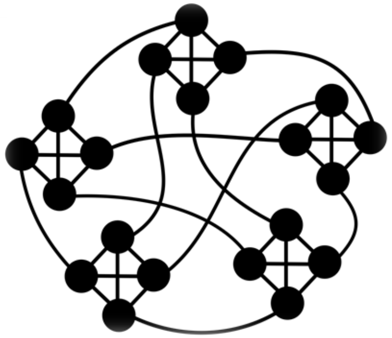
Low-Diameter Network Topologies

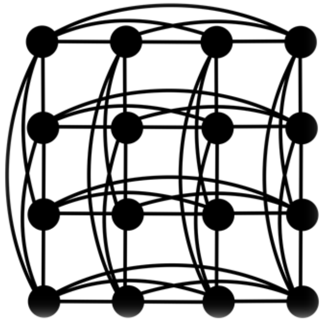
Why care? (there is already Clos, Fat tree, etc.).





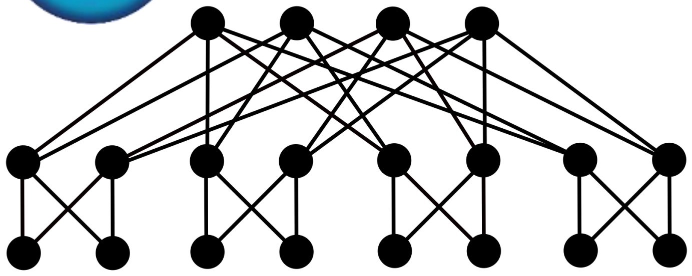


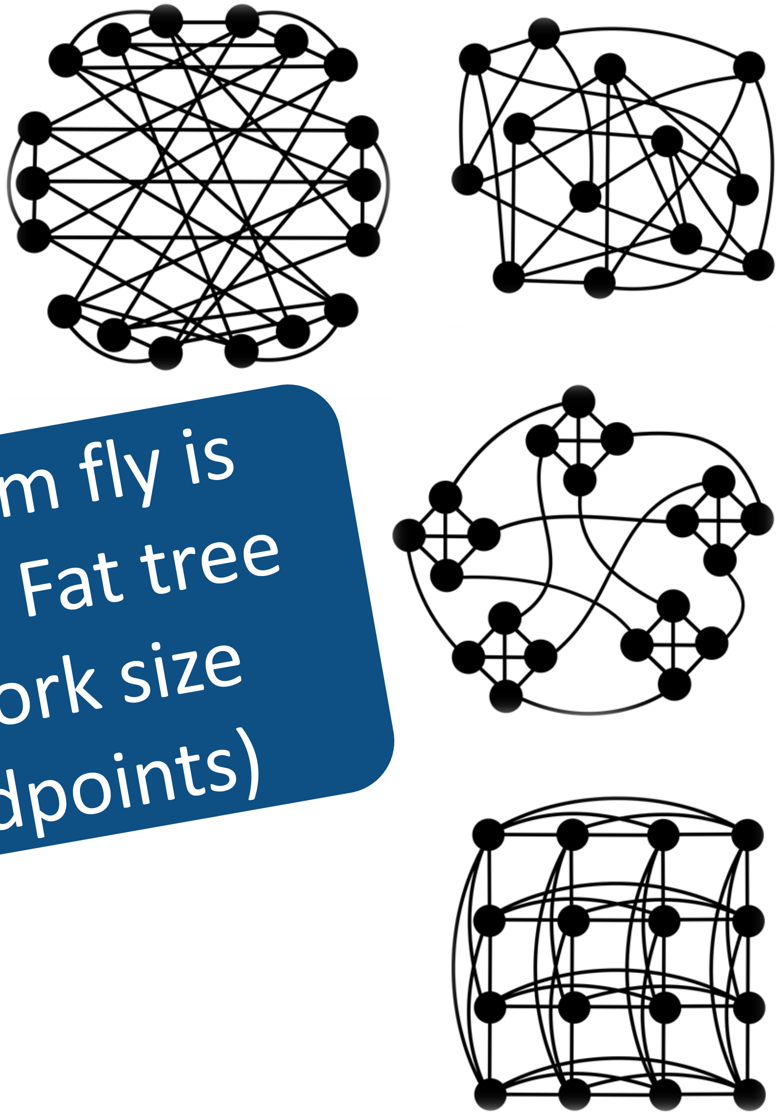




Low-Diameter Network Topologies

Why care? (there is already Clos, Fat tree, etc.).

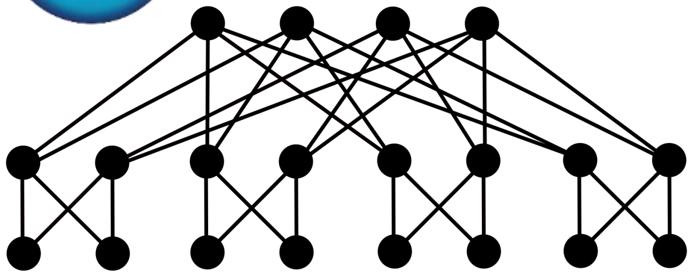




Cost-effective (Slim fly is 50% cheaper than Fat tree for a fixed network size counted in #endpoints)

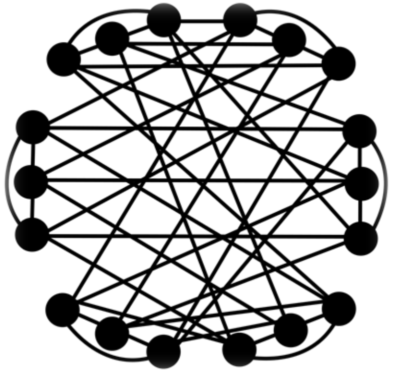
Low-Diameter Network Topologies

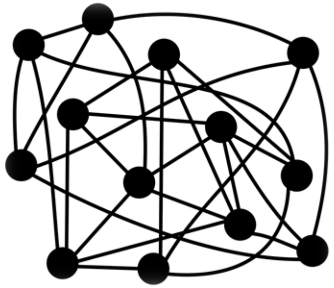
Why care? (there is already Clos, Fat tree, etc.).

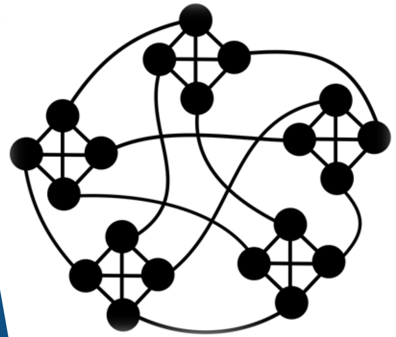


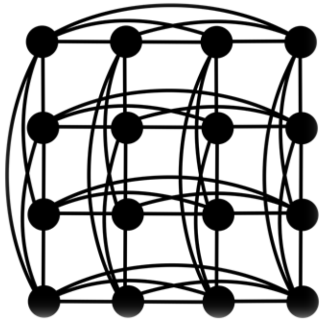
Energy-efficient (fewer routers and thus buffers)

Cost-effective (Slim fly is 50% cheaper than Fat tree for a fixed network size counted in #endpoints)



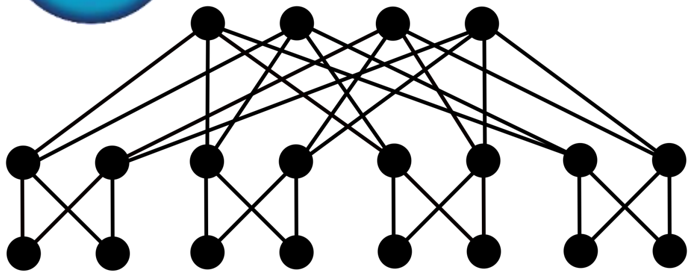






Low-Diameter Network Topologies

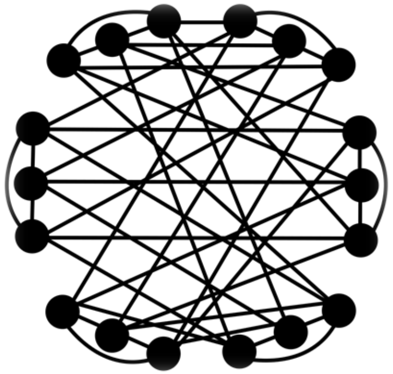
Why care? (there is already Clos, Fat tree, etc.).

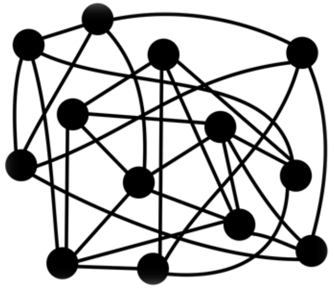


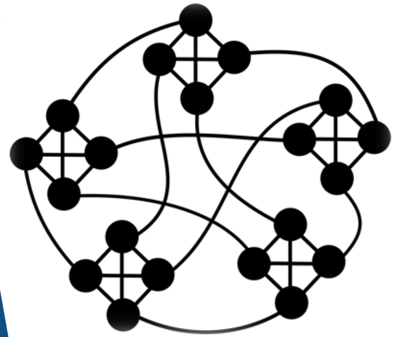
Energy-efficient (fewer routers and thus buffers)

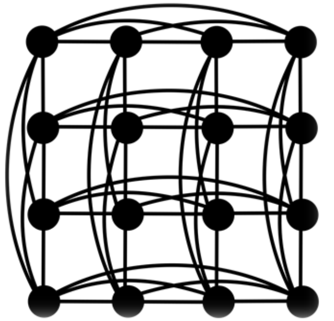
Cost-effective (Slim fly is 50% cheaper than Fat tree for a fixed network size counted in #endpoints)

Low latency (from low diameters)

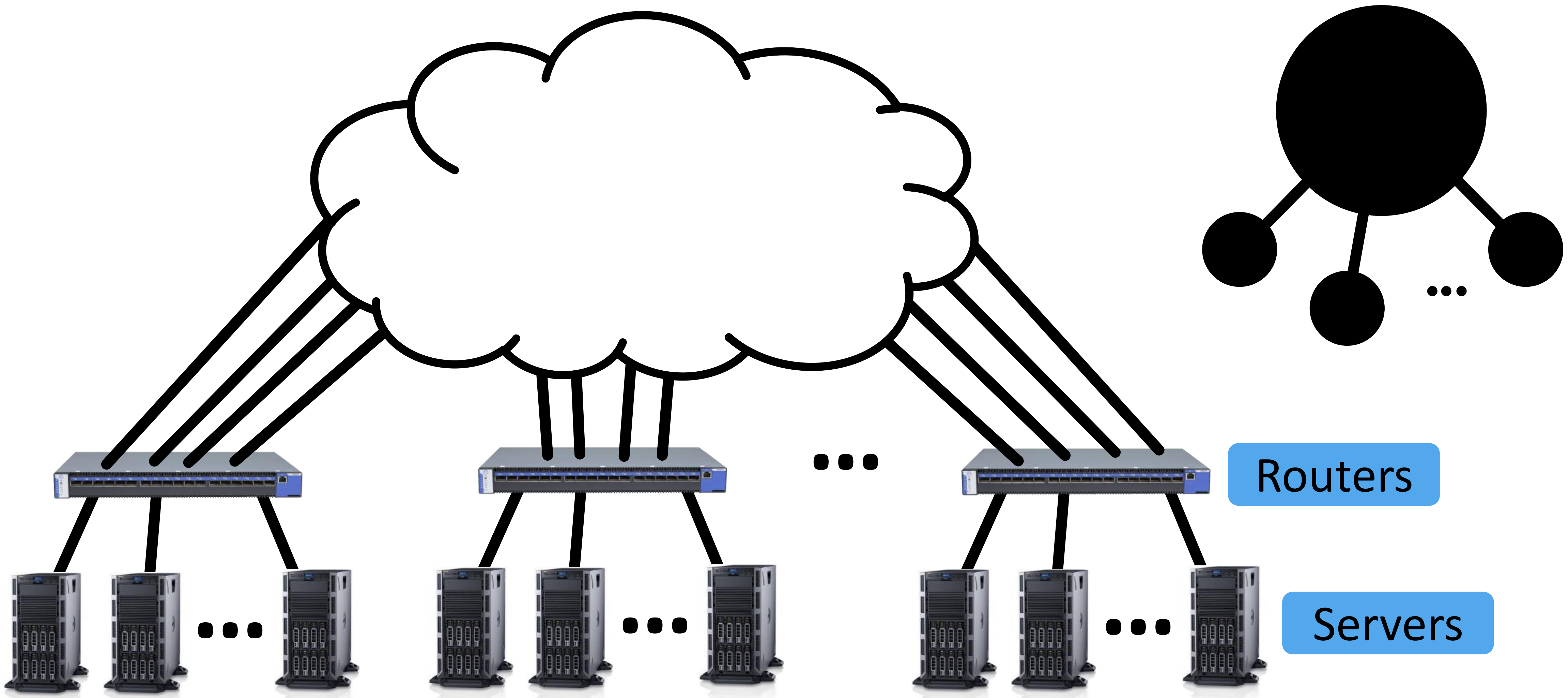




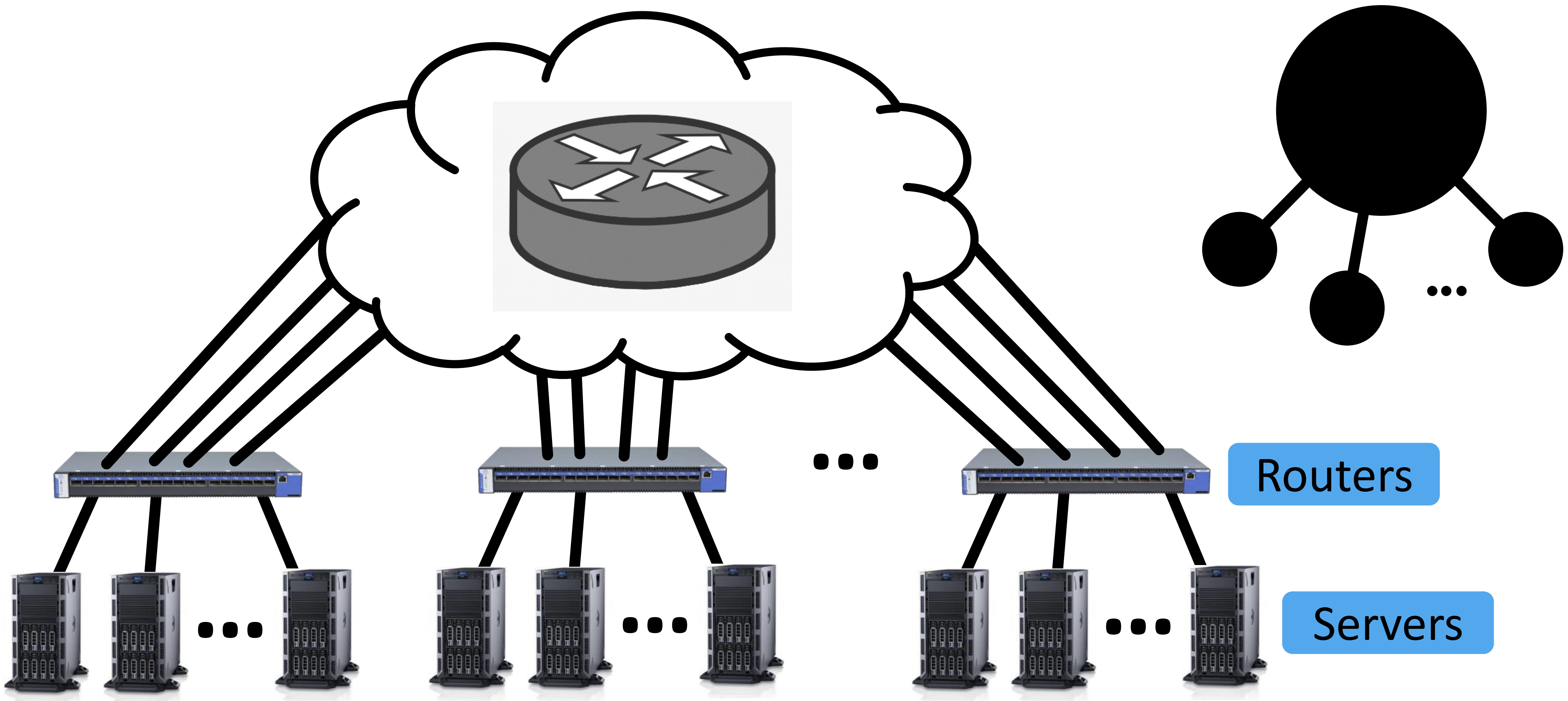


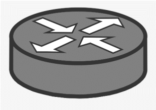


Network Topologies : Routing



Network Topologies : Routing



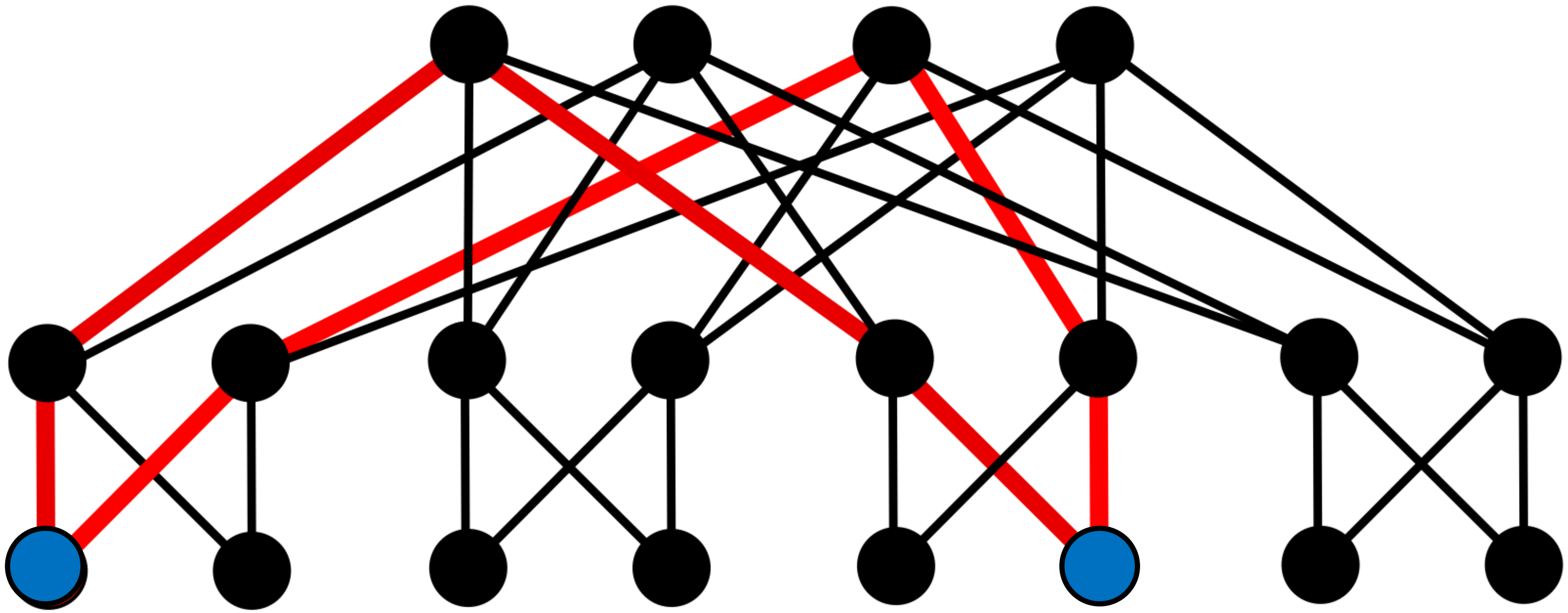
Routing in Fat Trees 

ROUTING IN FAT TREES

High-performance routing is facilitated by **numerous multiple shortest paths of equal lengths** between any endpoints

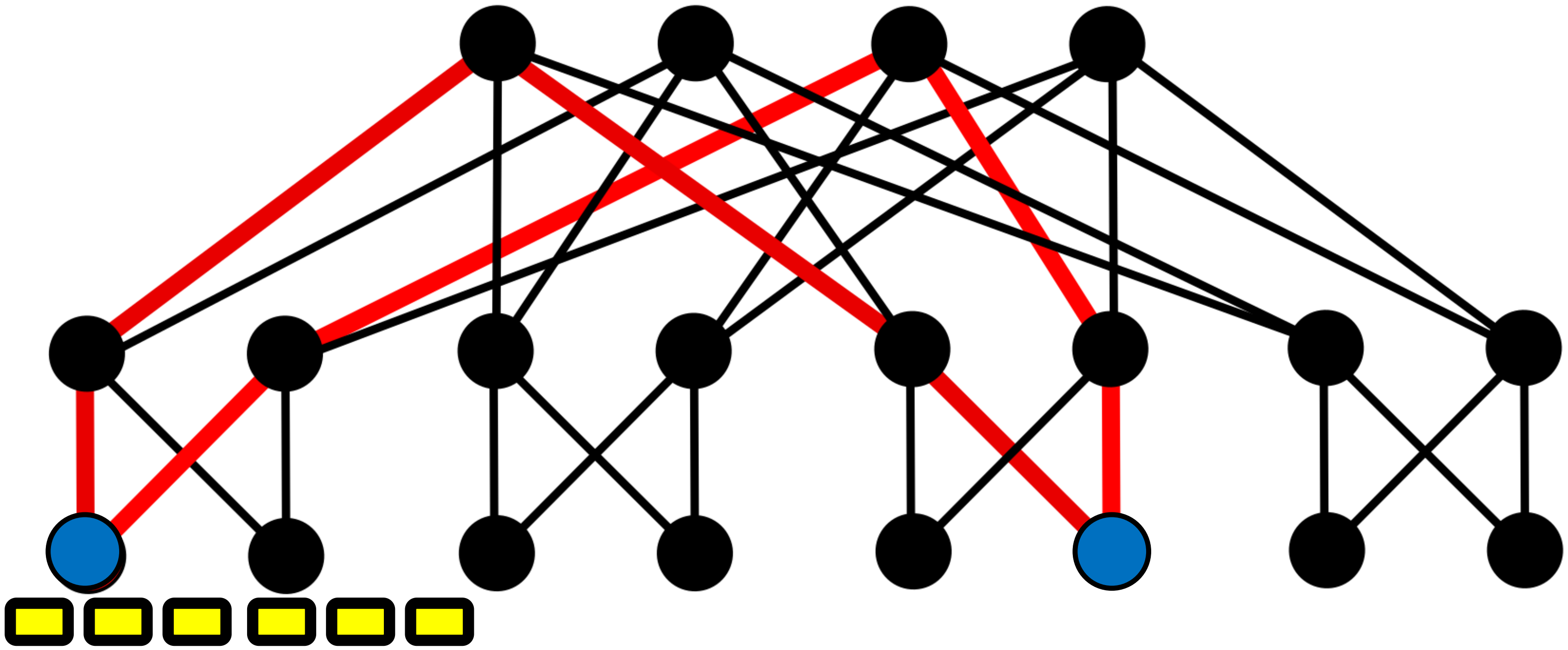
Routing in Fat Trees

High-performance routing is facilitated by **numerous multiple shortest paths of equal lengths** between any endpoints



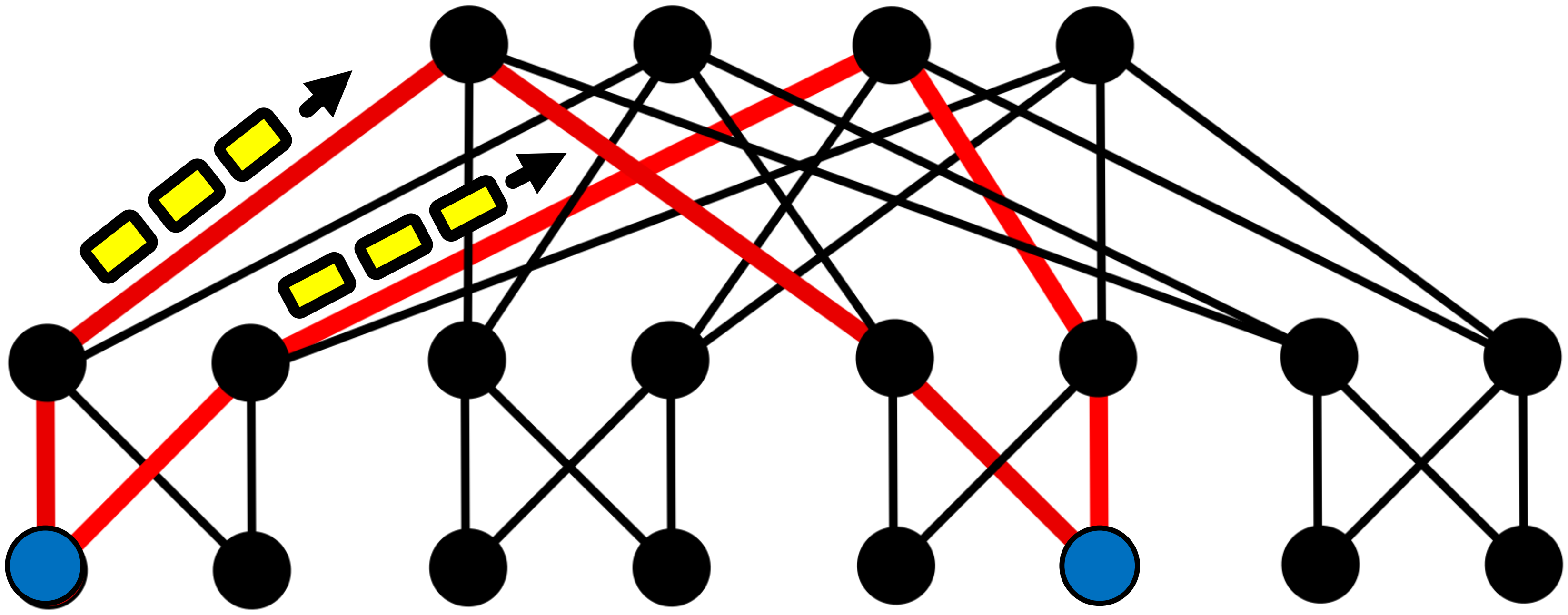
Routing in Fat Trees

High-performance routing is facilitated by **numerous multiple shortest paths of equal lengths** between any endpoints



Routing in Fat Trees

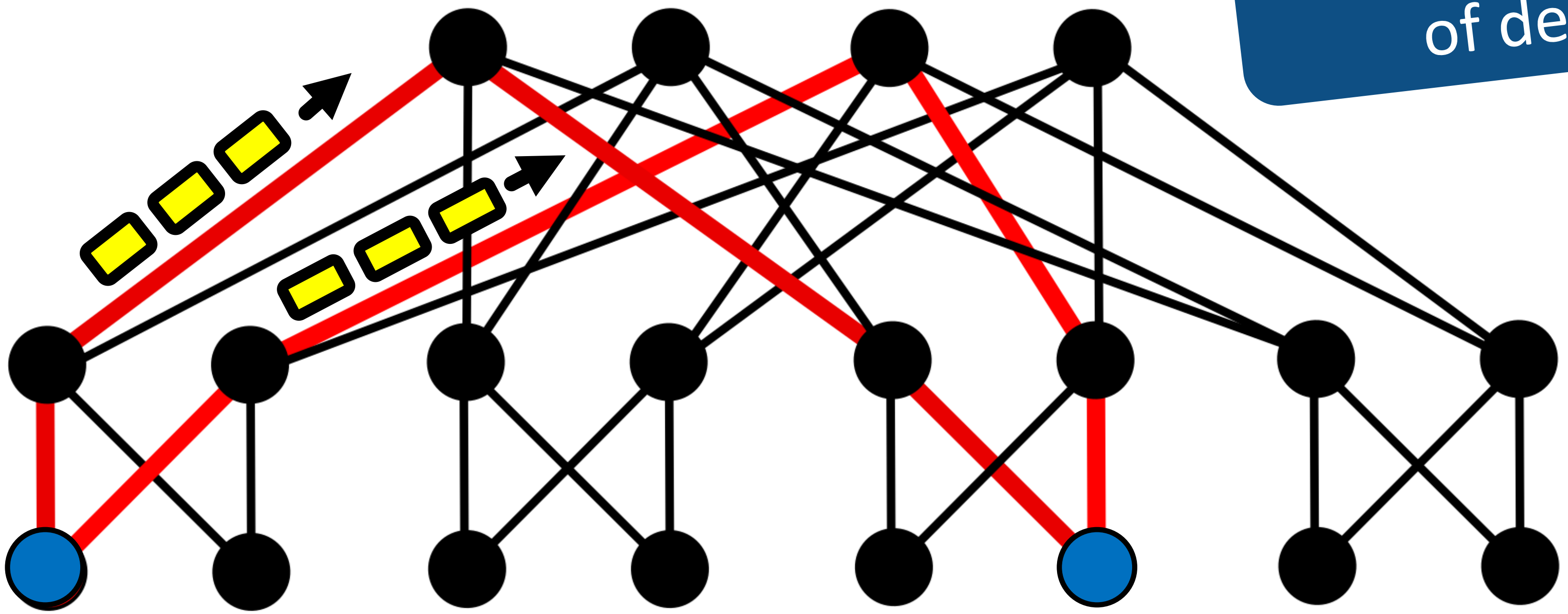
High-performance routing is facilitated by **numerous multiple shortest paths of equal lengths** between any endpoints



ROUTING IN FAT TREES

High-performance routing is facilitated by **numerous multiple shortest paths of equal lengths** between any endpoints

Established techniques for using multipathing & plethora of designs available



ROUTING IN FAT TREES

High-performance routing is facilitated by **numerous multiple shortest paths of equal lengths** between any endpoints

INFORMATIONAL

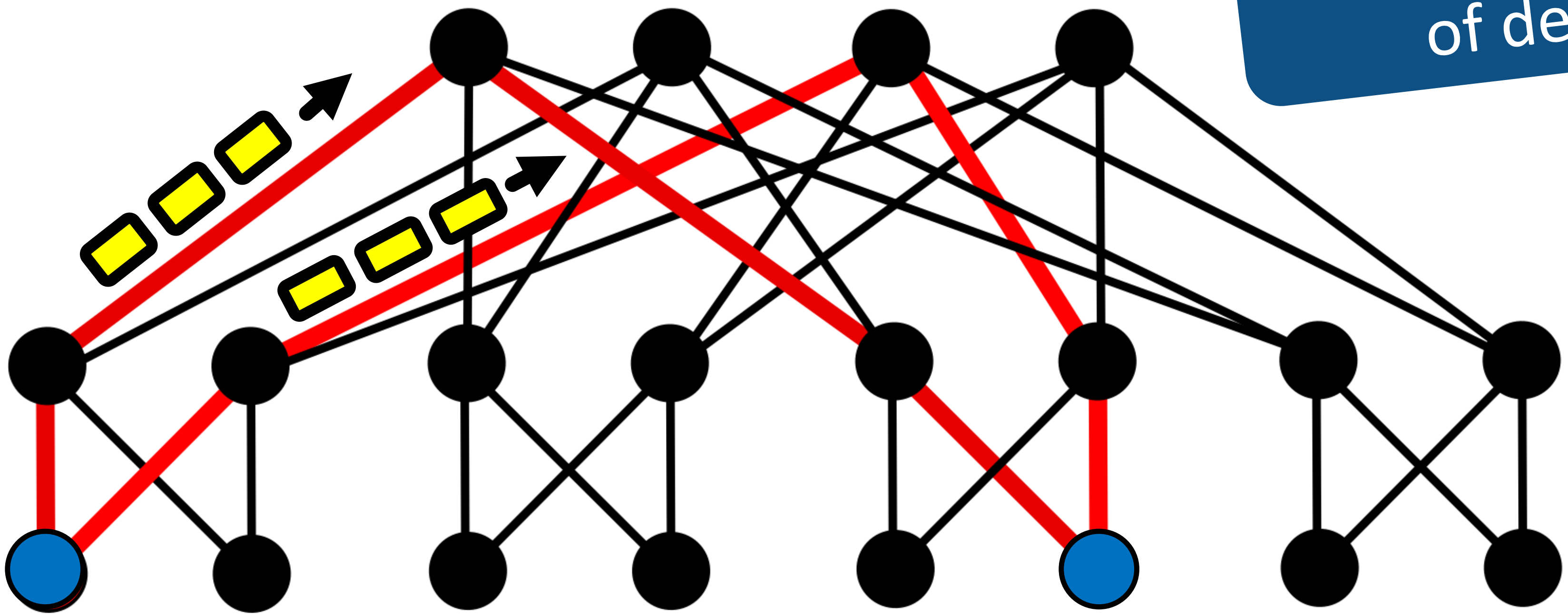
Network Working Group
Request for Comments: 2992
Category: Informational

ECMP

C. Hopps
NextHop Technologies
November 2000

Analysis of an Equal-Cost Multi-Path Algorithm

Established techniques for using multipathing & plethora of designs available



ROUTING IN FAT TREES

High-performance routing is facilitated by **numerous multiple shortest paths of equal lengths** between any endpoints

INFORMATIONAL

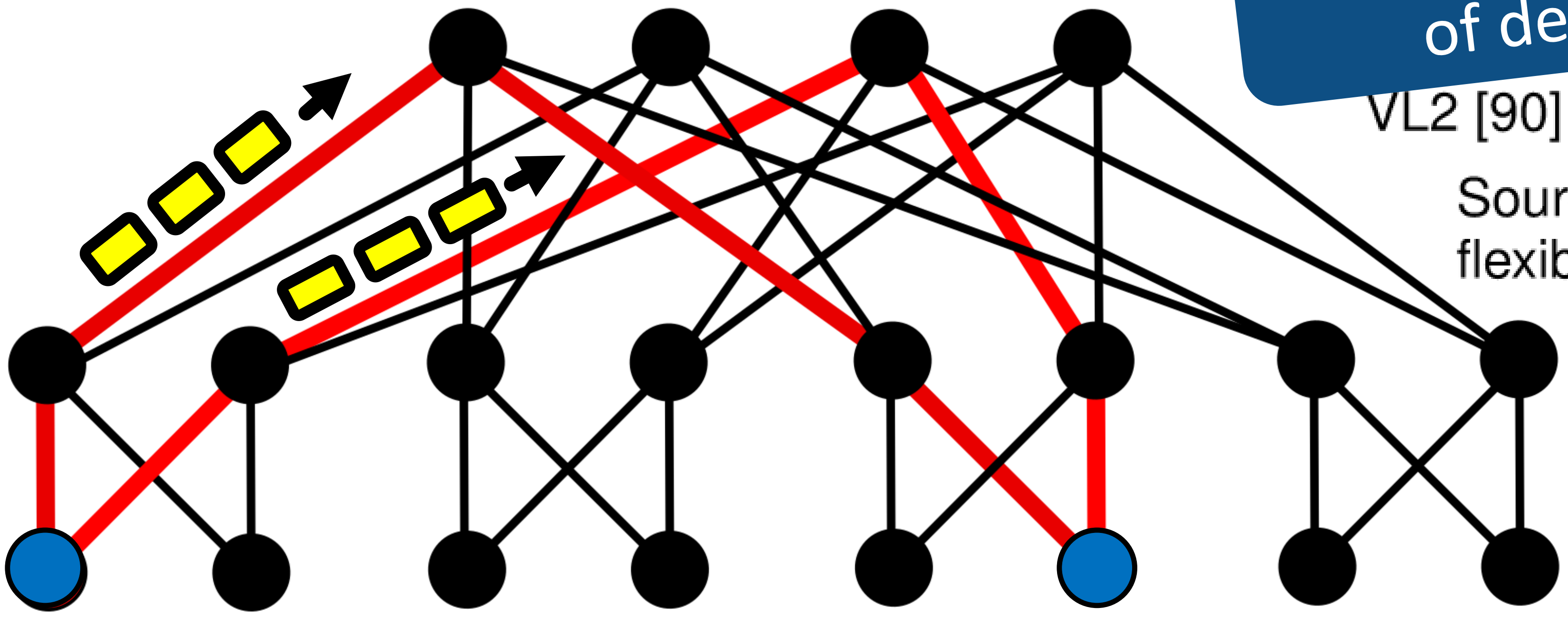
Network Working Group
Request for Comments: 2992
Category: Informational

C. Hopps
NextHop Technologies
November 2000

Analysis of an Equal-Cost Multi-Path Algorithm

ECMP

Established techniques for using multipathing & plethora of designs available



Work by Al-Fares et al. [7]

VL2 [90]

WCMP for DC [233]

Source routing for flexible DC fabric [117]

Monsoon [91]

PortLand [160]

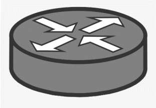
SPAIN [158]

ECMP-VLB [123]

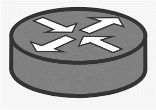
Work by Linden et al. [215]

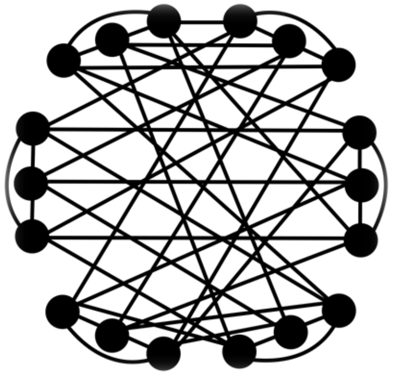
Work by Suchara et al. [204]

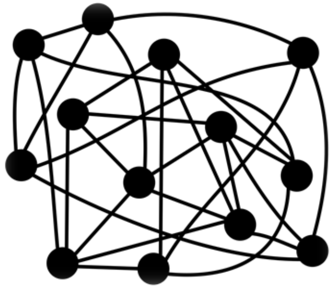
ROUTING IN LOW-DIAMETER TOPOLOGIES?

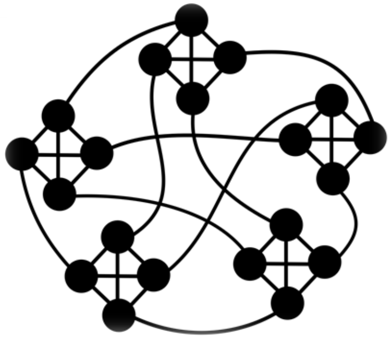


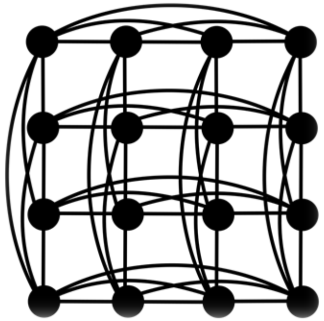
Routing in Low-Diameter Topologies?



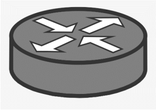


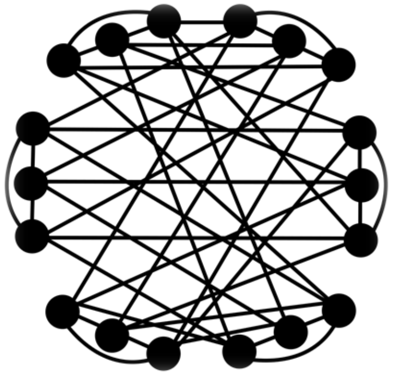


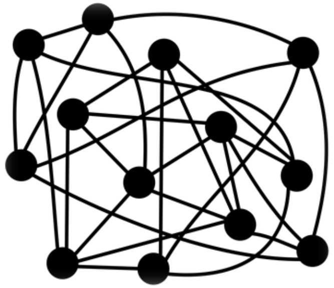




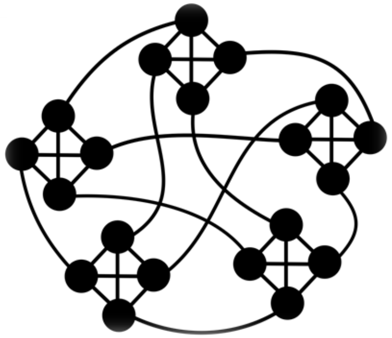
ROUTING IN LOW-DIAMETER TOPOLOGIES?

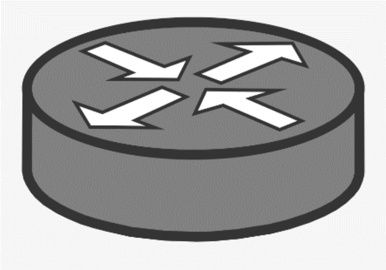


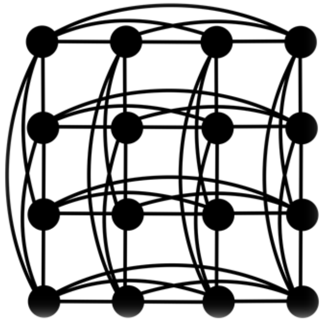




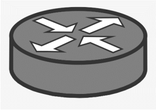
How to route modern low diameter topologies using multipathing?





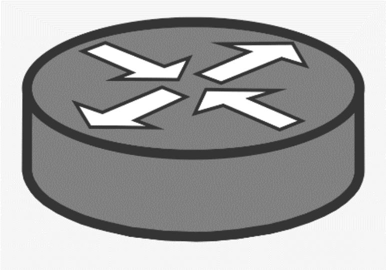


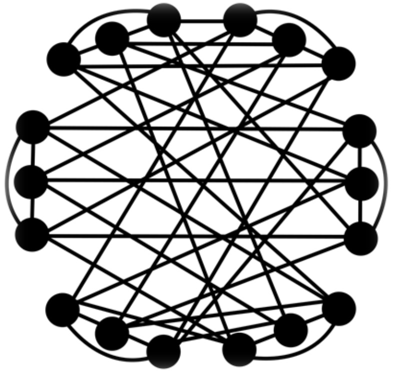
Routing in Low-Diameter Topologies?

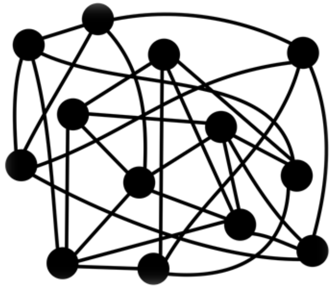


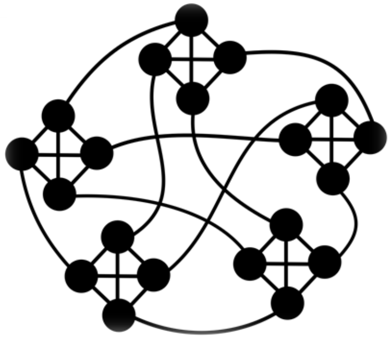
How to route modern low diameter topologies using multipathing?

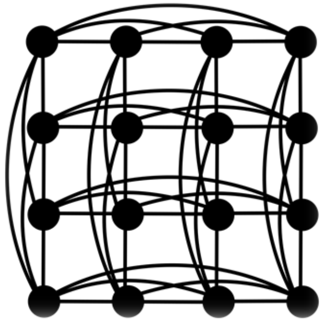
The driving question of this whole work



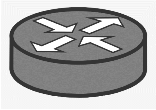






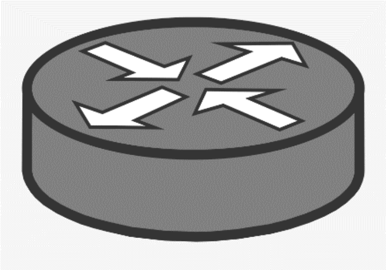


ROUTING IN LOW-DIAMETER TOPOLOGIES?

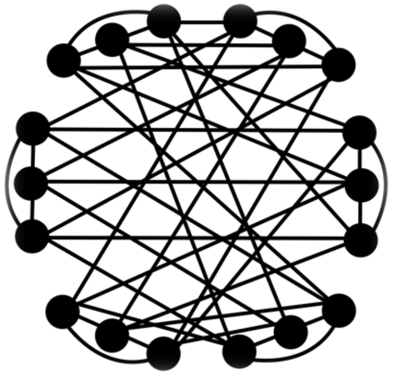


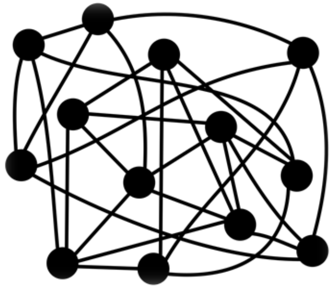
How to route modern low diameter topologies using multipathing?

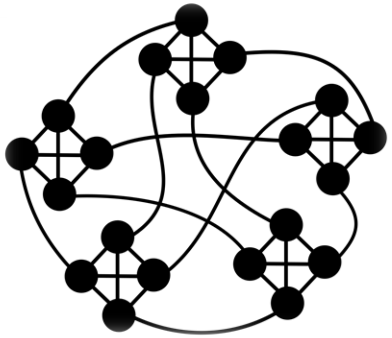
The driving question of this whole work

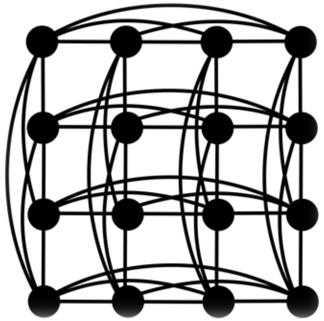


...but it touches on many aspects. Let's go!

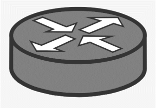








MULTIPATH ROUTING: MOTIVATION

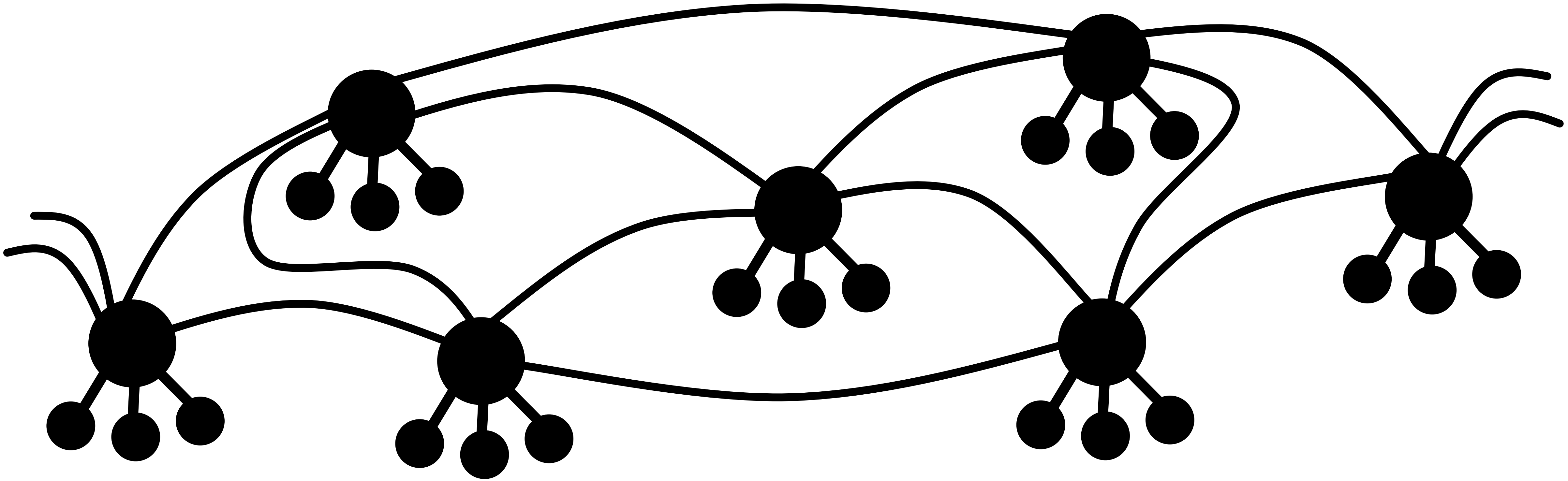


MULTIPATH ROUTING: MOTIVATION

What are the problems that we want to tackle with multipathing?

MULTIPATH ROUTING: MOTIVATION

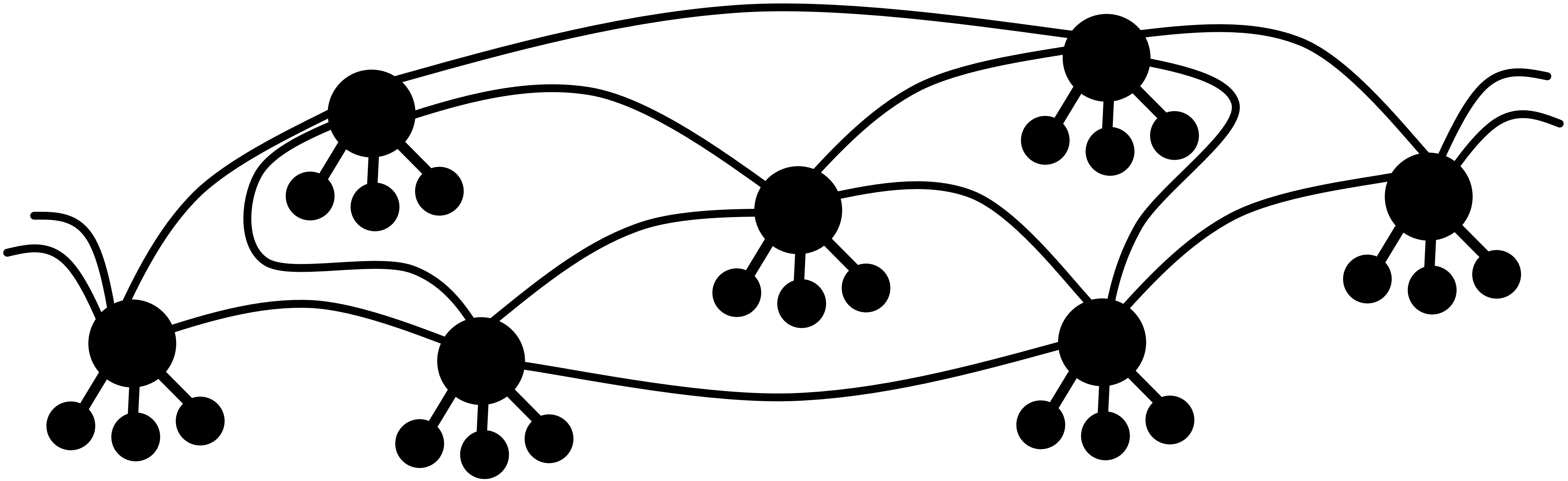
What are the problems that we want to tackle with multipathing?



MULTIPATH ROUTING: MOTIVATION

What are the problems that we want to tackle with multipathing?

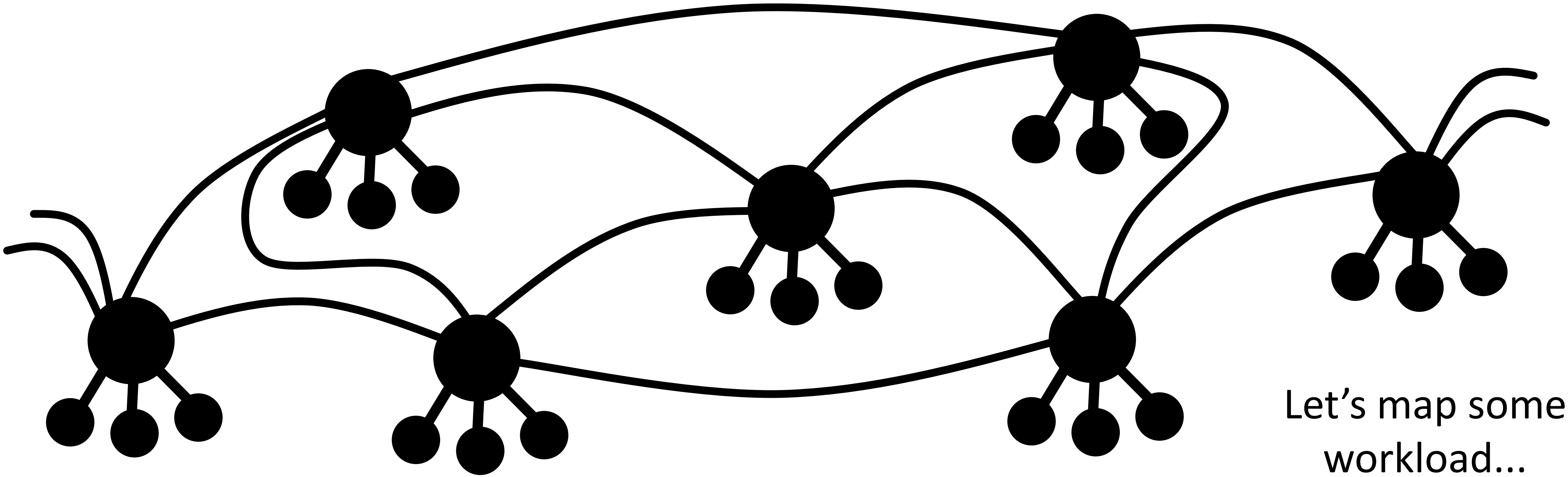
Flows collide!



MULTIPATH ROUTING: MOTIVATION

What are the problems that we want to tackle with multipathing?

Flows collide!

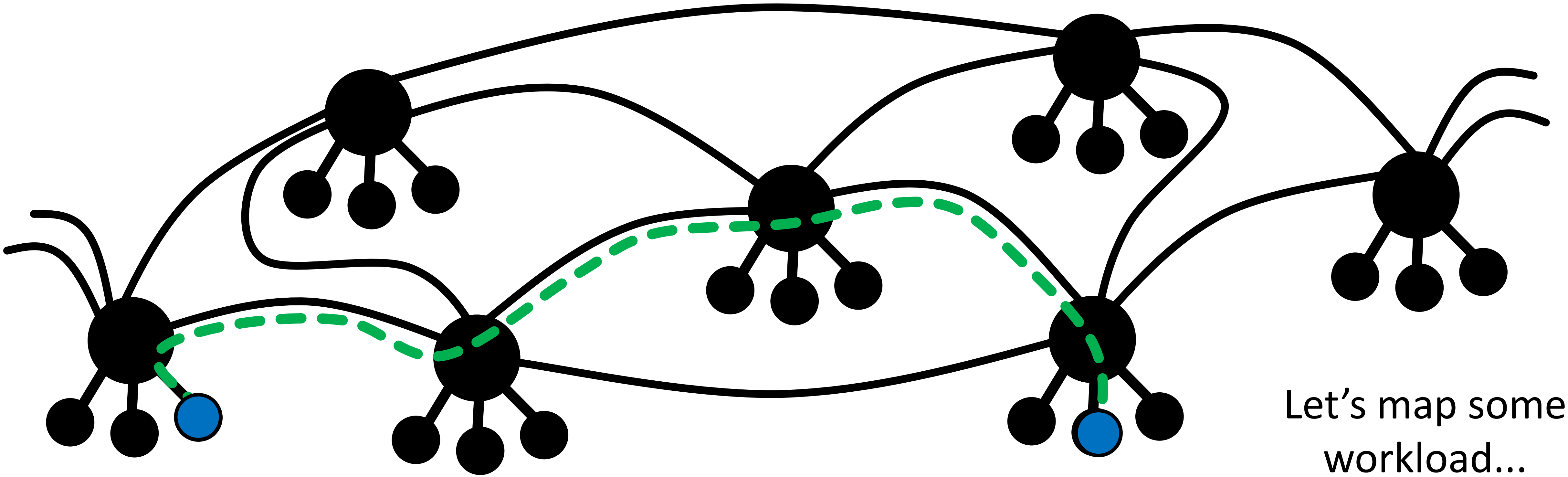


Let's map some workload...

MULTIPATH ROUTING: MOTIVATION

What are the problems that we want to tackle with multipathing?

Flows collide!

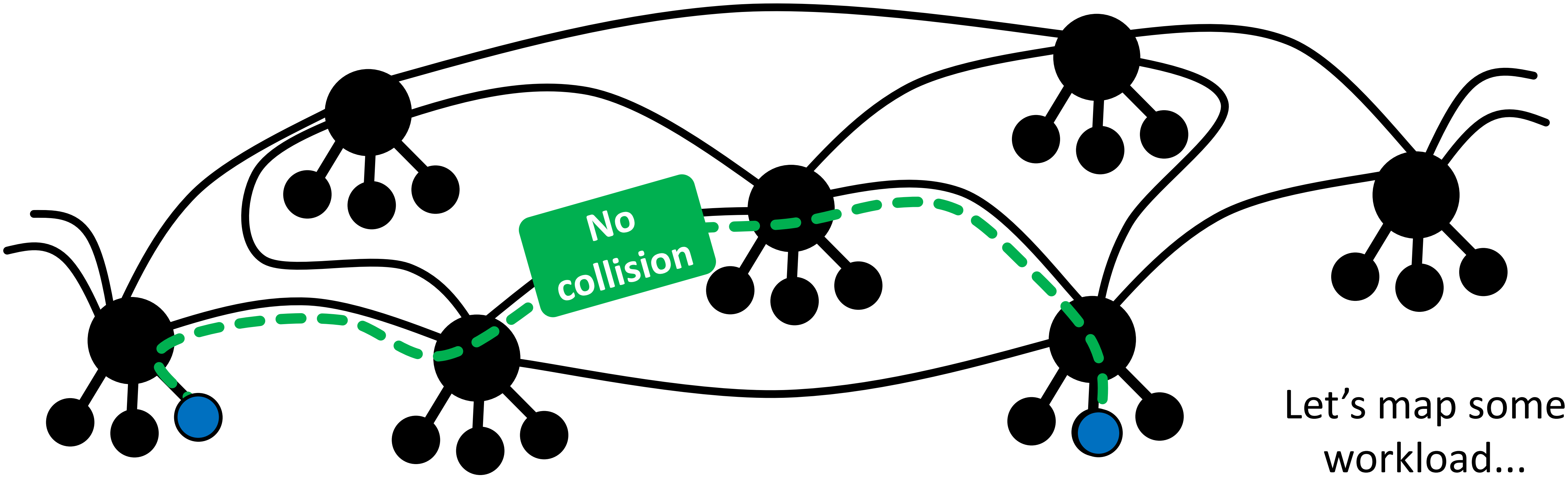


Let's map some workload...

MULTIPATH ROUTING: MOTIVATION

What are the problems that we want to tackle with multipathing?

Flows collide!

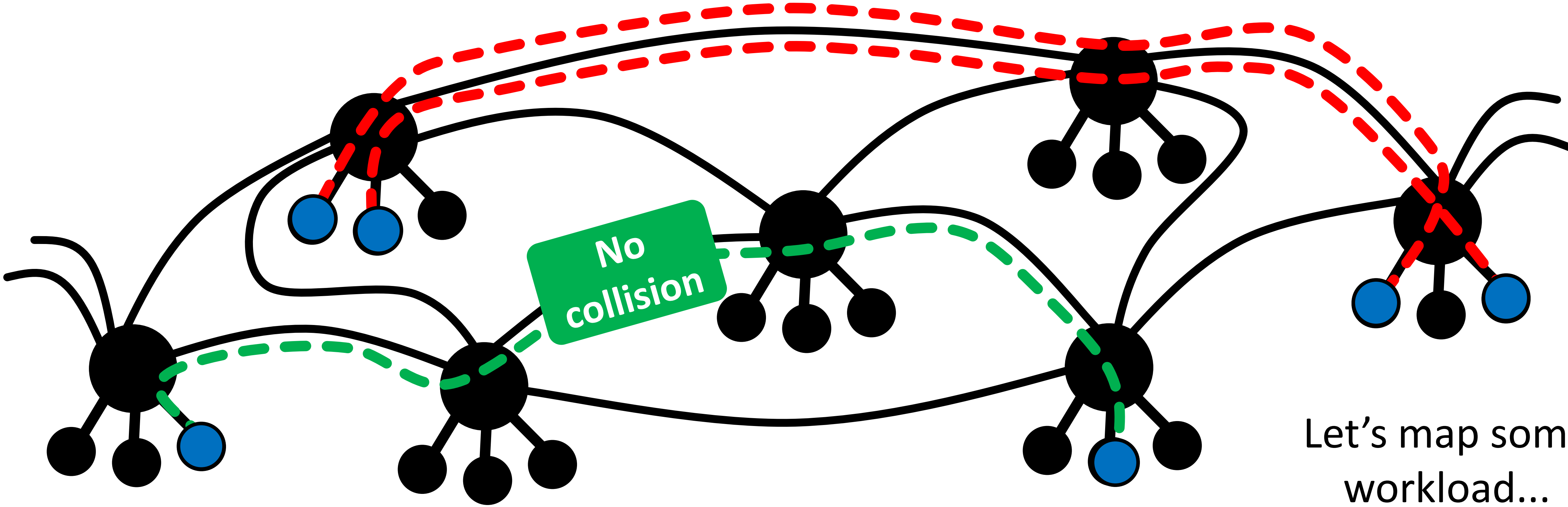


Let's map some workload...

Multipath Routing: Motivation

What are the problems that we want to tackle with multipathing?

Flows collide!

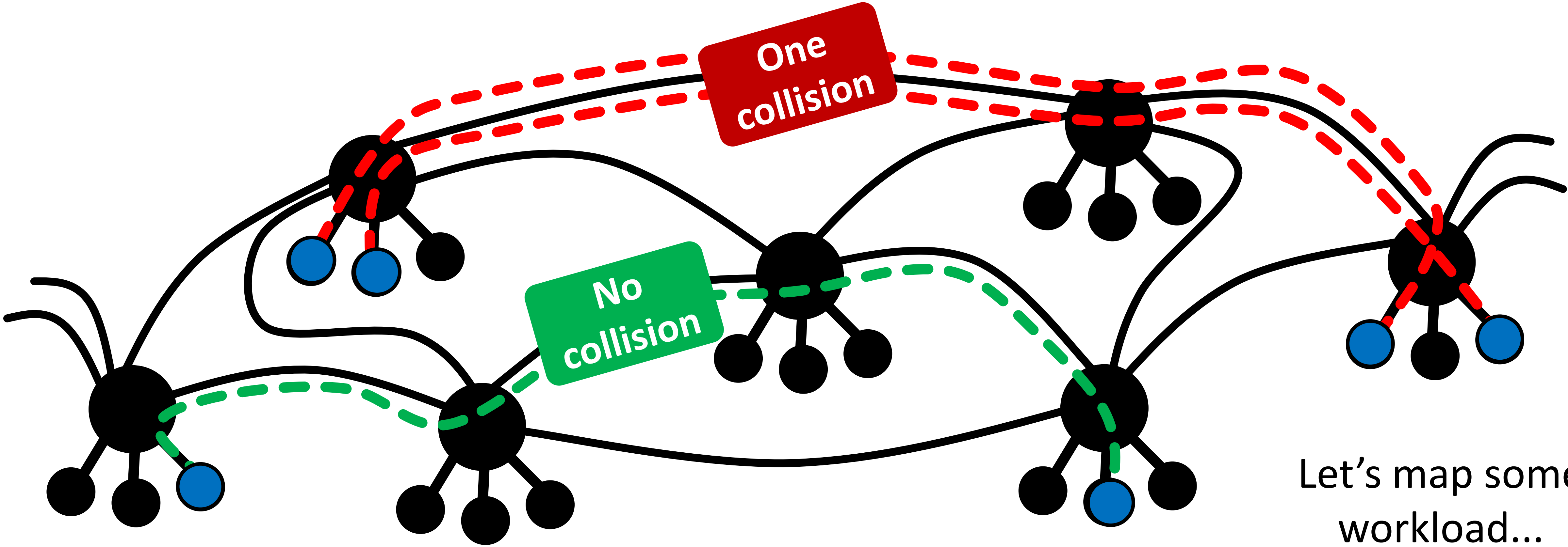


Let's map some workload...

MULTIPATH ROUTING: MOTIVATION

What are the problems that we want to tackle with multipathing?

Flows collide!

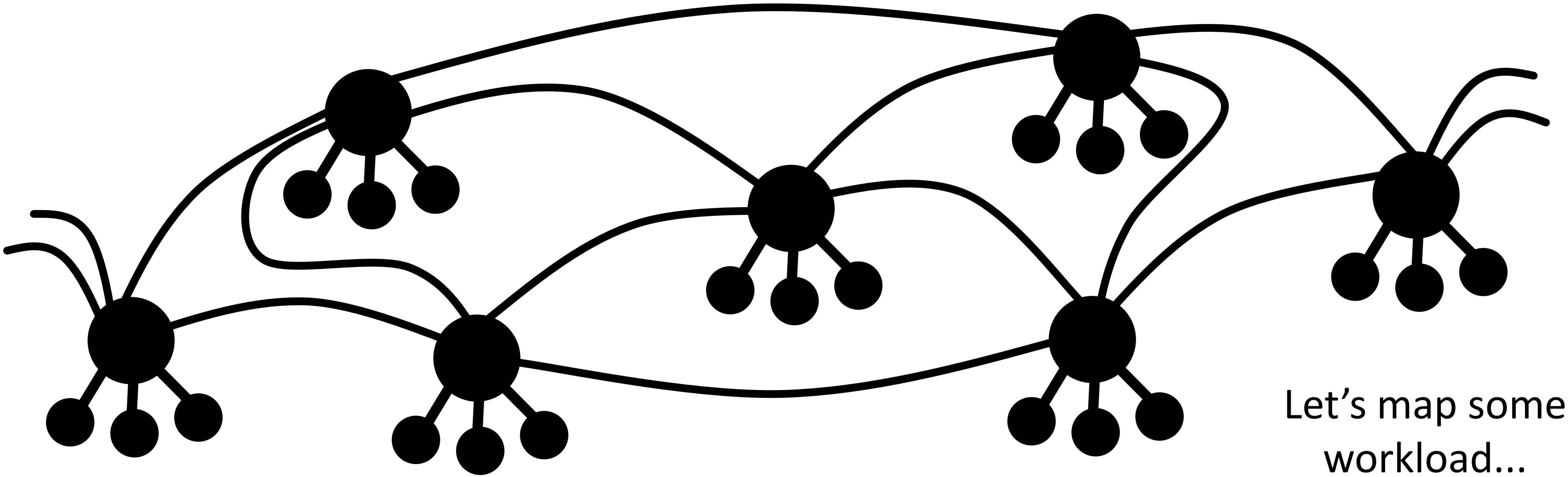


Let's map some workload...

MULTIPATH ROUTING: MOTIVATION

What are the problems that we want to tackle with multipathing?

Flows collide!

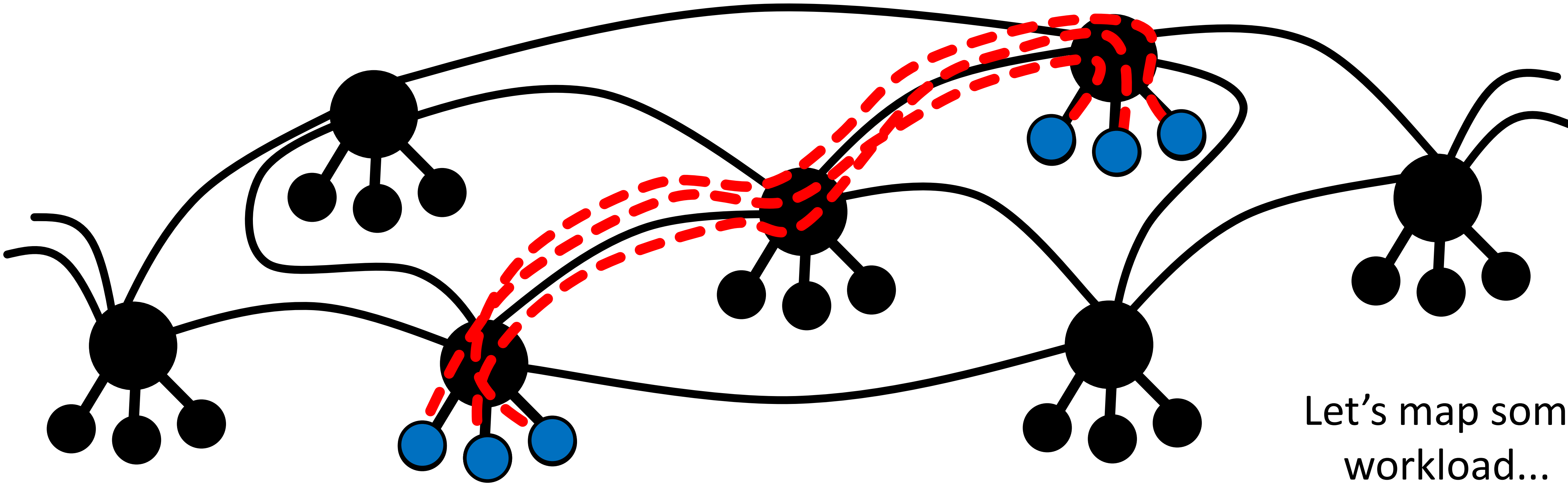


Let's map some workload...

MULTIPATH ROUTING: MOTIVATION

What are the problems that we want to tackle with multipathing?

Flows collide!

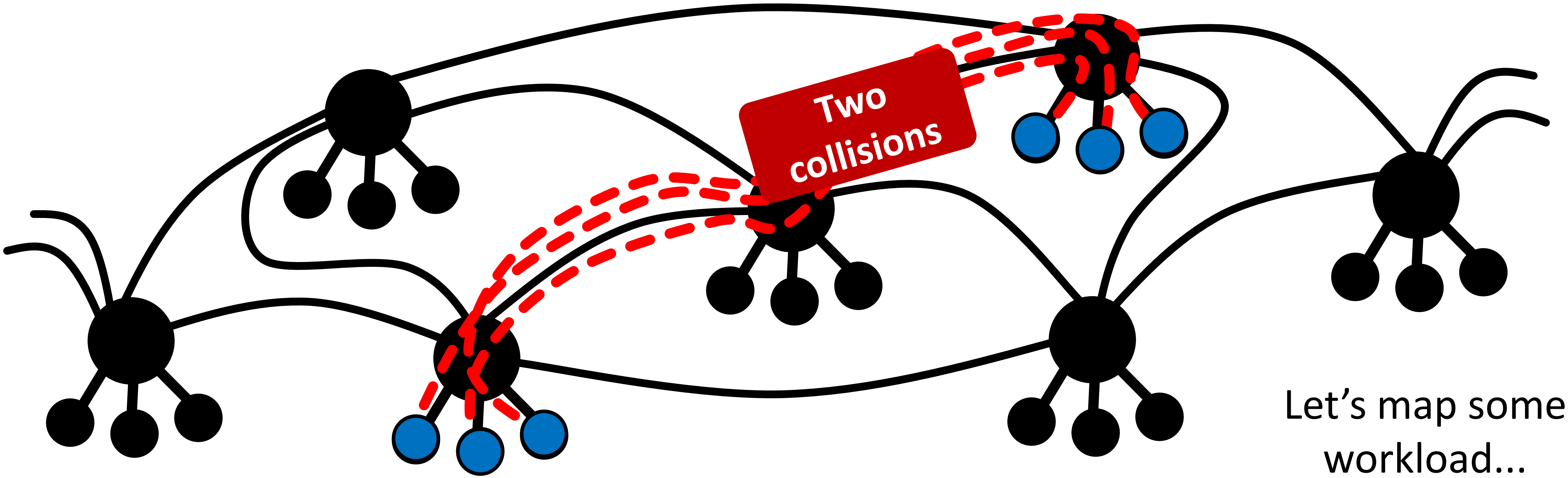


Let's map some workload...

MULTIPATH ROUTING: MOTIVATION

What are the problems that we want to tackle with multipathing?

Flows collide!

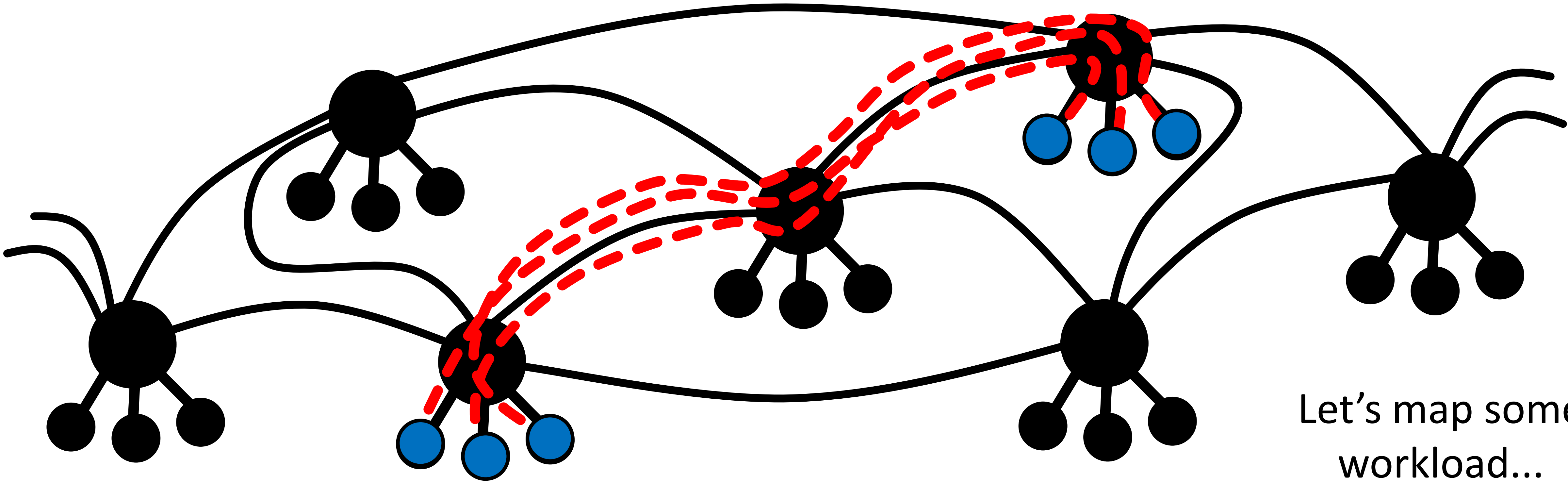


Let's map some workload...

MULTIPATH ROUTING: MOTIVATION

What are the problems that we want to tackle with multipathing?

Flows collide!



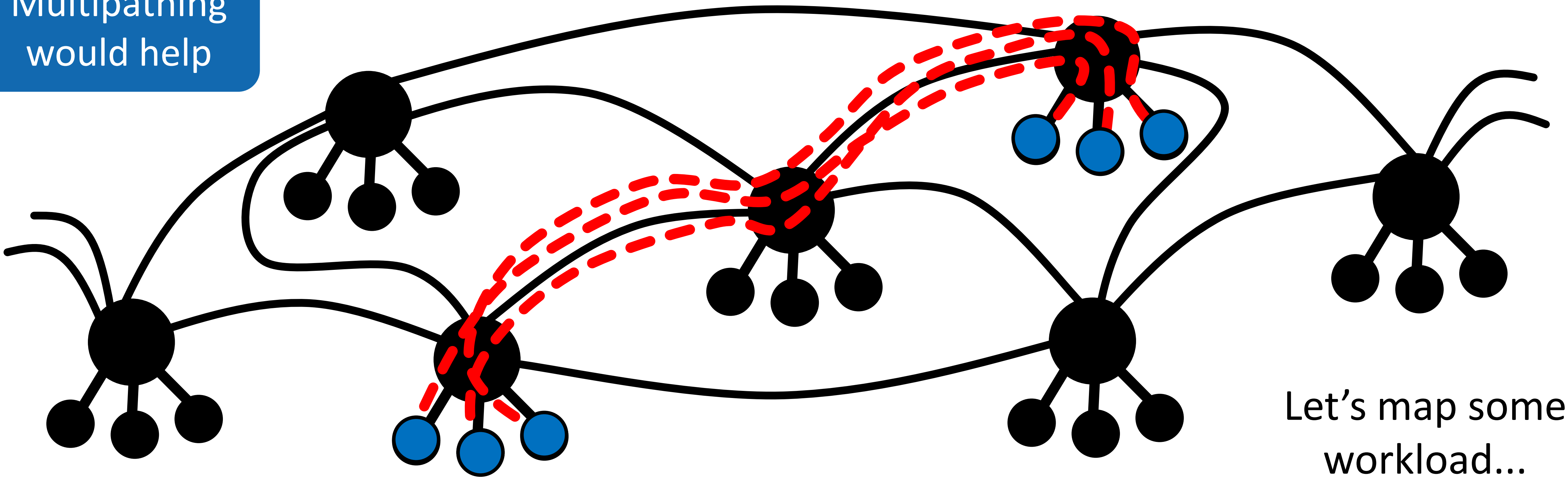
Let's map some workload...

MULTIPATH ROUTING: MOTIVATION

What are the problems that we want to tackle with multipathing?

Multipathing would help

Flows collide!



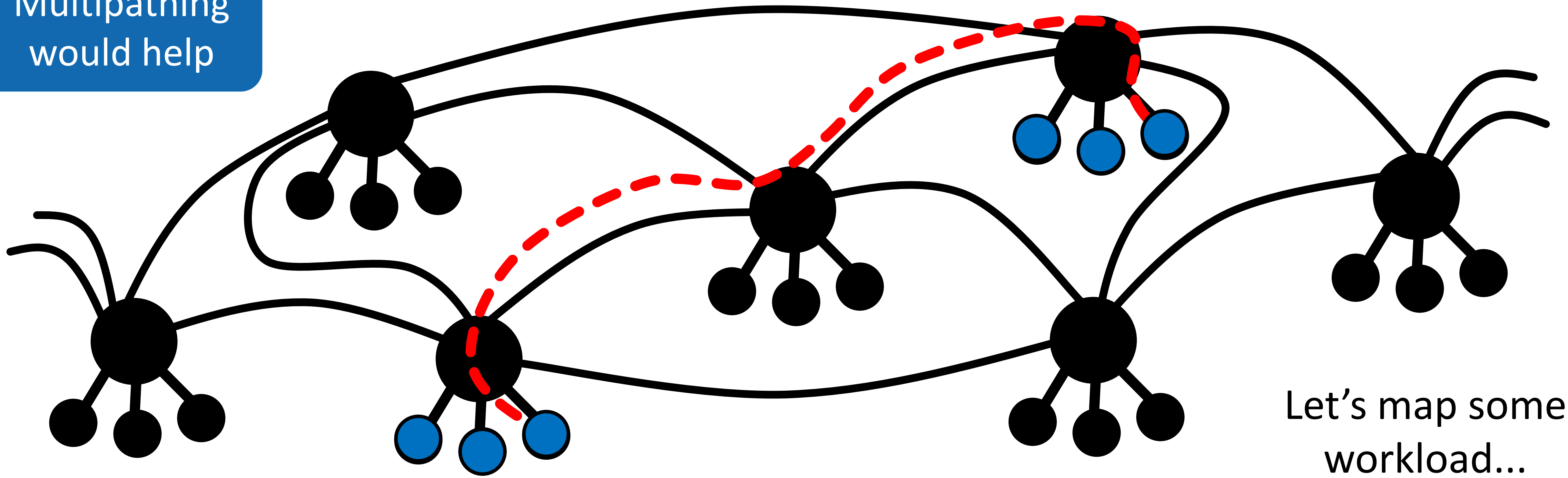
Let's map some workload...

Multipath Routing: Motivation

What are the problems that we want to tackle with multipathing?

Multipathing would help

Flows collide!



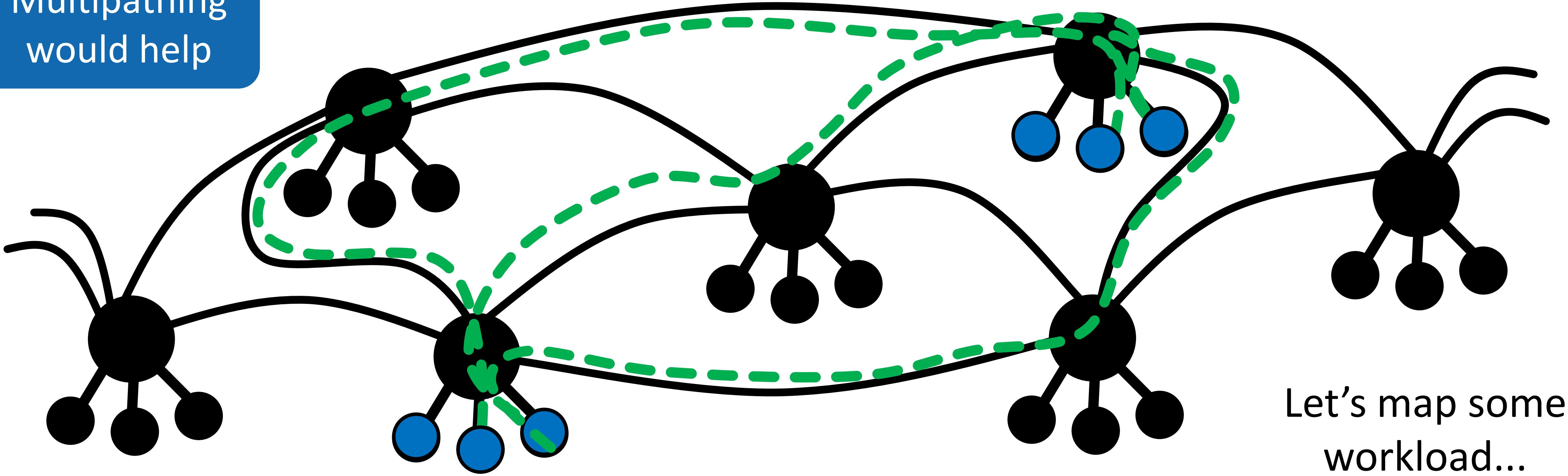
Let's map some workload...

Multipath Routing: Motivation

What are the problems that we want to tackle with multipathing?

Flows collide!

Multipathing would help



Let's map some workload...

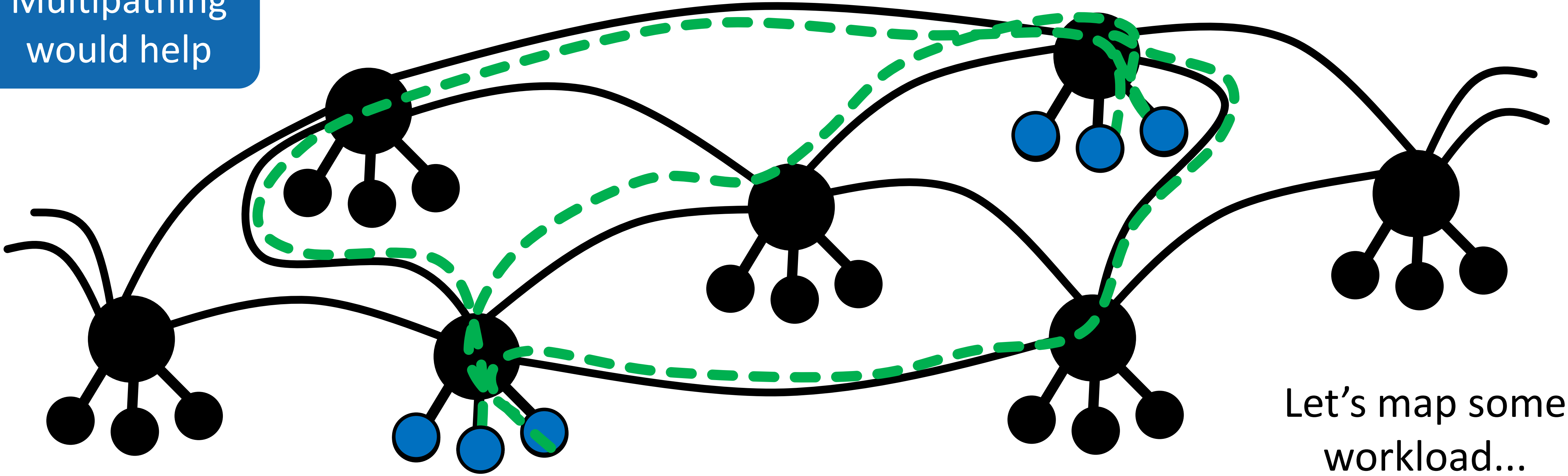
MULTIPATH ROUTING: MOTIVATION

What are the problems that we want to tackle with multipathing?

Multipathing would help

Flows collide!

How many paths (in the network) do we need to „accommodate" collisions?



Let's map some workload...

MULTIPATH ROUTING: MOTIVATION

What are the problems that we want to tackle with multipathing?

Multipathing would help

Flows collide!

How many paths (in the network) do we need to „accommodate” collisions?

Let’s map some workload...

MULTIPATH ROUTING: MOTIVATION

What are the problems that we want to tackle with multipathing?

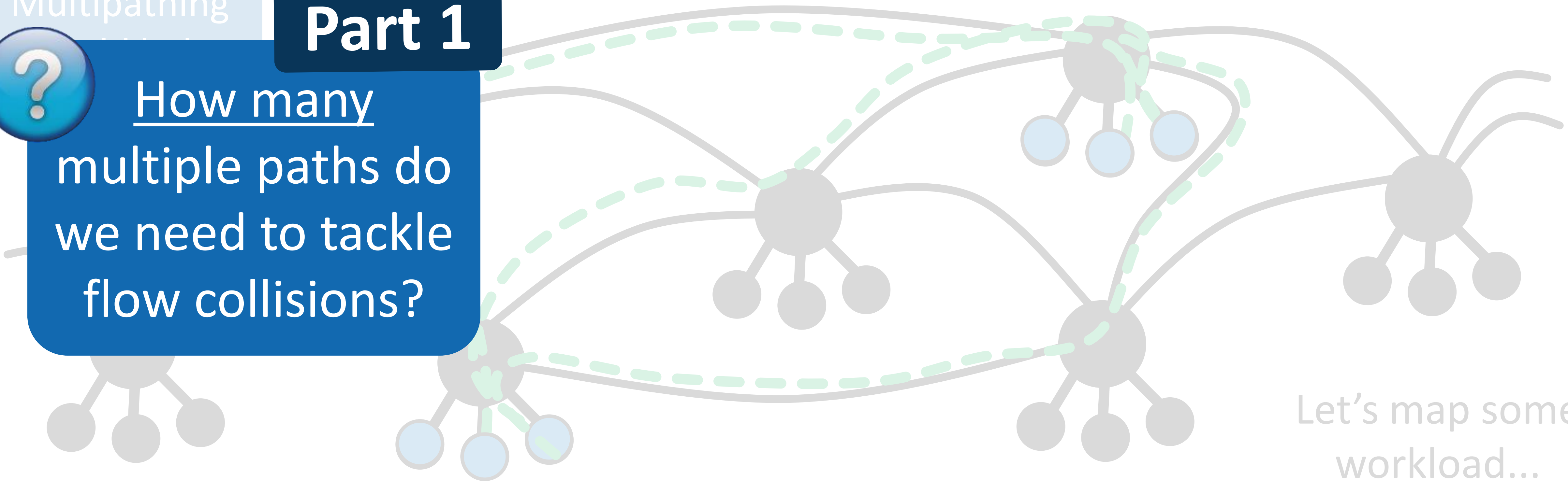
Flows collide!

How many paths (in the network) do we need to „accommodate" collisions?

Multipathing

Part 1

How many multiple paths do we need to tackle flow collisions?



Let's map some workload...

MULTIPATH ROUTING: MOTIVATION

What are the problems that we want to tackle with multipathing?

Flows collide!

How many paths (in the network) do we need to „accommodate” collisions?

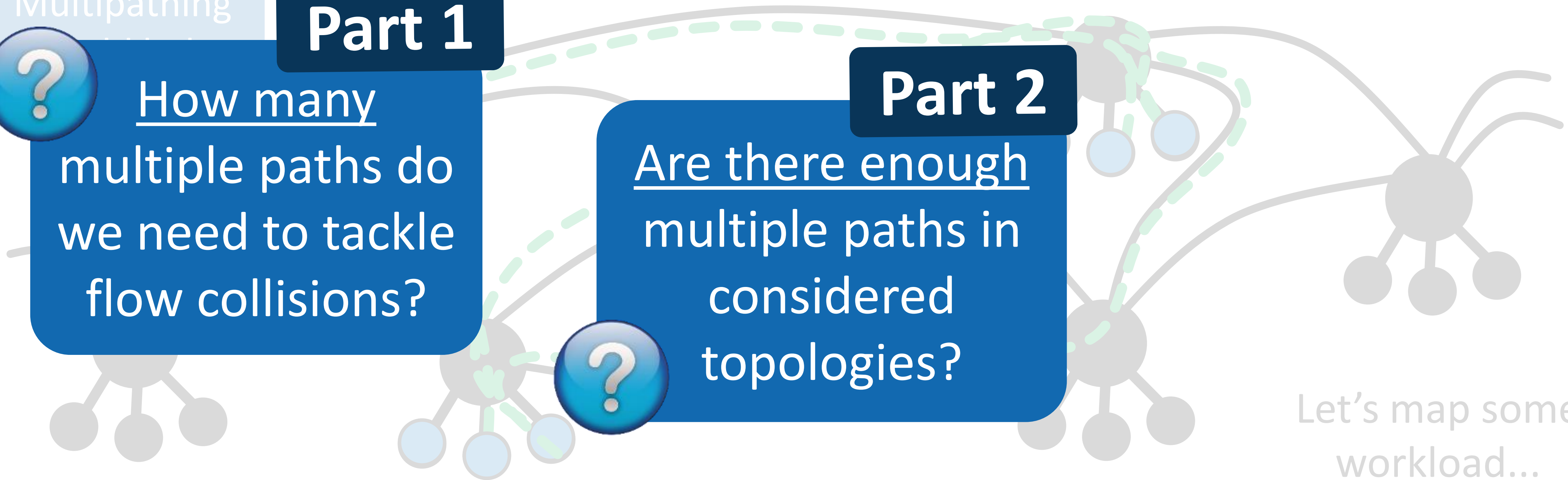
Multipathing

Part 1

How many multiple paths do we need to tackle flow collisions?

Part 2

Are there enough multiple paths in considered topologies?



Let’s map some workload...

MULTIPATH ROUTING: MOTIVATION

What are the problems that we want to tackle with multipathing?

Flows collide!

How many paths (in the network) do we need to „accommodate" collisions?

Multipathing

Part 1

How many multiple paths do we need to tackle flow collisions?

Part 2

Are there enough multiple paths in considered topologies?

Part 3

How to use such multiple paths in considered topologies?

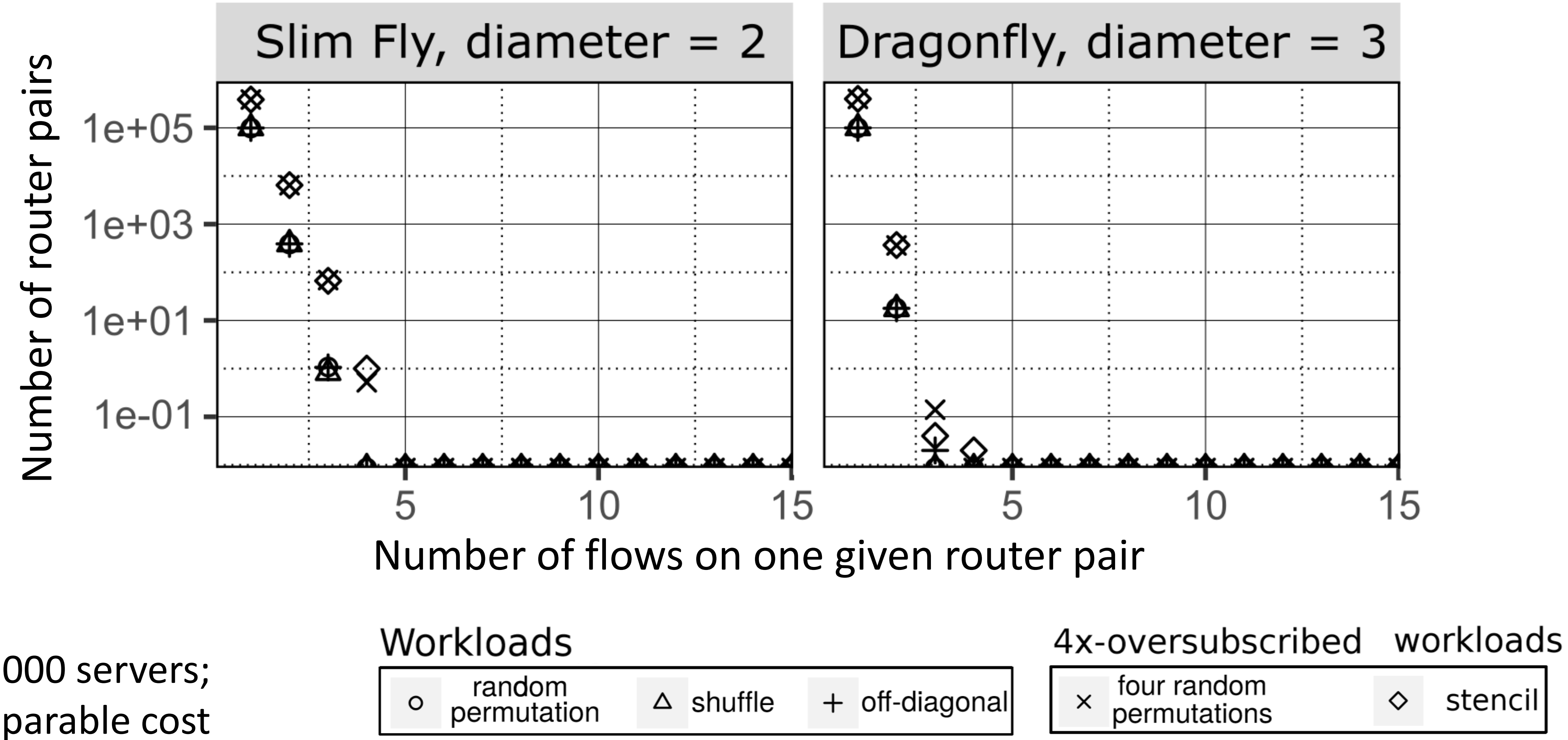
...ne workload...

How many multiple paths do we need to tackle flow collisions?

Part 1

How many multiple paths do we need to tackle flow collisions?

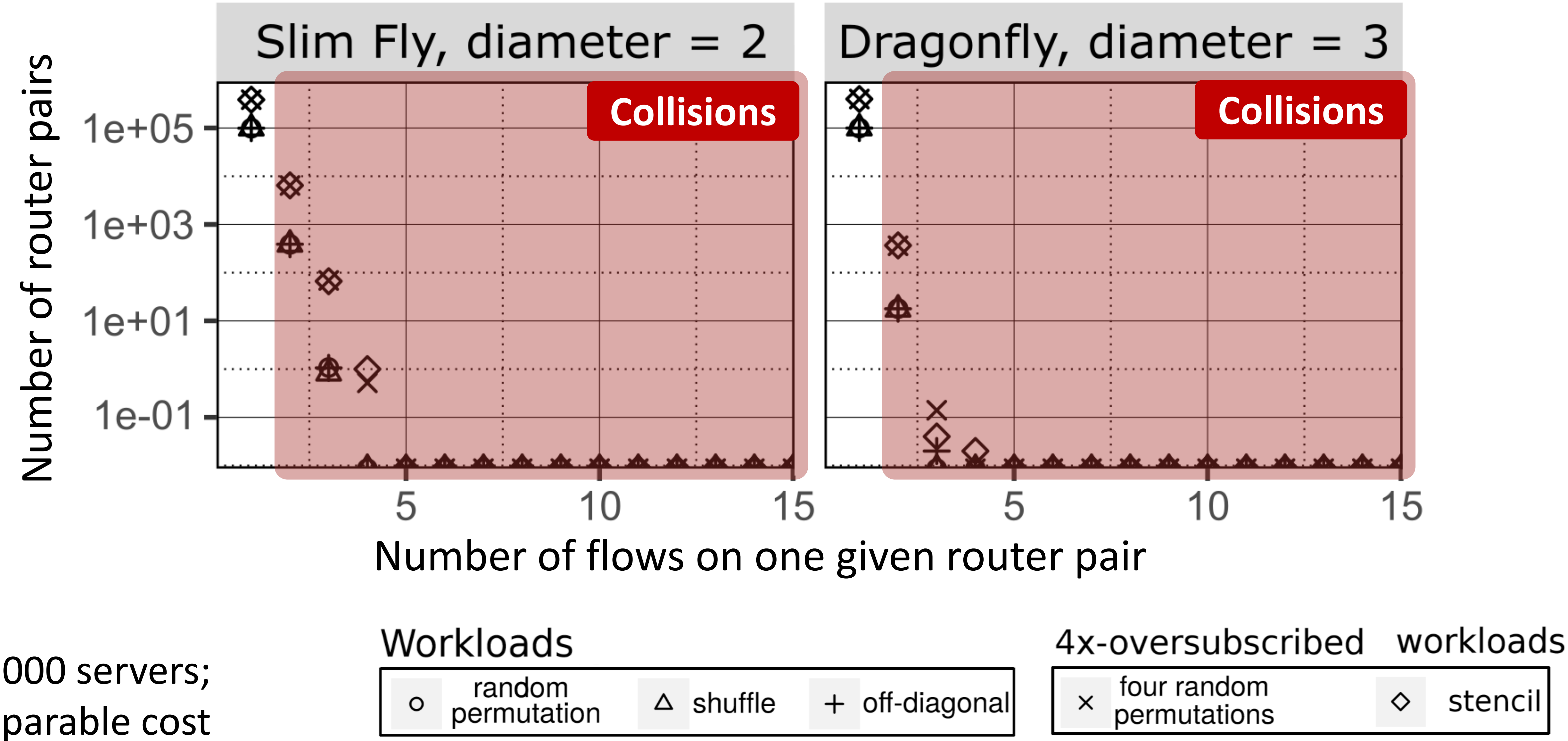
Part 1



≈10,000 servers; comparable cost

How many multiple paths do we need to tackle flow collisions?

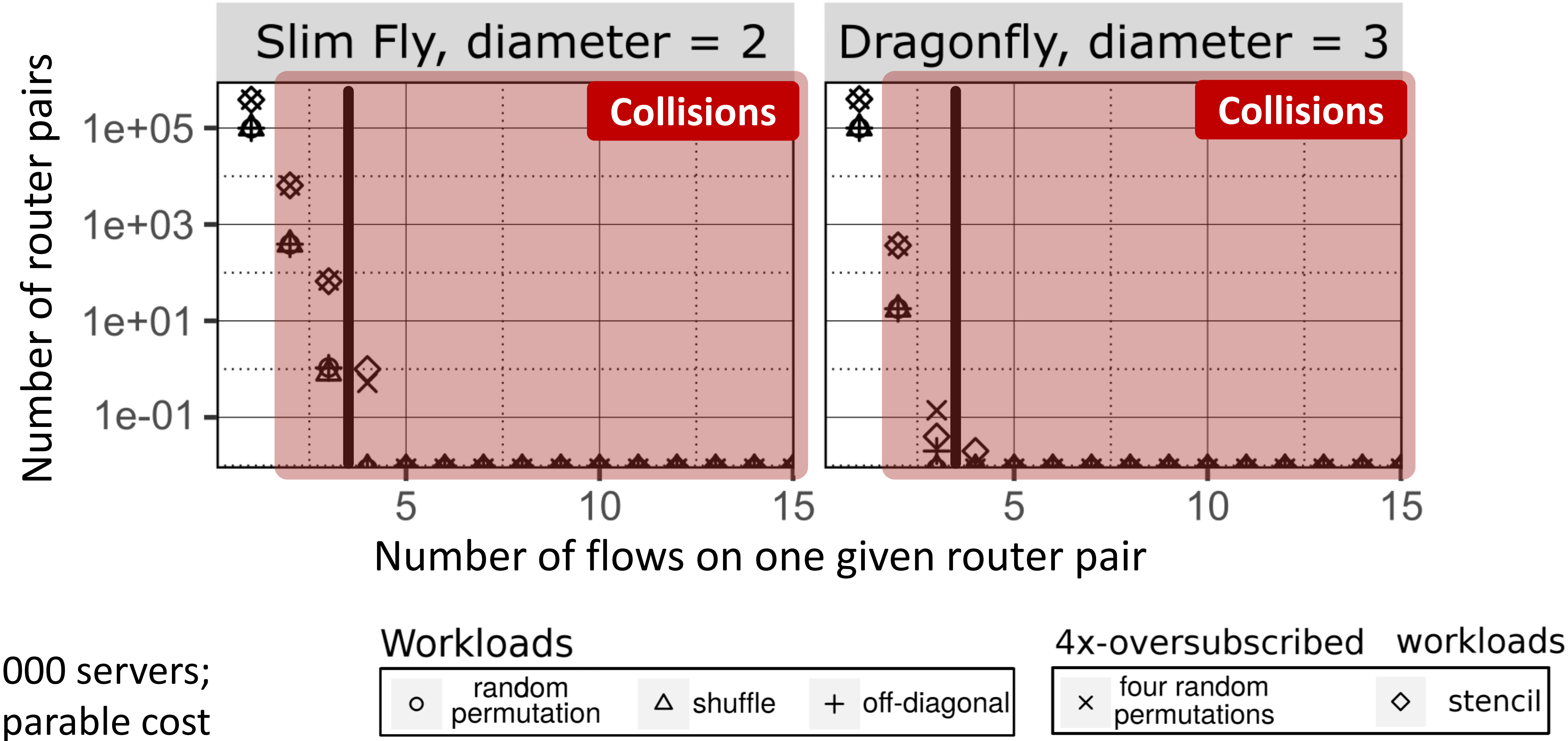
Part 1



≈10,000 servers; comparable cost

HOW MANY MULTIPLE PATHS DO WE NEED TO TACKLE FLOW COLLISIONS?

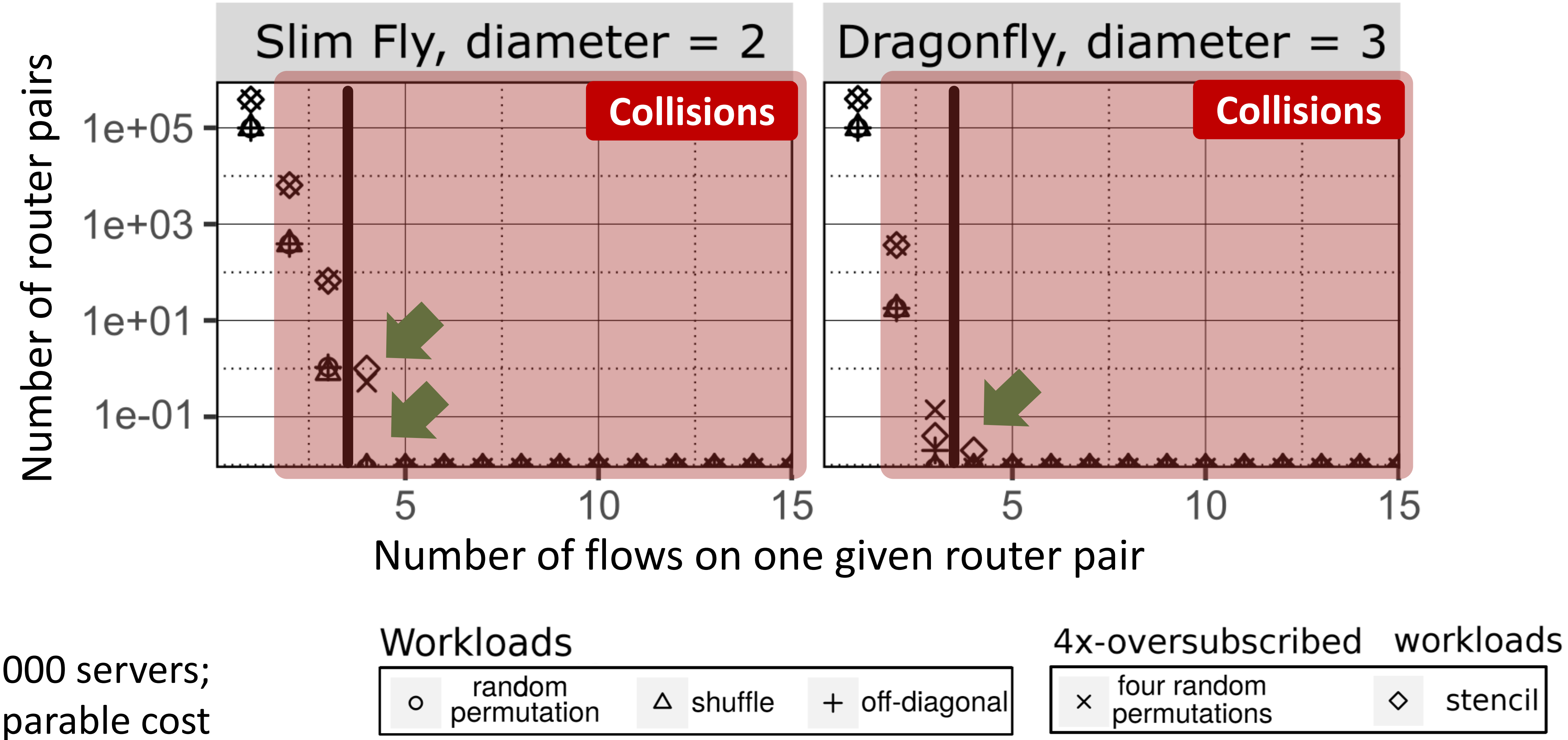
Part 1



≈10,000 servers; comparable cost

How many multiple paths do we need to tackle flow collisions?

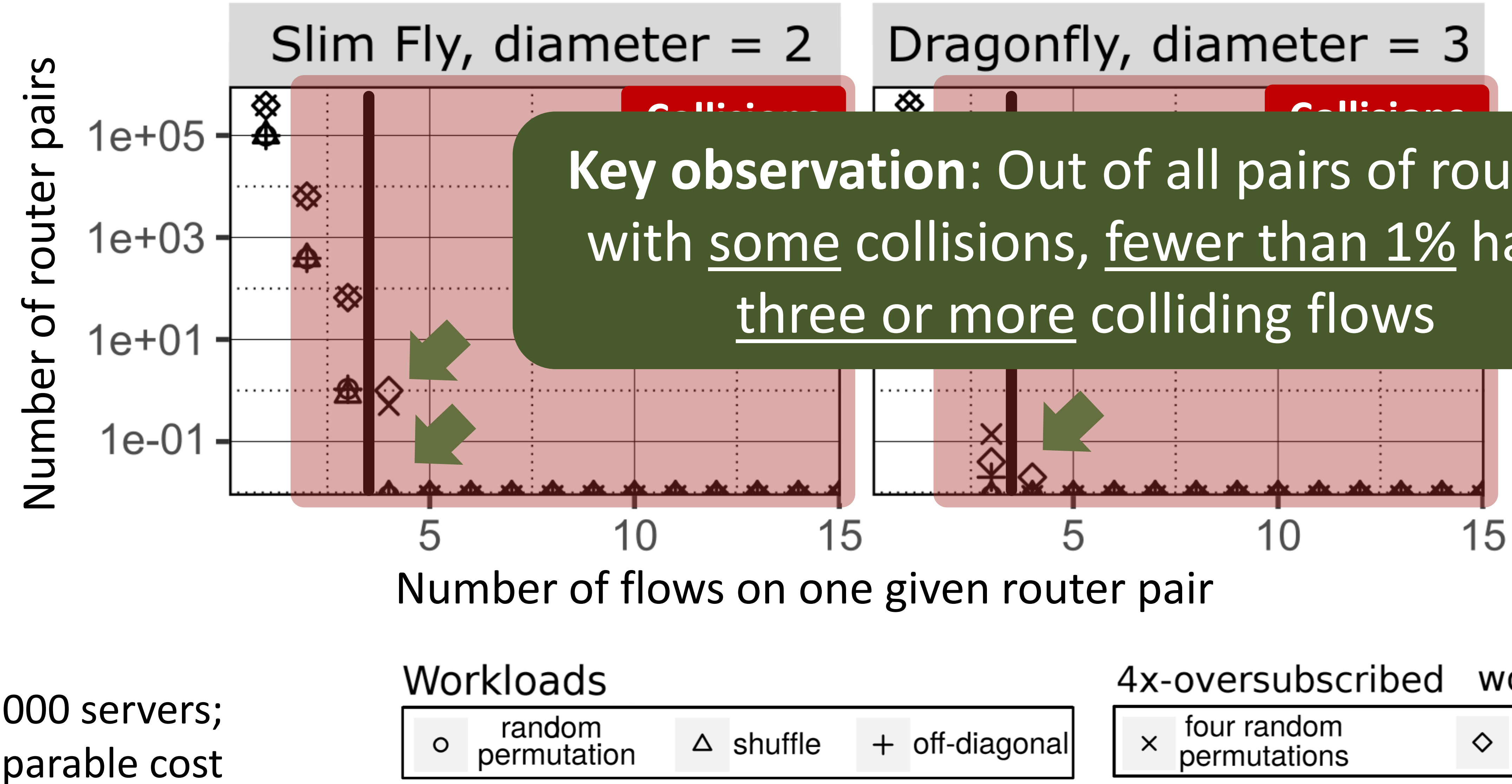
Part 1



≈10,000 servers; comparable cost

HOW MANY MULTIPLE PATHS DO WE NEED TO TACKLE FLOW COLLISIONS?

Part 1

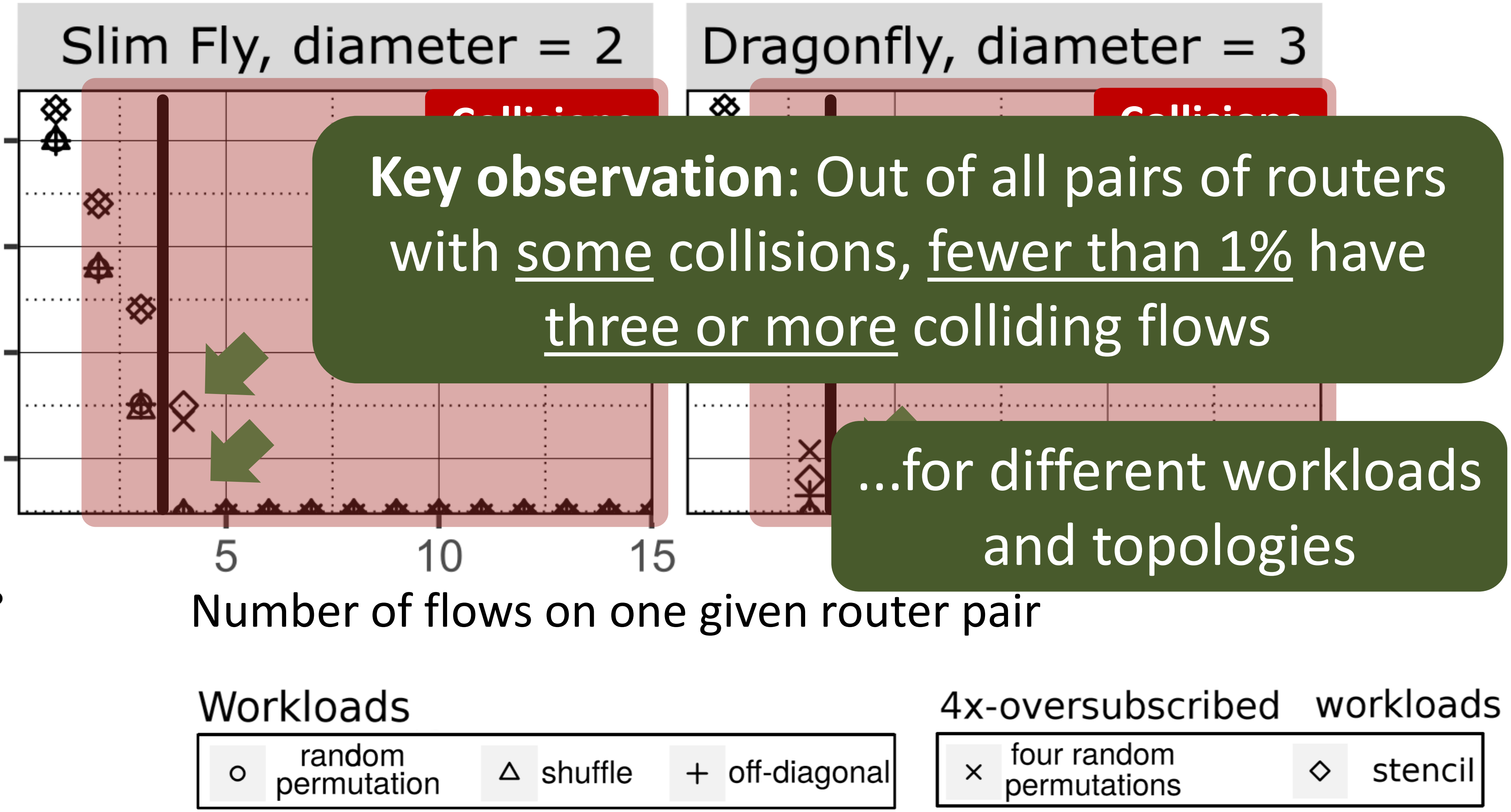


Key observation: Out of all pairs of routers with <u>some</u> collisions, <u>fewer than 1%</u> have <u>three or more</u> colliding flows

≈10,000 servers; comparable cost

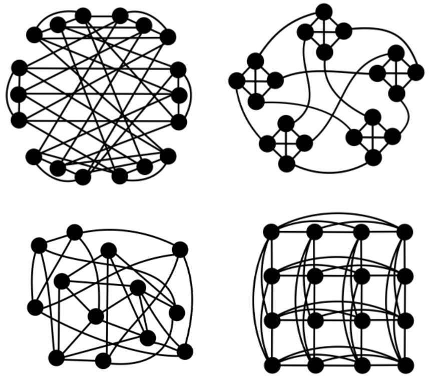
How many multiple paths do we need to tackle flow collisions?

Part 1



Key observation: Out of all pairs of routers with some collisions, fewer than 1% have three or more colliding flows

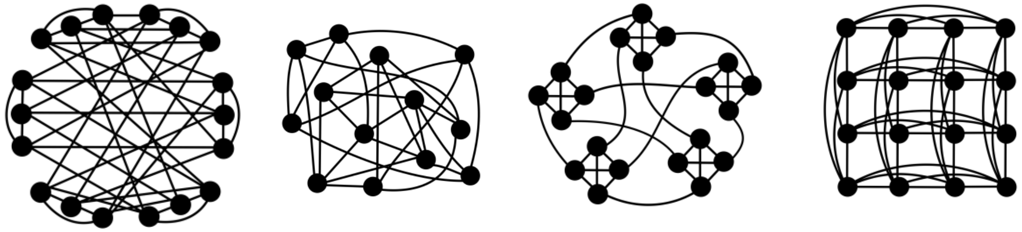
...for different workloads and topologies



How many multiple paths do we need to tackle flow collisions?

Part 1

...For different topologies



...For different workloads

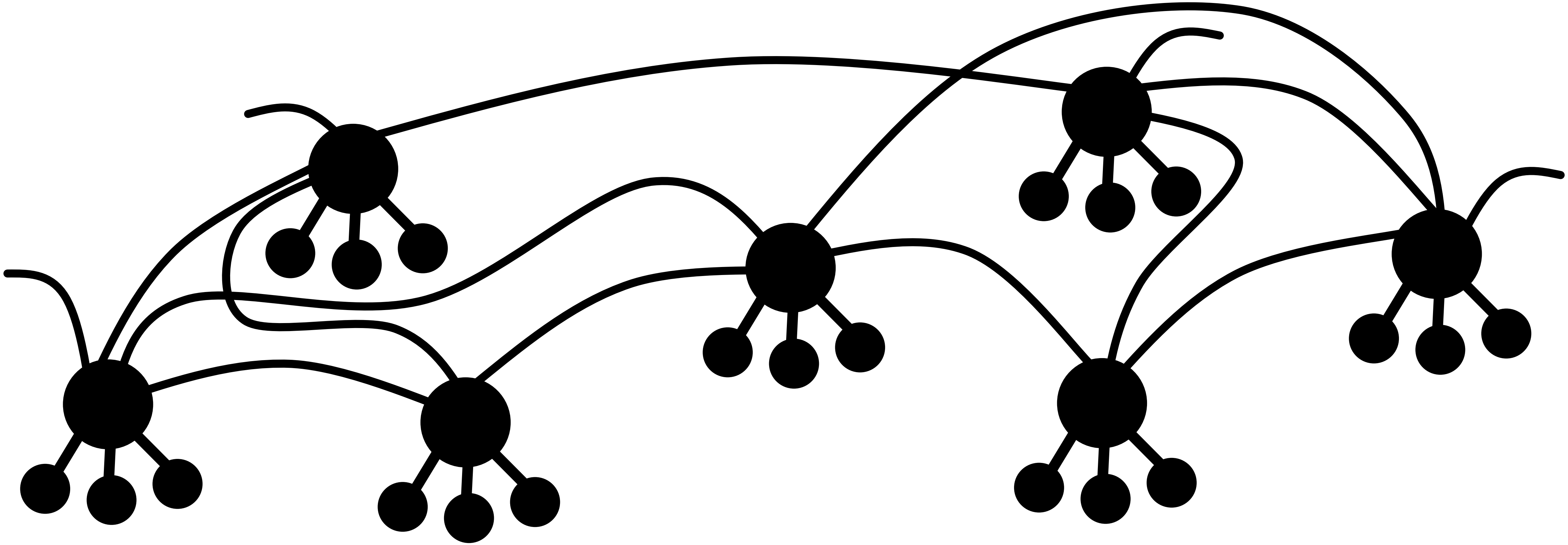
Random uniform

Off-diagonal

Stencil

Shuffle

...



How many multiple paths do we need to tackle flow collisions?

Part 1

Key Insight: We need **three** disjoint paths per router pair to handle [almost all] colliding flows

.For different workloads

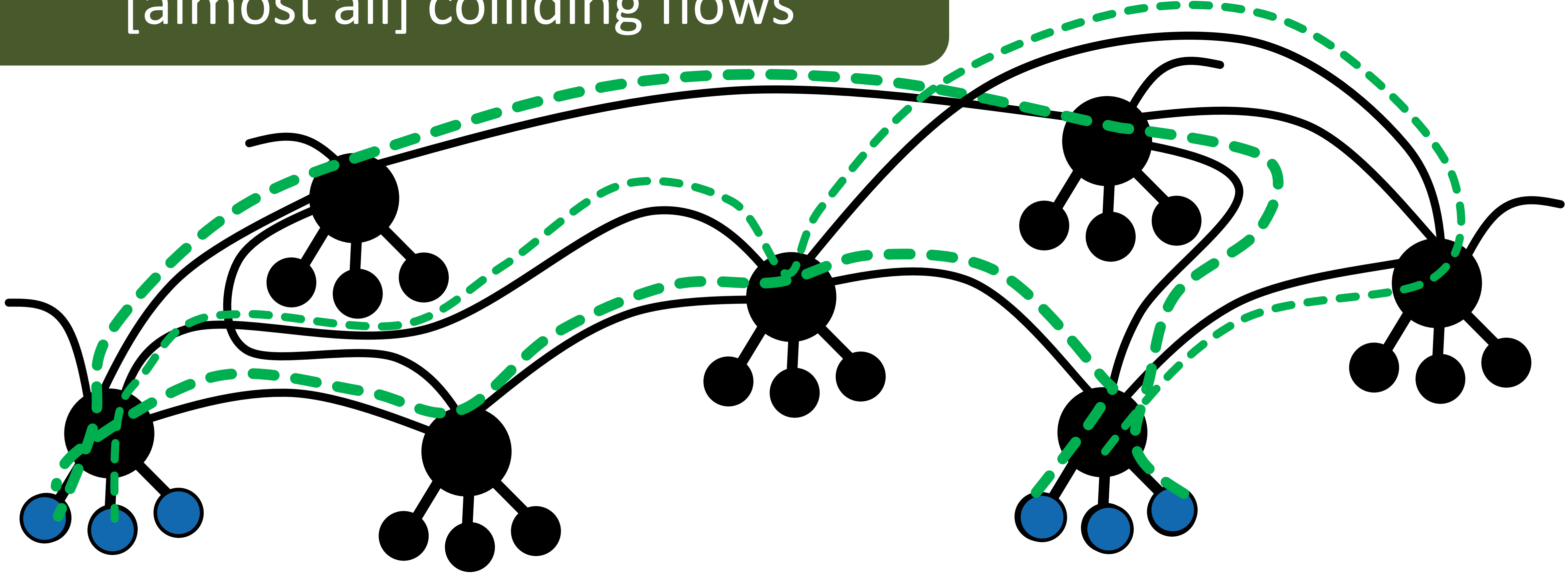
Random uniform

Off-diagonal

Stencil

Shuffle

...



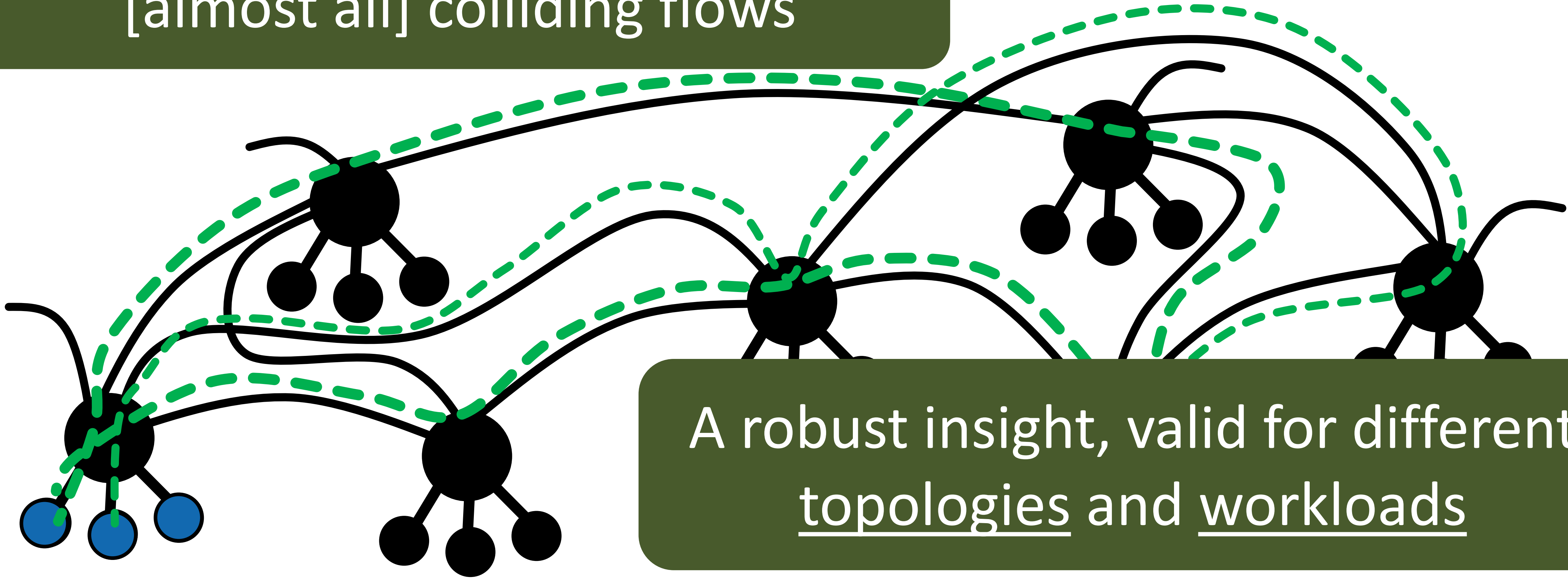
HOW MANY MULTIPLE PATHS DO WE NEED TO TACKLE FLOW COLLISIONS?

Part 1

...For different workloads

Random uniform | Off-diagonal | Stencil | Shuffle | ...

Key Insight: We need **three** disjoint paths per router pair to handle [almost all] colliding flows



A robust insight, valid for different topologies and workloads

How many multiple paths do we need to tackle flow collisions?

Part 1

Key Insight: We need **three** disjoint paths per router pair to handle [almost all] colliding flows

…For different workloads

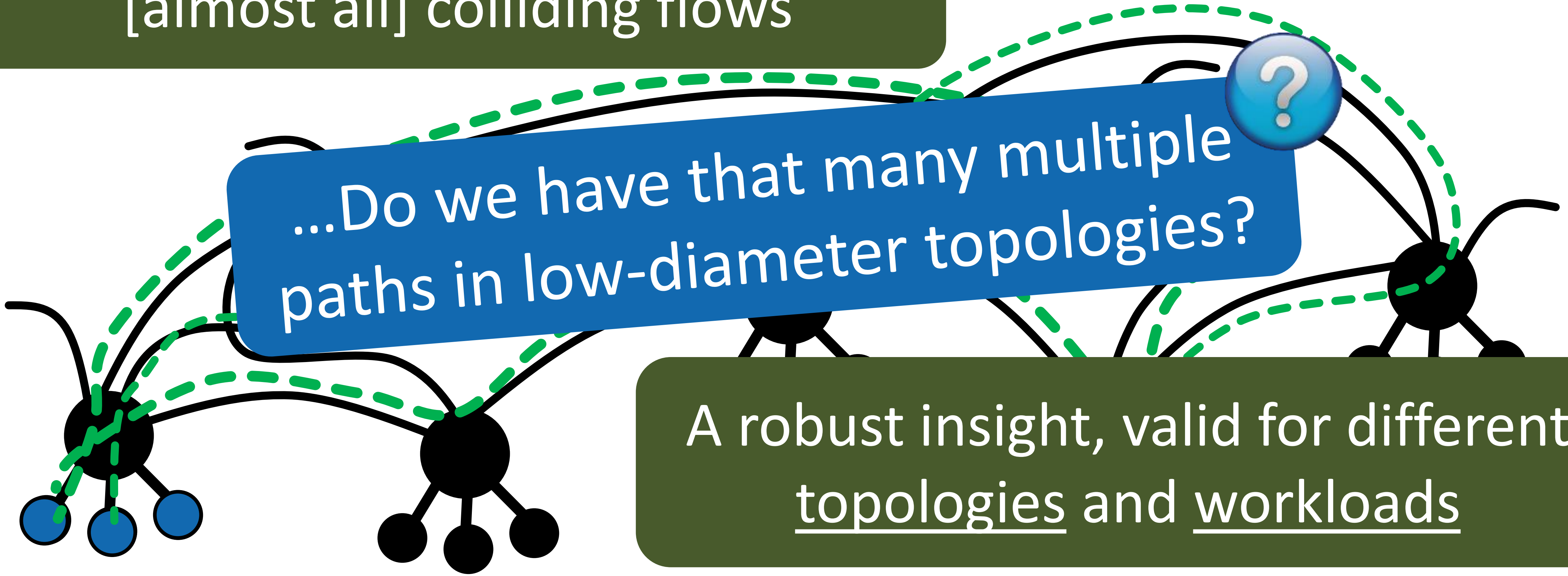
Random uniform

Off-diagonal

Stencil

Shuffle

…



…Do we have that many multiple paths in low-diameter topologies?

A robust insight, valid for different topologies and workloads

ARE THERE ENOUGH MULTIPLE PATHS IN LOW-DIAMETER TOPOLOGIES?

Part 2

ARE THERE ENOUGH MULTIPLE PATHS IN LOW-DIAMETER TOPOLOGIES?

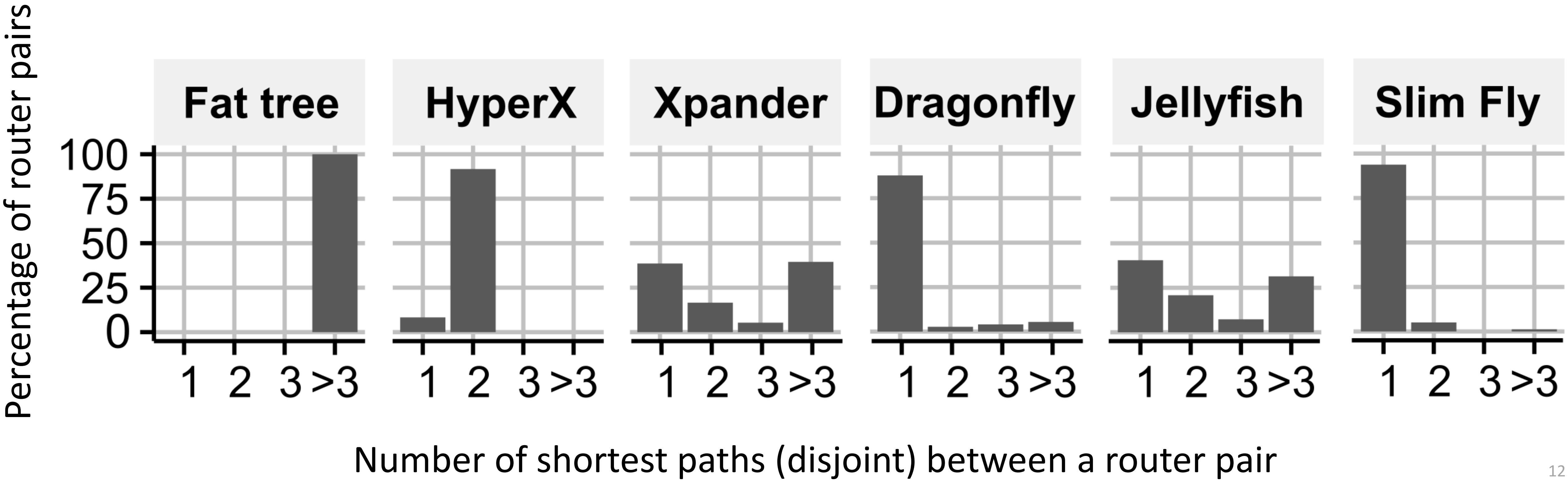
Part 2

Shortest paths

ARE THERE ENOUGH MULTIPLE PATHS IN LOW-DIAMETER TOPOLOGIES?

Part 2

Shortest paths

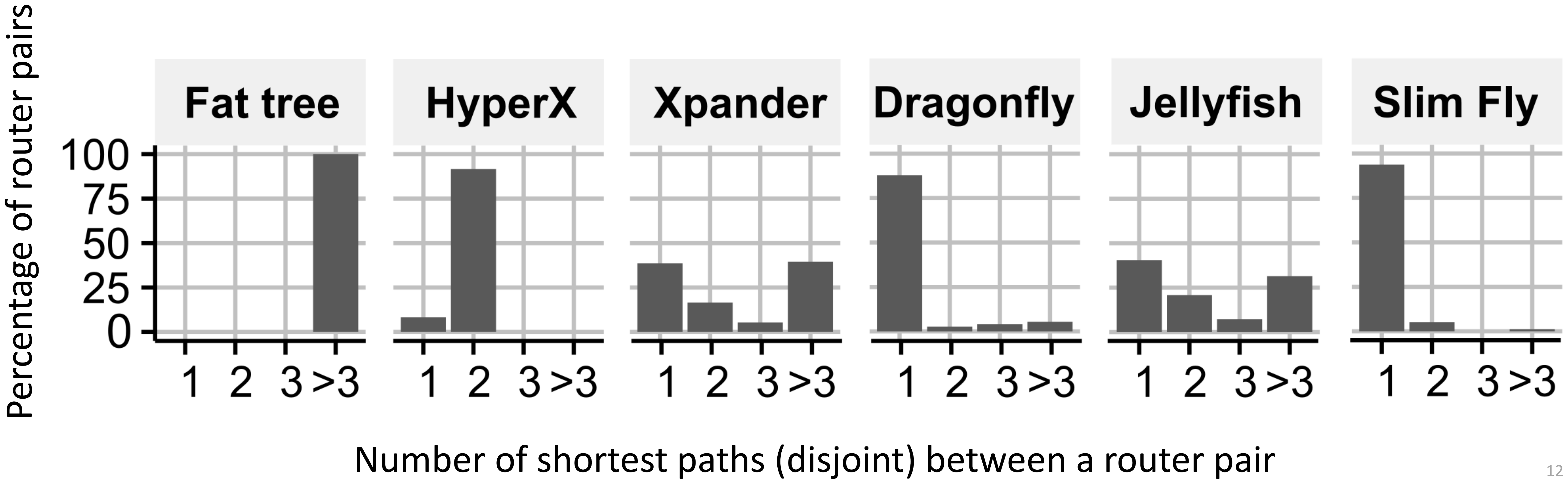


ARE THERE ENOUGH MULTIPLE PATHS IN LOW-DIAMETER TOPOLOGIES?

Part 2

Shortest paths

≈10,000 servers;
comparable cost

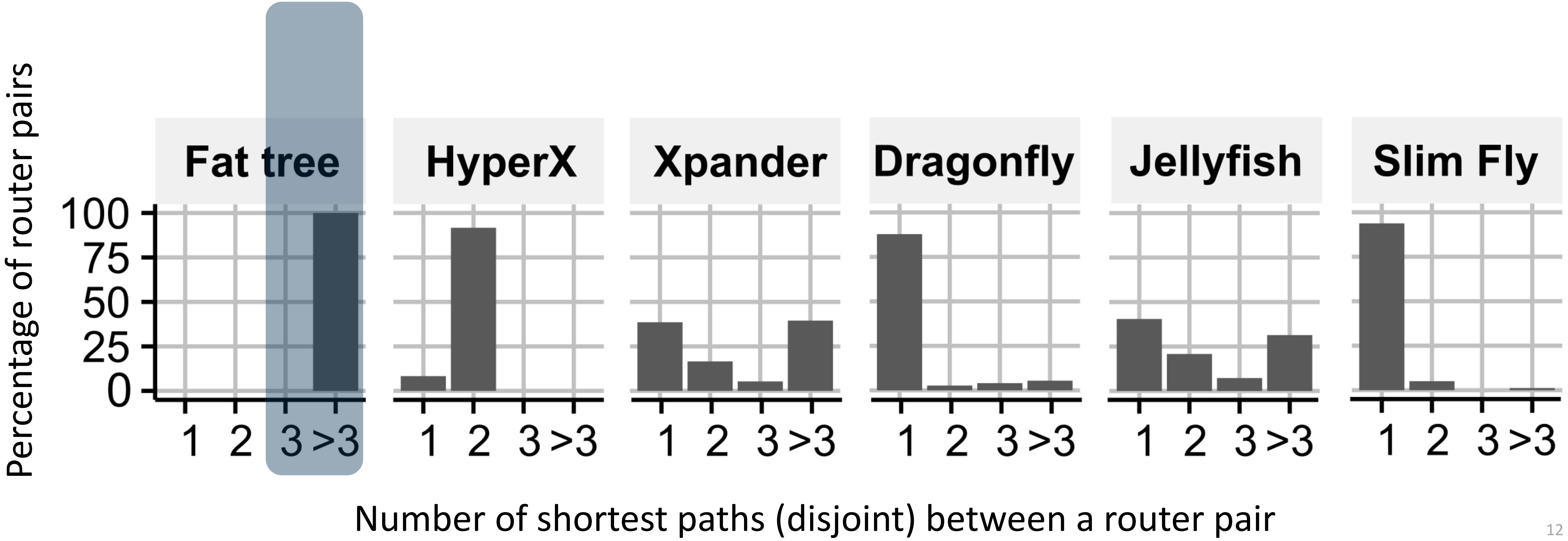


Are there enough multiple paths in low-diameter topologies?

Part 2

Shortest paths

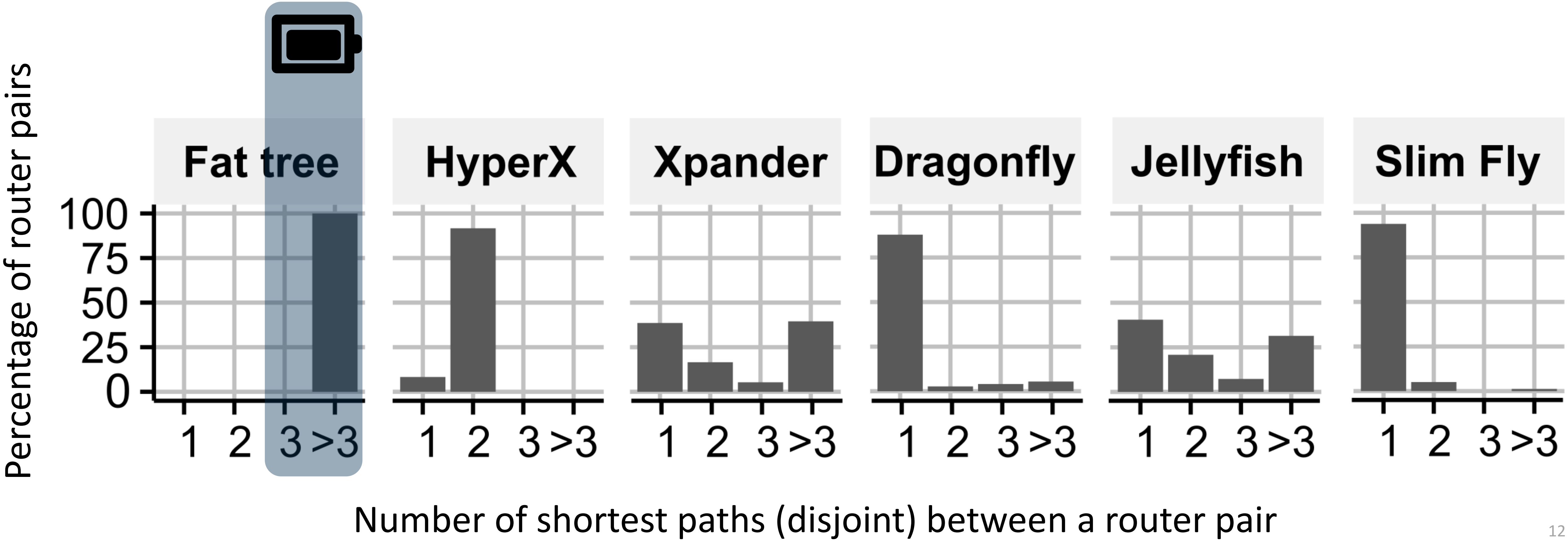
≈10,000 servers;
comparable cost



ARE THERE ENOUGH MULTIPLE PATHS IN LOW-DIAMETER TOPOLOGIES?

Part 2
Shortest paths

≈10,000 servers;
comparable cost

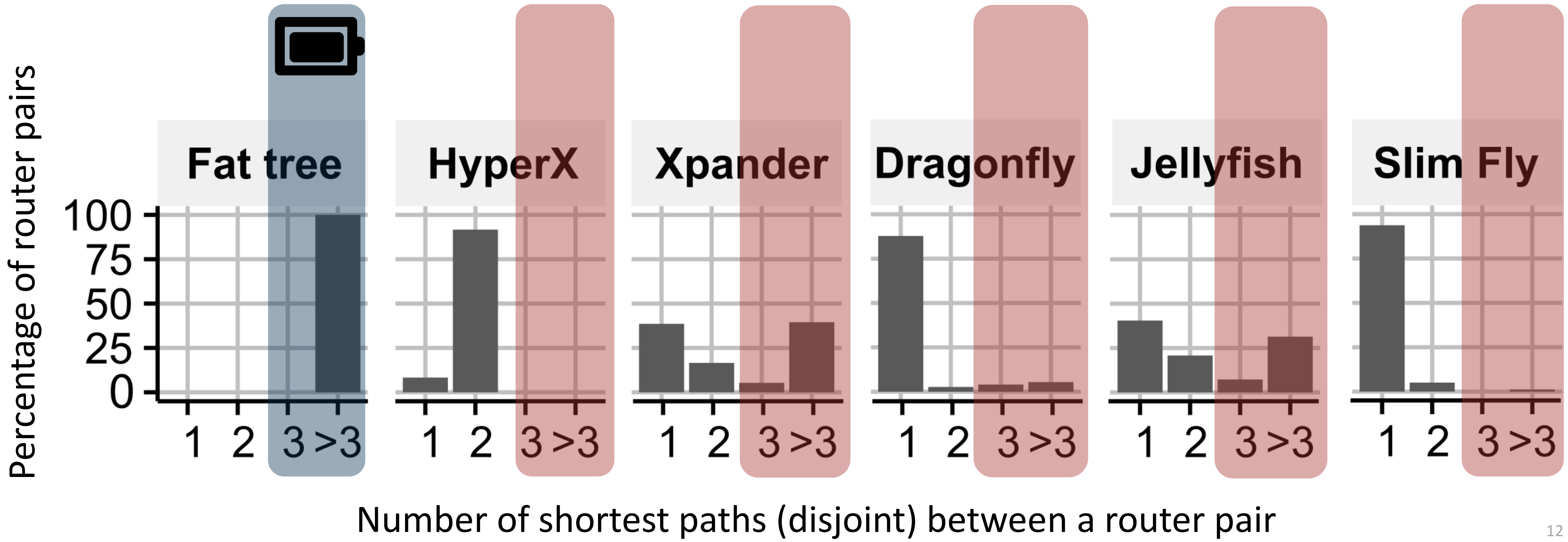


ARE THERE ENOUGH MULTIPLE PATHS IN LOW-DIAMETER TOPOLOGIES?

Part 2

Shortest paths

≈10,000 servers;
comparable cost

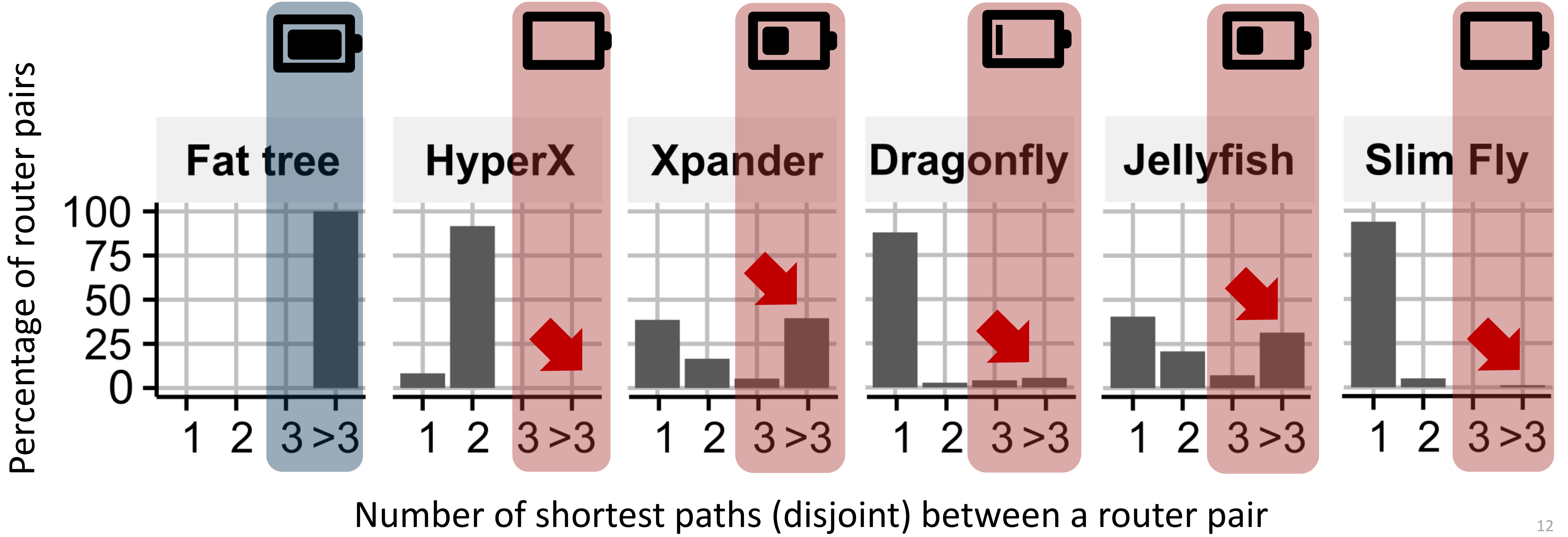


ARE THERE ENOUGH MULTIPLE PATHS IN LOW-DIAMETER TOPOLOGIES?

Part 2

Shortest paths

≈10,000 servers;
comparable cost



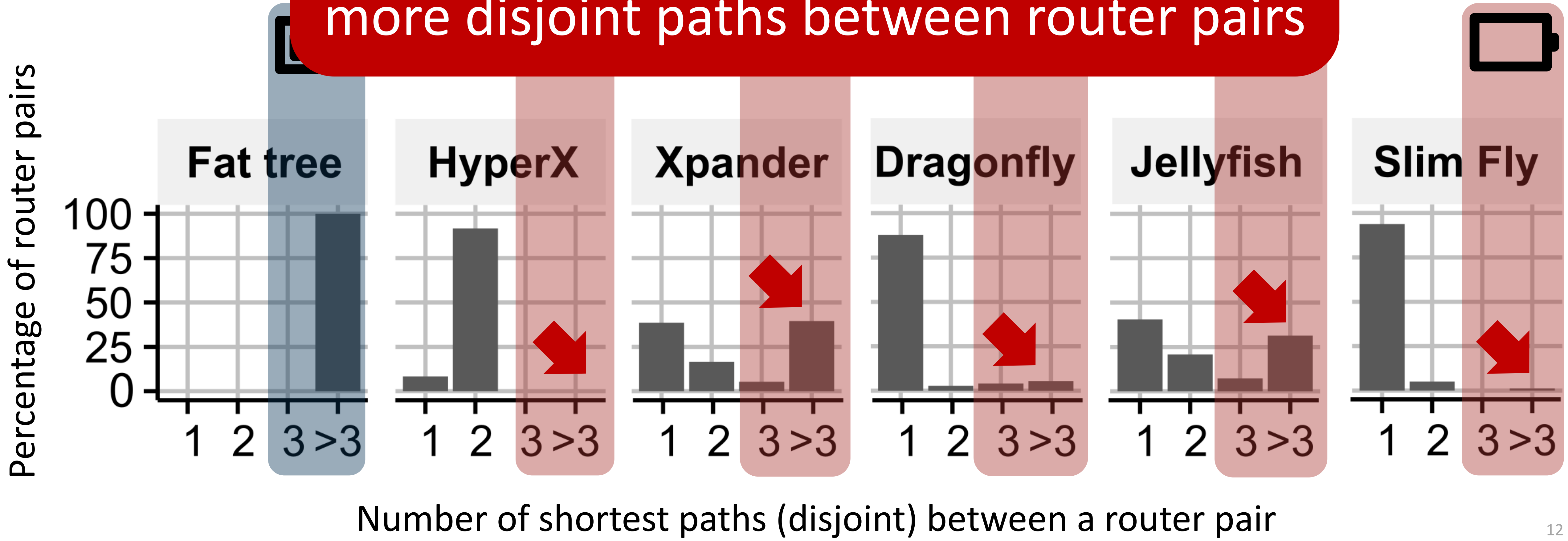
ARE THERE ENOUGH MULTIPLE PATHS IN LOW-DIAMETER TOPOLOGIES?

Part 2

Shortest paths

≈10,000 servers; comparable cost

Key observation: In most cases, there is not enough path diversity for three or more disjoint paths between router pairs



ARE THERE ENOUGH MULTIPLE PATHS IN LOW-DIAMETER TOPOLOGIES?

Part 2

Non- shortest paths

ARE THERE ENOUGH MULTIPLE PATHS IN LOW-DIAMETER TOPOLOGIES?

Part 2

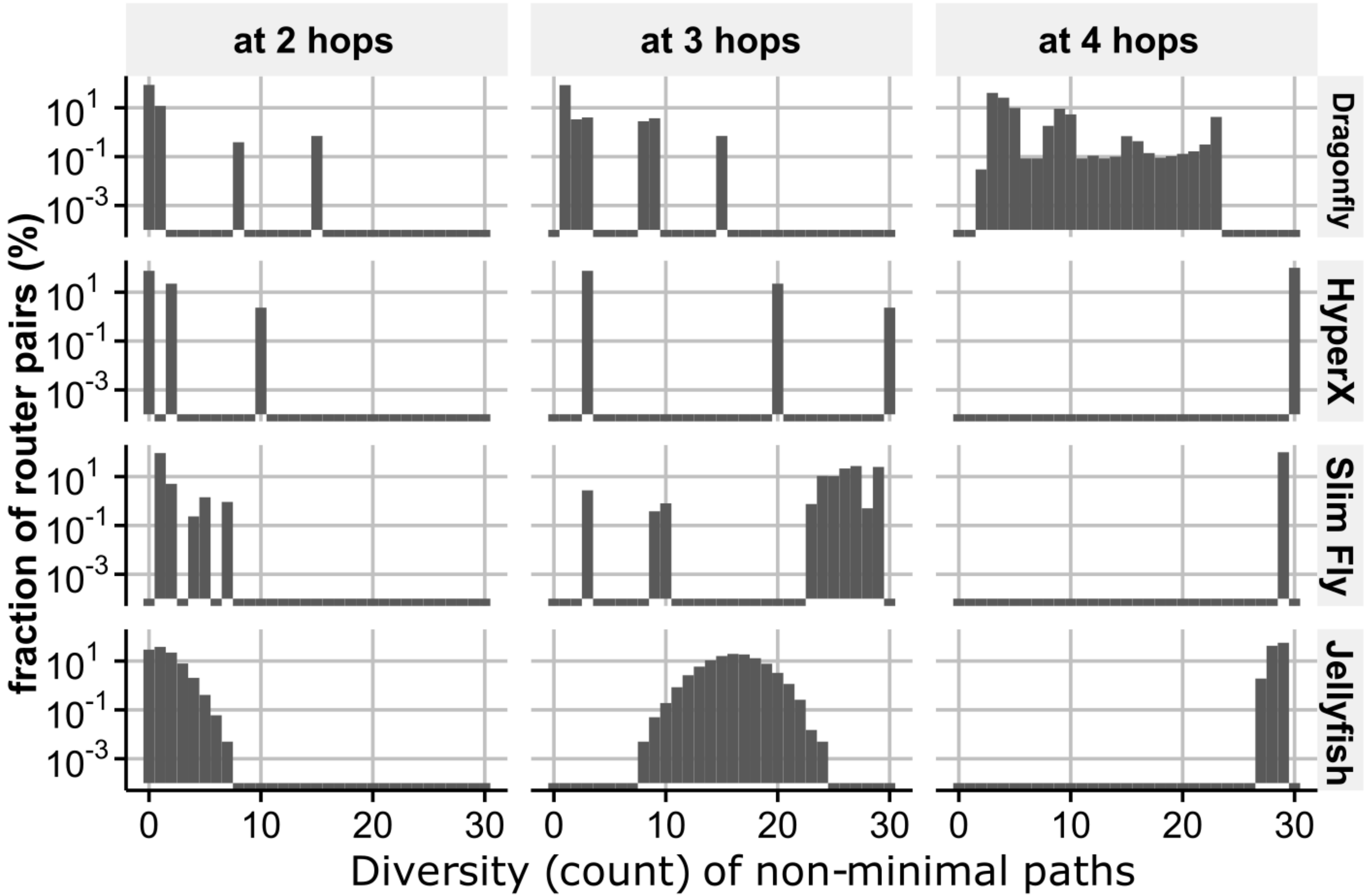
Non- shortest paths

How about **non-shortest** paths?

ARE THERE ENOUGH MULTIPLE PATHS IN LOW-DIAMETER TOPOLOGIES?

Part 2

Non- shortest paths

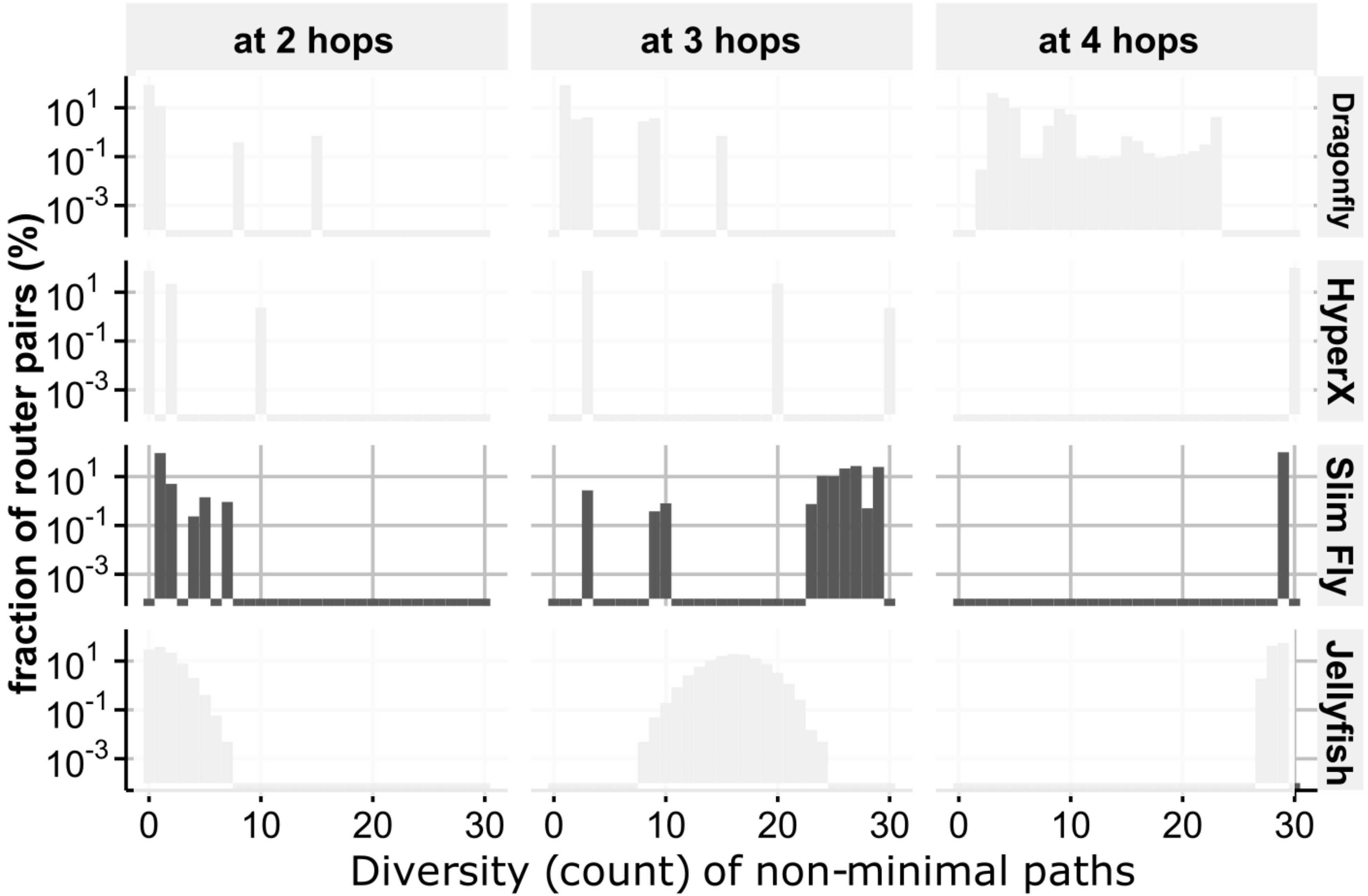


How about **non-shortest** paths?

ARE THERE ENOUGH MULTIPLE PATHS IN LOW-DIAMETER TOPOLOGIES?

Part 2

Non- shortest paths



How about **non-shortest** paths?

ARE THERE ENOUGH MULTIPLE PATHS IN LOW-DIAMETER TOPOLOGIES?

Part 2

Non- shortest paths

How about **non-shortest** paths?

ARE THERE ENOUGH MULTIPLE PATHS IN LOW-DIAMETER TOPOLOGIES?

Part 2

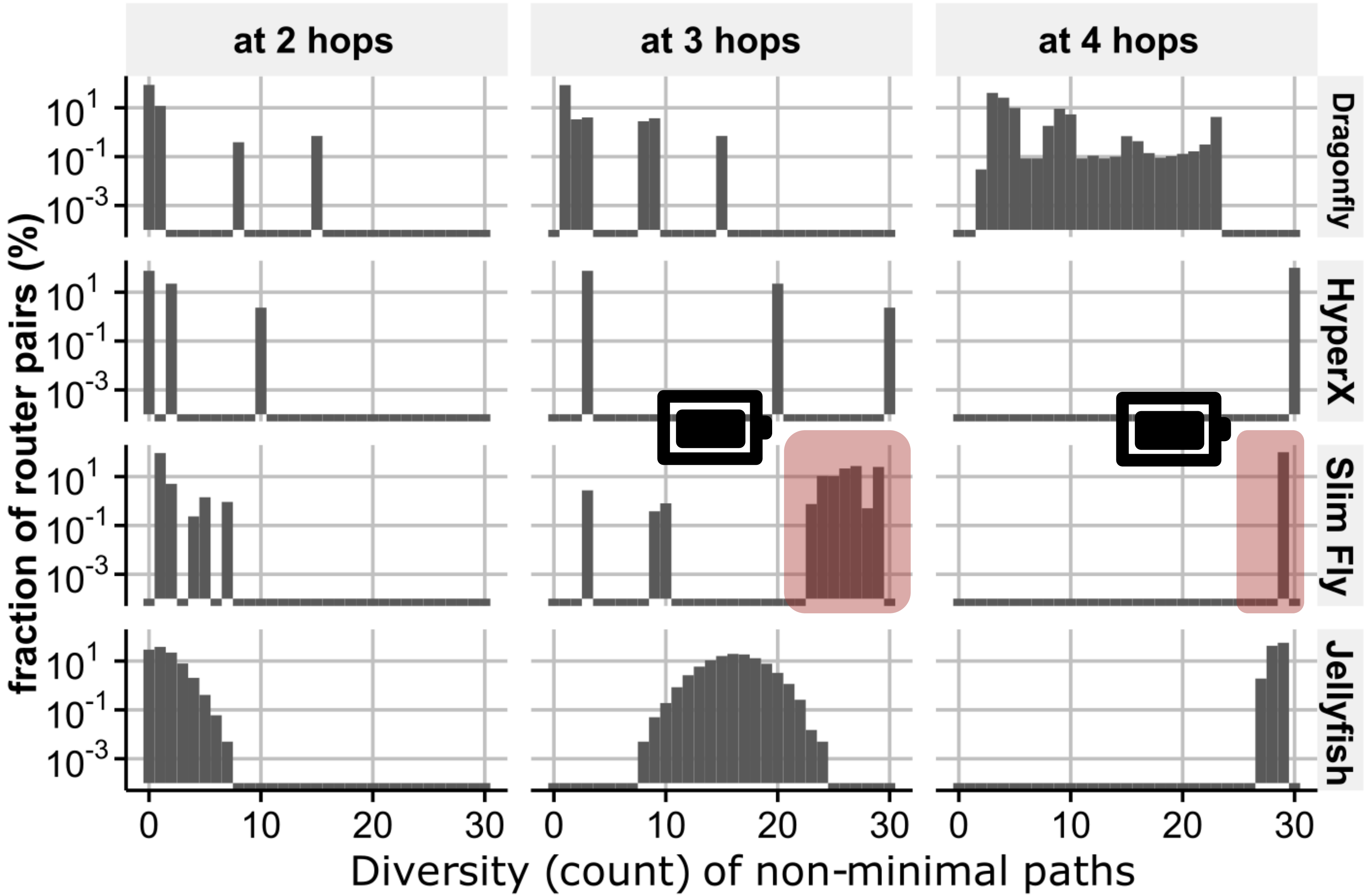
Non- shortest paths

How about **non-shortest** paths?

ARE THERE ENOUGH MULTIPLE PATHS IN LOW-DIAMETER TOPOLOGIES?

Part 2

Non- shortest paths

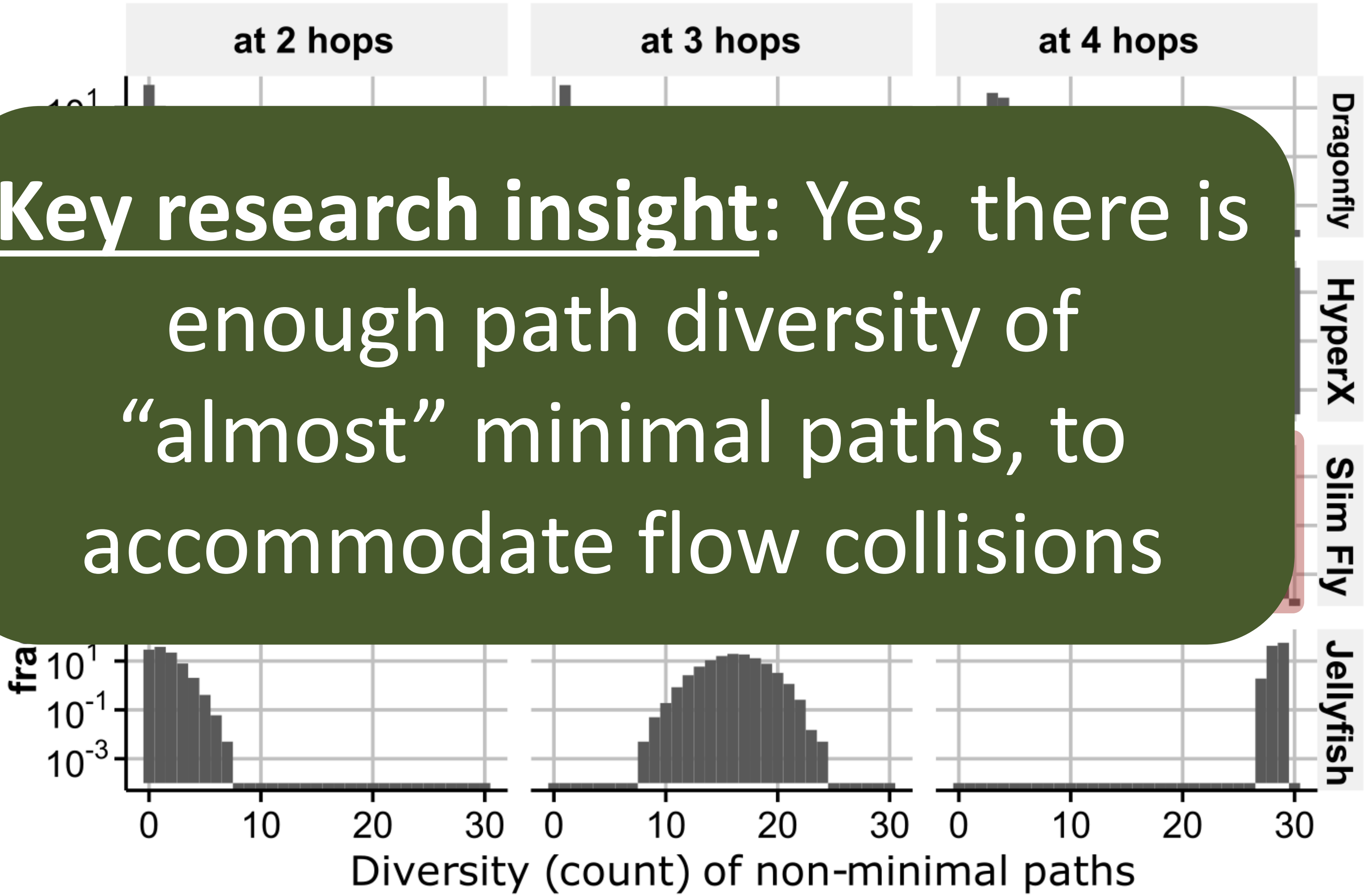


How about **non-shortest** paths?

ARE THERE ENOUGH MULTIPLE PATHS IN LOW-DIAMETER TOPOLOGIES?

Part 2

Non- shortest paths



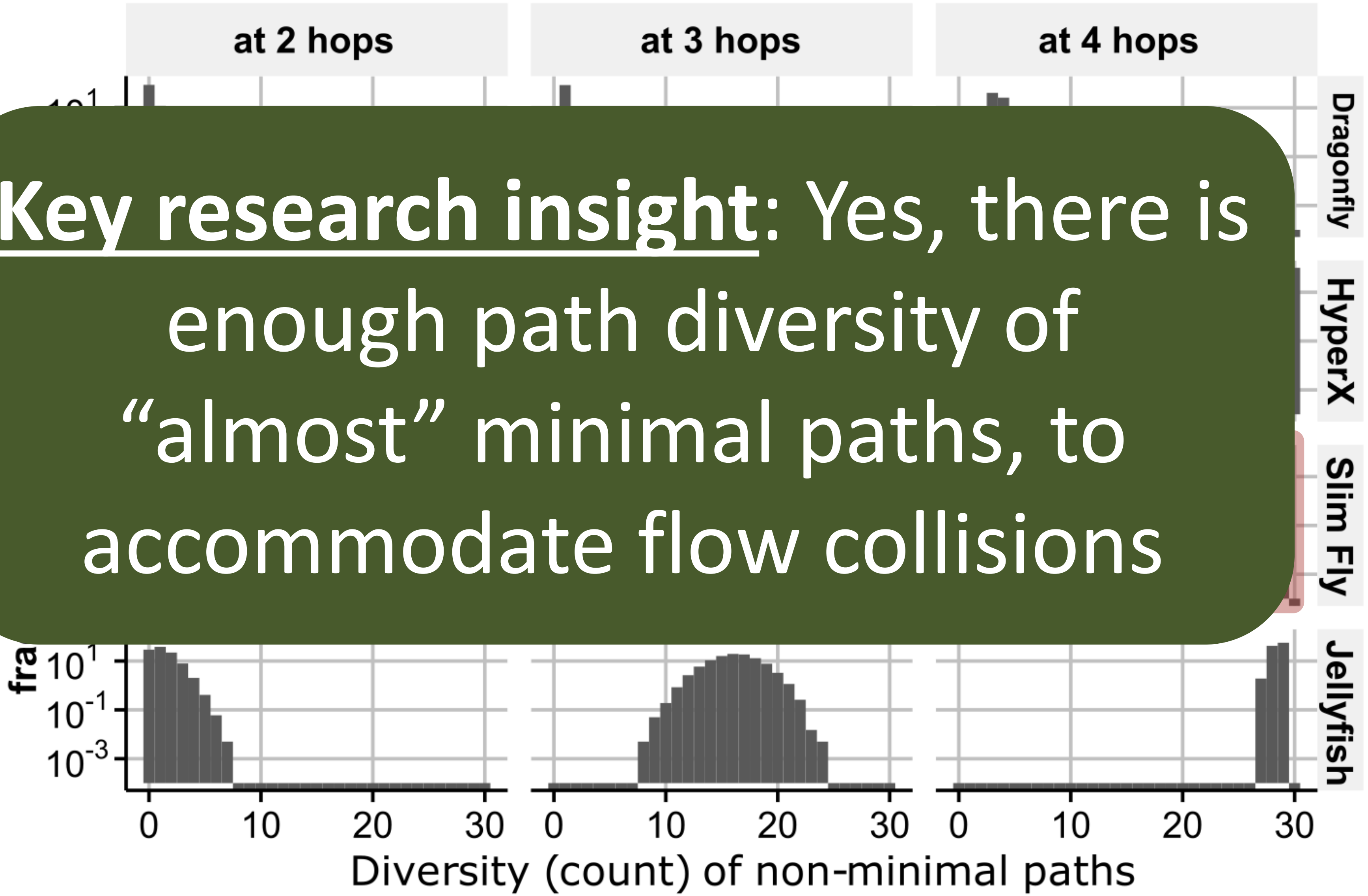
Key research insight: Yes, there is enough path diversity of "almost" minimal paths, to accommodate flow collisions

How about **non-shortest** paths?

ARE THERE ENOUGH MULTIPLE PATHS IN LOW-DIAMETER TOPOLOGIES?

Part 2

Non- shortest paths



Key research insight: Yes, there is enough path diversity of "almost" minimal paths, to accommodate flow collisions

How about **non-shortest** paths?

Let's sum up...

Multipath Routing: Motivation

What are the problems that we want to tackle with multipathing?

Flows collide!

How many paths (in the network) do we need to „accommodate" collisions?

Multipathing

One collision

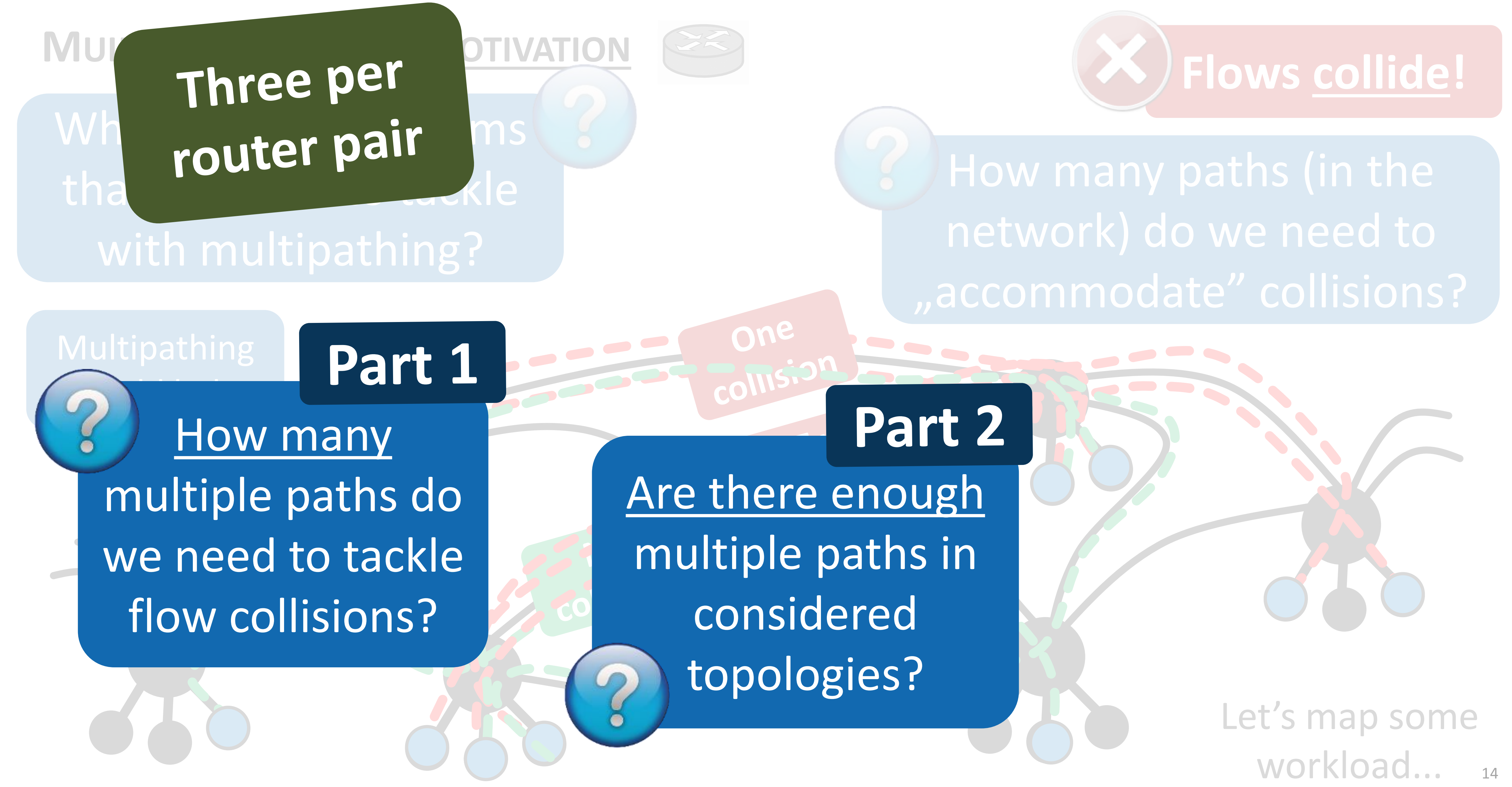
Part 1

How many multiple paths do we need to tackle flow collisions?

Part 2

Are there enough multiple paths in considered topologies?

Let's map some workload...



Three per router pair
Flows collide!
How many paths (in the network) do we need to „accommodate" collisions?
One collision
Part 1
How many multiple paths do we need to tackle flow collisions?
Part 2
Are there enough multiple paths in considered topologies?
Let's map some workload...

spcl.inf.ethz.ch
@spcl_eth
ETH zürich

MU... OTIVATION

Wh... ms tha... tackle with multipathing?

Three per router pair

Yes

Flows collide!

How many paths (in the network) do we need to „accommodate” collisions?

Multipathing

One collision

Part 1

How many multiple paths do we need to tackle flow collisions?

Part 2

Are there enough multiple paths in considered topologies?

Let's map some workload...

Three per router pair

Yes

Flows collide!

How many paths (in the network) do we need to „accommodate" collisions?

What … problems that … tackle with multipathing?

Part 1

How many multiple paths do we need to tackle flow collisions?

Part 2

Are there enough multiple paths in considered topologies?

Part 3

How to use such multiple paths in considered topologies?

One collision

workload...

Three per router pair

Yes

Let's develop a multipathing-focused routing architecture using these insights

Part 1

How many multiple paths do we need to tackle flow collisions?

Part 2

Are there enough multiple paths in considered topologies?

Part 3

How to use such multiple paths in considered topologies?

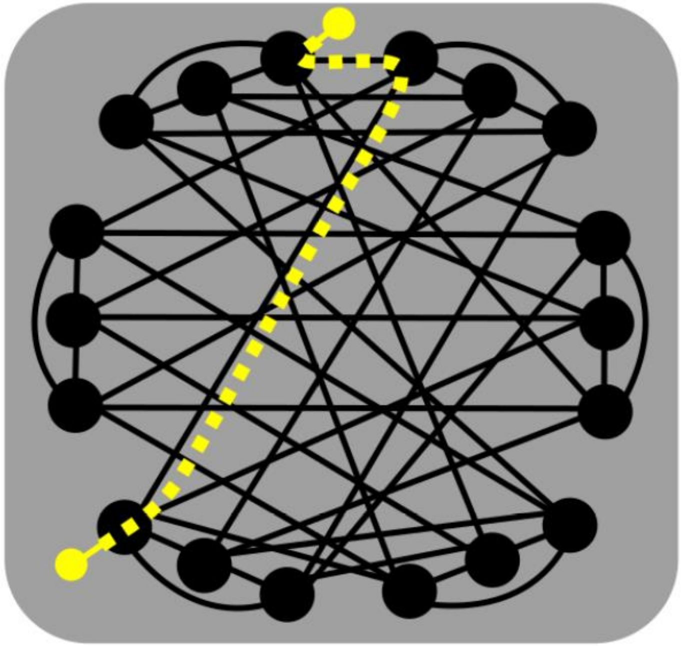
FatPaths architecture: layered routing protocol

Part 3.1

FATPATHS ARCHITECTURE: LAYERED ROUTING PROTOCOL

Part 3.1

Default topology
Shortest path: 2 hops



Default topology
Shortest path: 2 hops

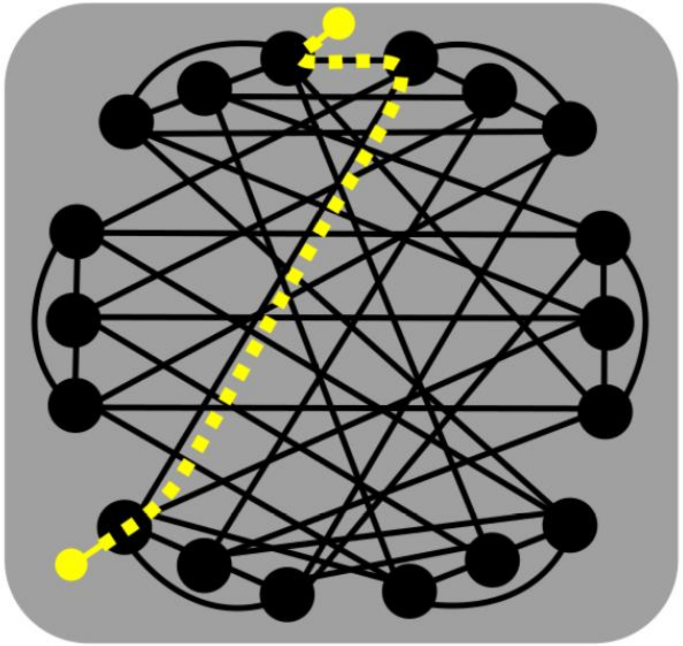
FatPaths architecture: Layered routing protocol

Part 3.1

(a.1) Divide links into subsets (layers).

(a.2) Create a layer by removing a fraction of edges (e.g., random uniform sampling)

Default topology
Shortest path: 2 hops



Default topology
Shortest path: 2 hops

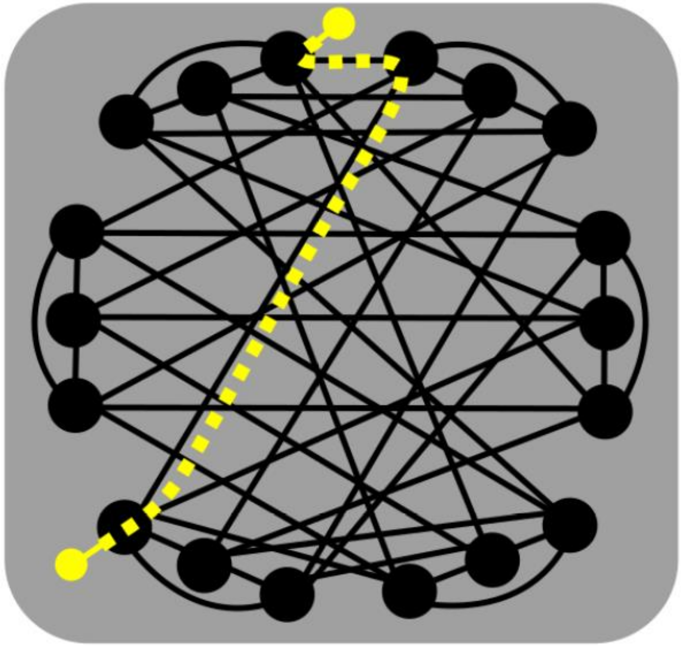
FatPaths architecture: Layered Routing Protocol

Part 3.1

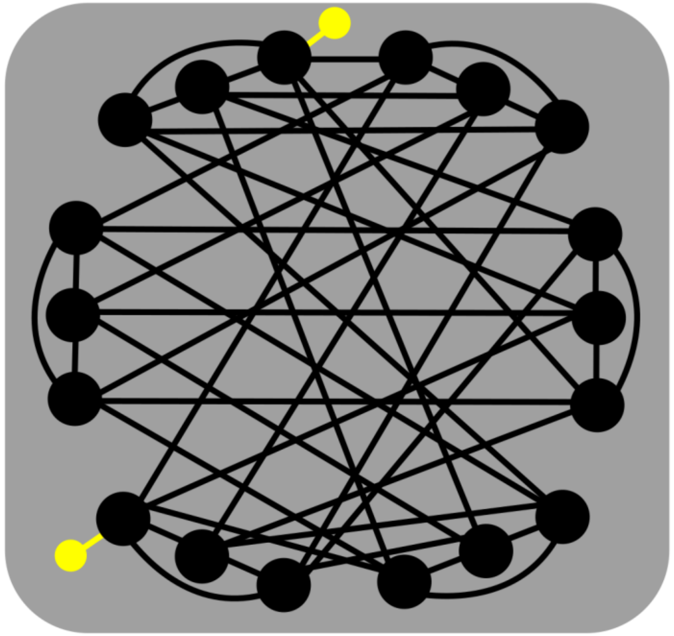
(a.1) Divide links into subsets (layers).

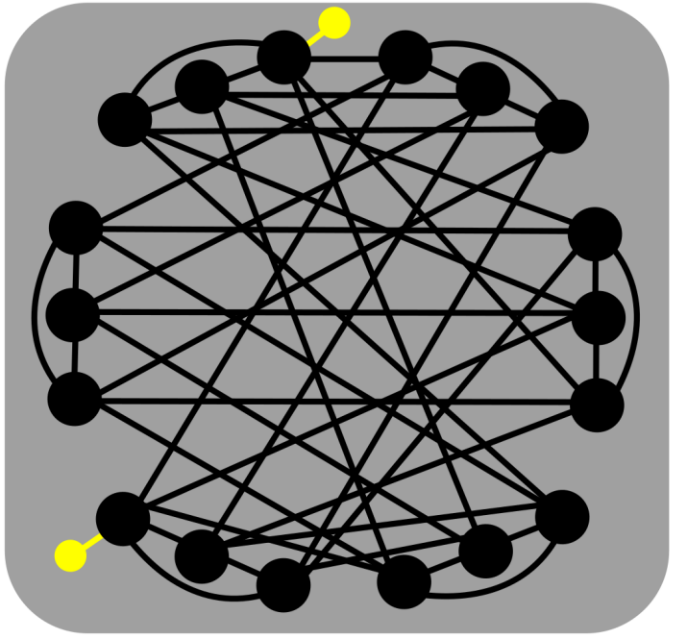
(a.2) Create a layer by removing a fraction of edges (e.g., random uniform sampling)

Default topology
Shortest path: 2 hops



Default topology





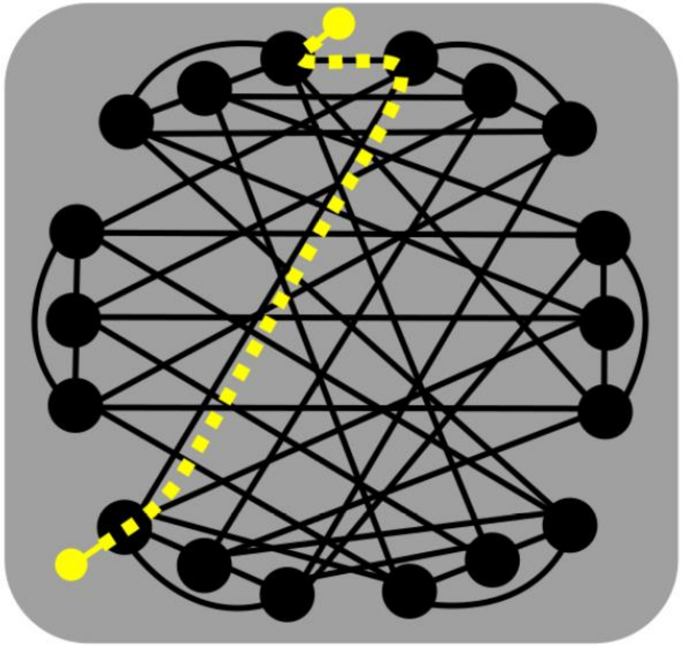
FatPaths Architecture: Layered Routing Protocol

Part 3.1

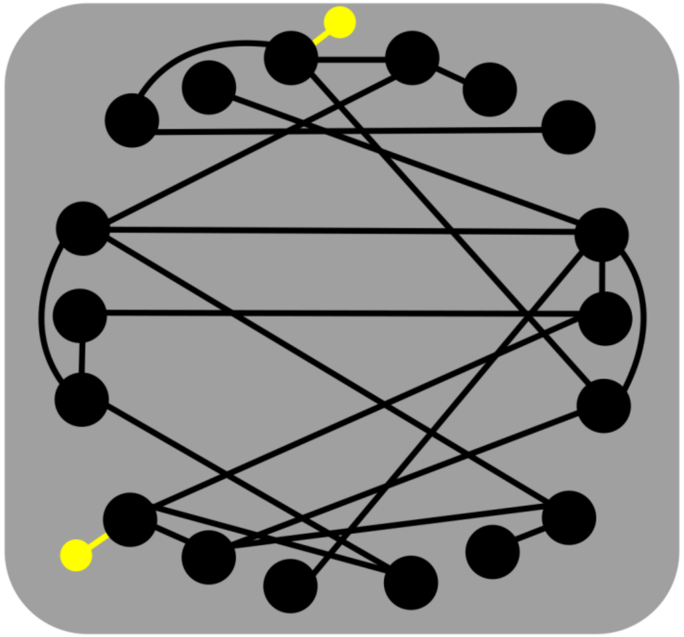
(a.1) Divide links into subsets (layers).

(a.2) Create a layer by removing a fraction of edges (e.g., random uniform sampling)

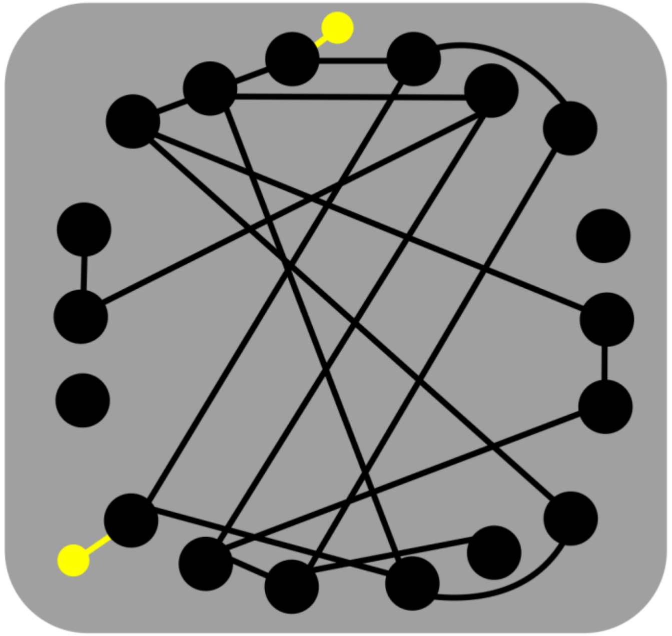
Default topology
Shortest path: 2 hops



Layer 1: "Almost"-
-shortest path: 3 hops
Default topology



Layer 2: "Almost"-
-shortest path: 3 hops



...

FatPaths architecture: Layered routing protocol

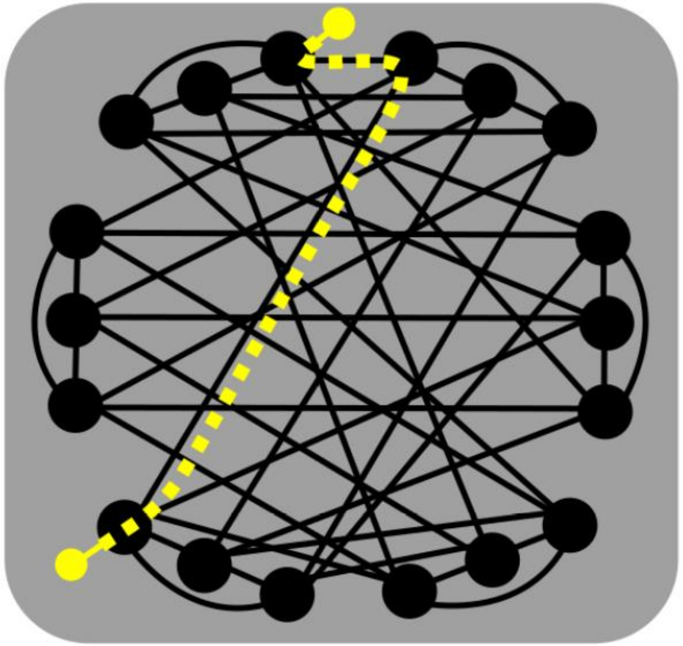
Part 3.1

(a.1) Divide links into subsets (layers).

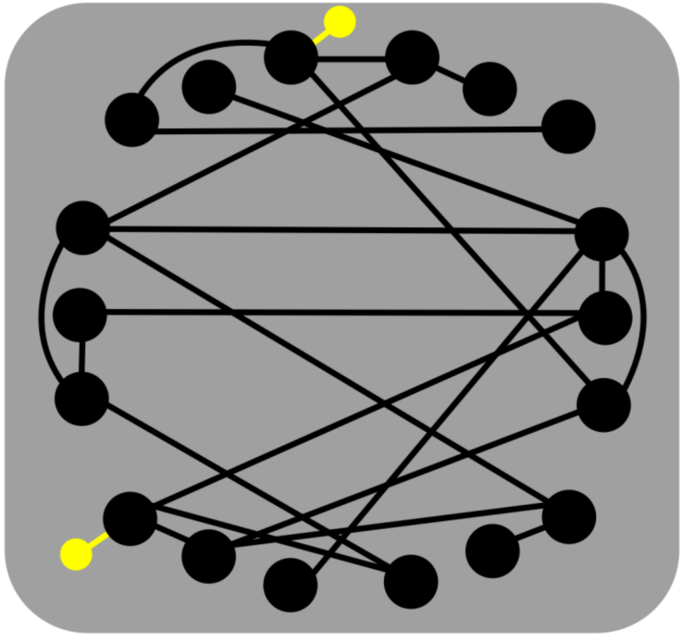
(a.2) Create a layer by removing a fraction of edges (e.g., random uniform sampling)

(b) Divide one flow into subflows and send subflows across different layers

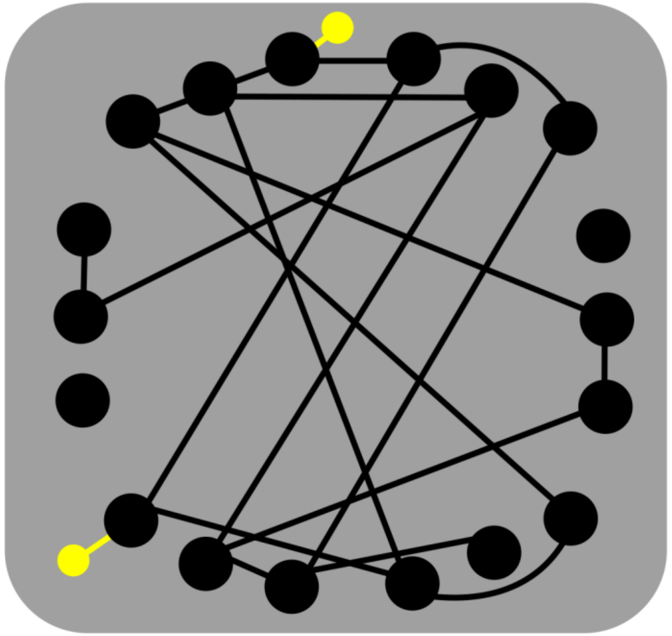
Default topology
Shortest path: 2 hops



Layer 1: "Almost"-
-shortest path: 3 hops
Default topology



Layer 2: "Almost"-
-shortest path: 3 hops



...

FATPATHS ARCHITECTURE: LAYERED ROUTING PROTOCOL

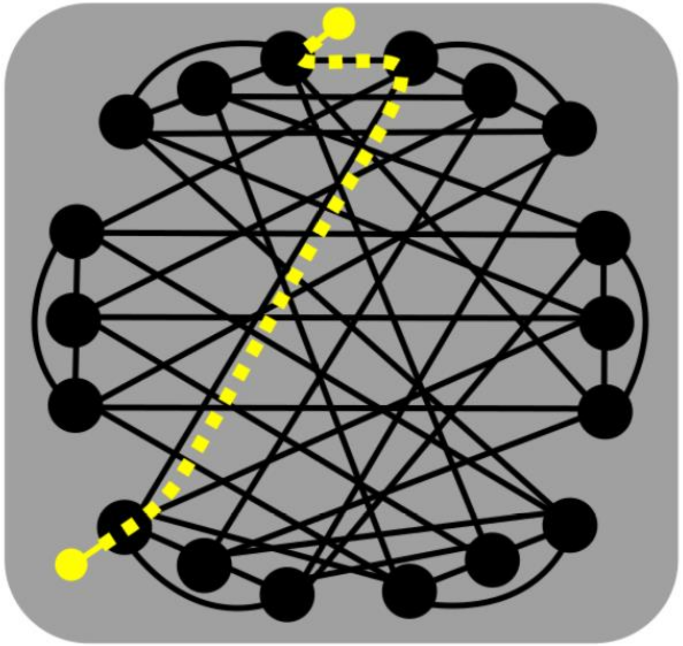
Part 3.1

(a.1) Divide links into subsets (layers).

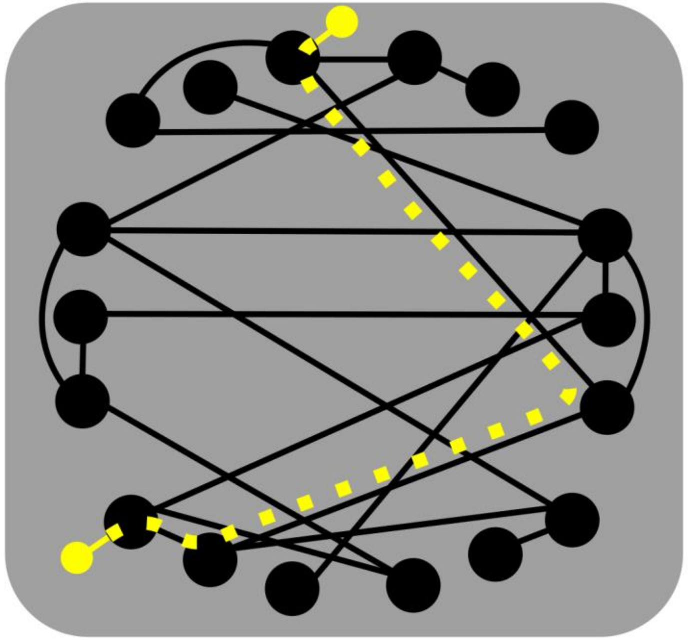
(a.2) Create a layer by removing a fraction of edges (e.g., random uniform sampling)

(b) Divide one flow into subflows and send subflows across different layers

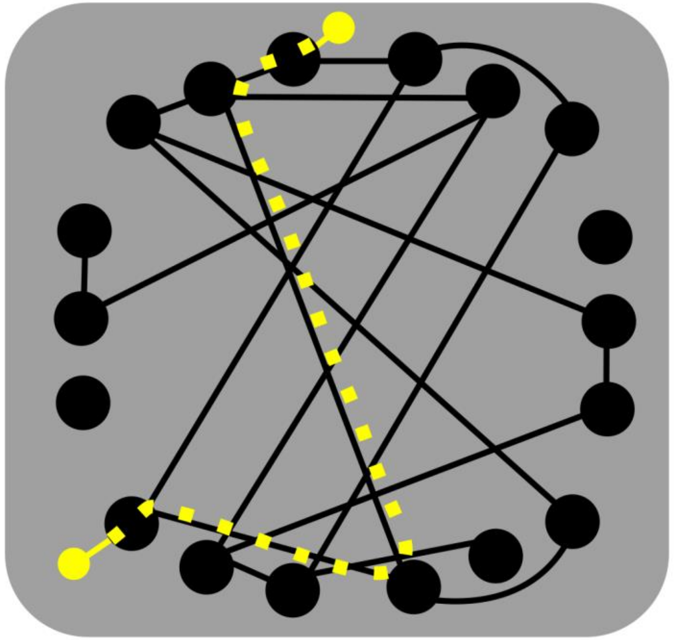
Default topology
Shortest path: 2 hops



Layer 1: "Almost"-
-shortest path: 3 hops
Default topology



Layer 2: "Almost"-
-shortest path: 3 hops



...

FatPaths architecture: layered routing protocol

Part 3.1

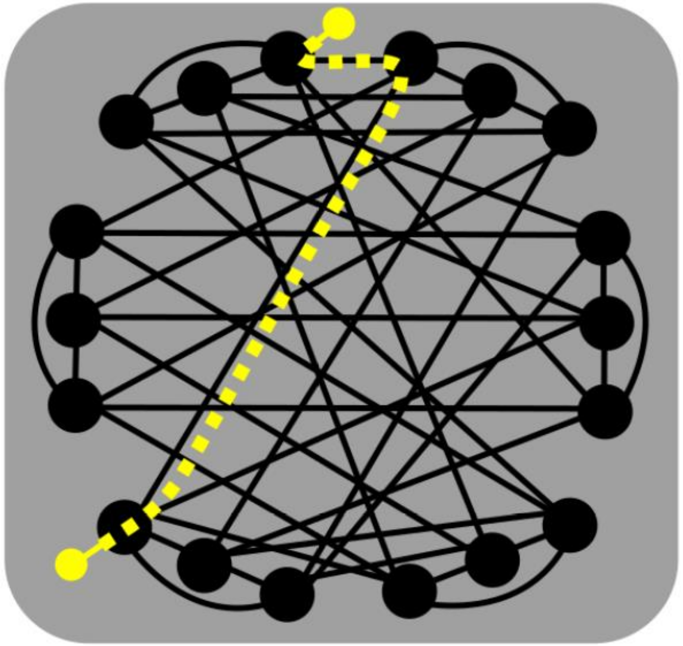
(a.1) Divide links into subsets (layers).

(a.2) Create a layer by removing a fraction of edges (e.g., random uniform sampling)

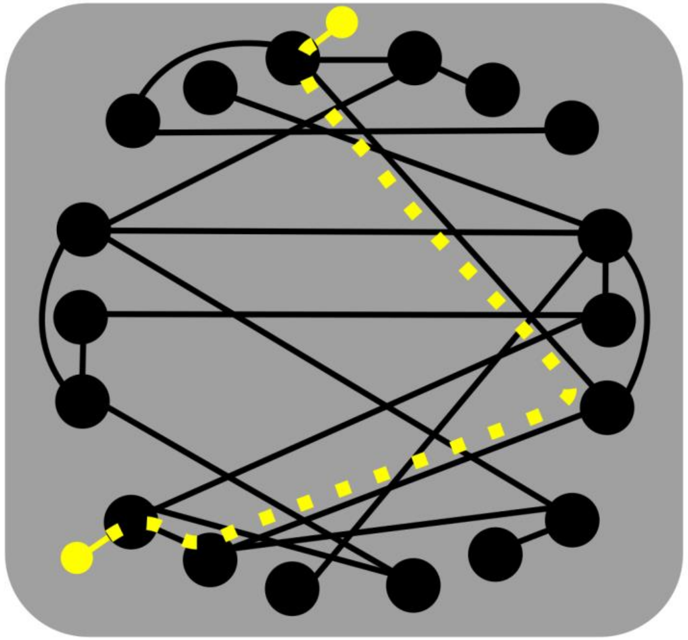
(b) Divide one flow into subflows and send subflows across different layers

(c) Route minimally in each layer. A minimal route in one layer is often non-minimal when considering all links

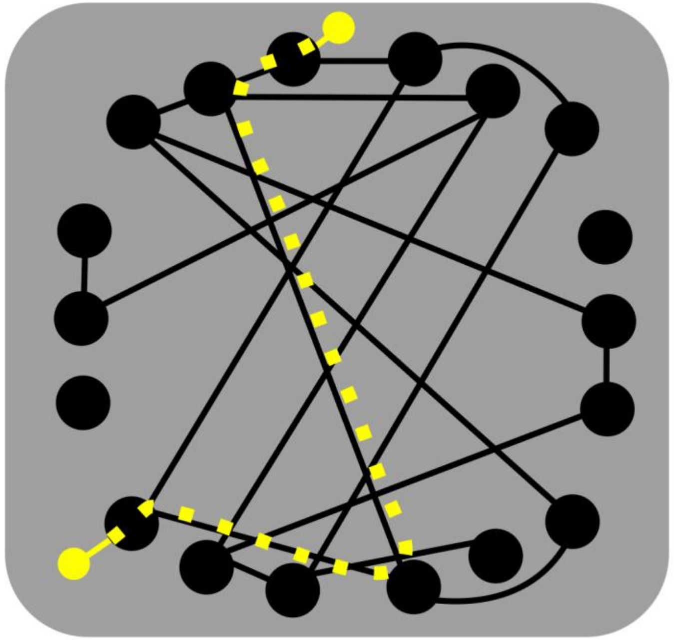
Default topology
Shortest path: 2 hops



Layer 1: "Almost"-
Default topology
-shortest path: 3 hops



Layer 2: "Almost"-
-shortest path: 3 hops



...

FatPaths Architecture: Layered Routing Protocol

Part 3.1

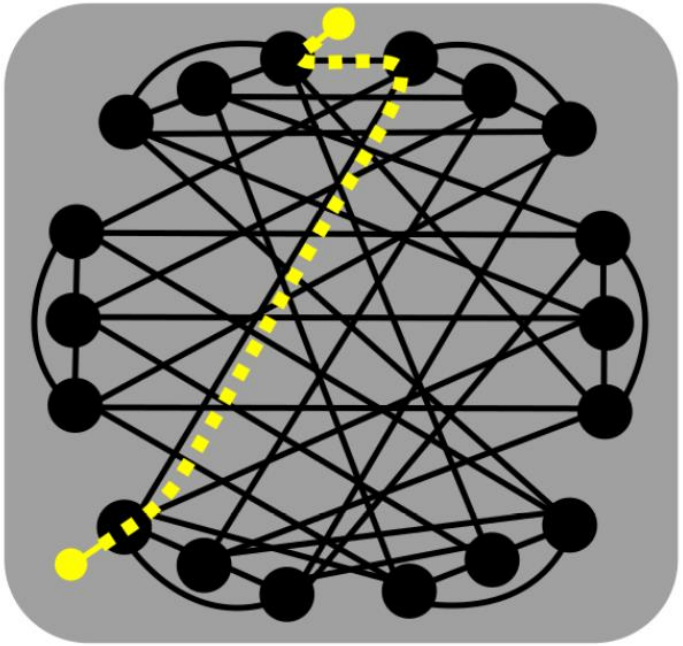
(a.1) Divide links into subsets (layers).

(a
re

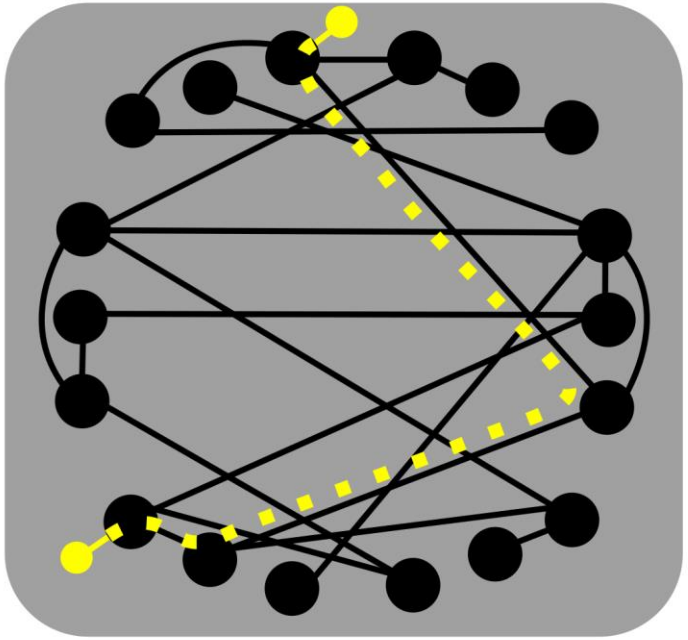
…inimally in each
…mal route in one
…en non-minimal
…idering all links

Design justification: Different layers enable multipathing

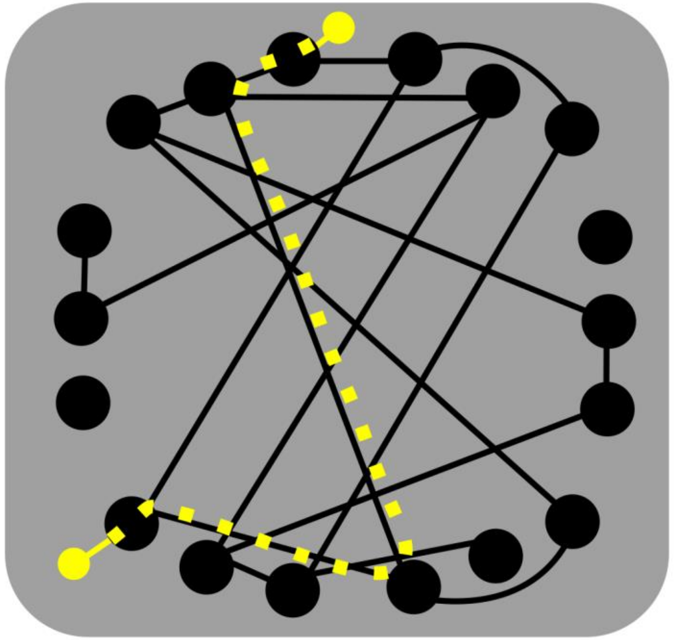
Default topology
Shortest path: 2 hops



Layer 1: "Almost"-
-shortest path: 3 hops
Default topology



Layer 2: "Almost"-
-shortest path: 3 hops



…

FatPaths Architecture: Layered Routing Protocol

Part 3.1

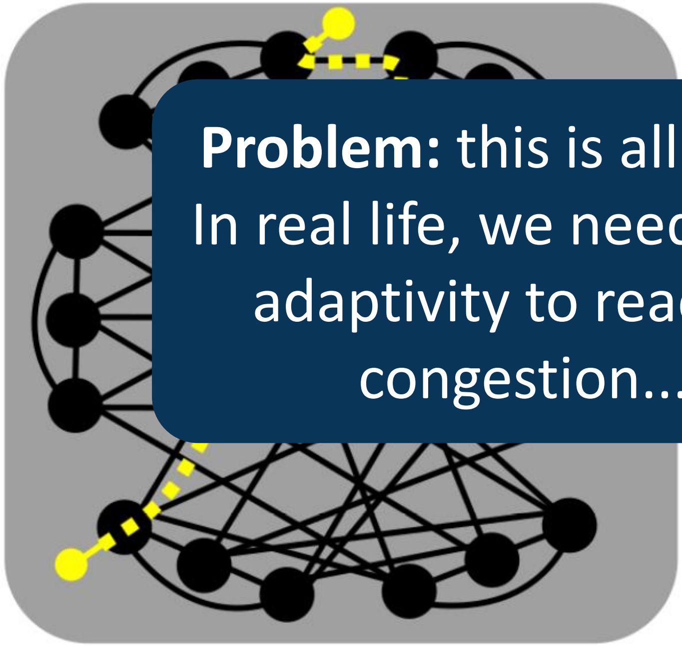
(a.1) Divide links into subsets (layers).

(a
re
(

…inimally in each
…mal route in one
…en non-minimal
…idering all links

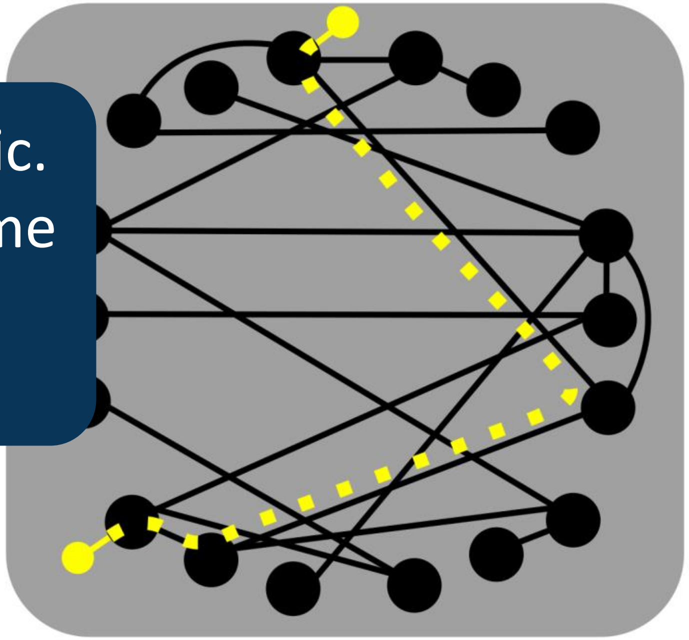
Design justification: Different layers enable multipathing

Default topology
Shortest path: 2 hops

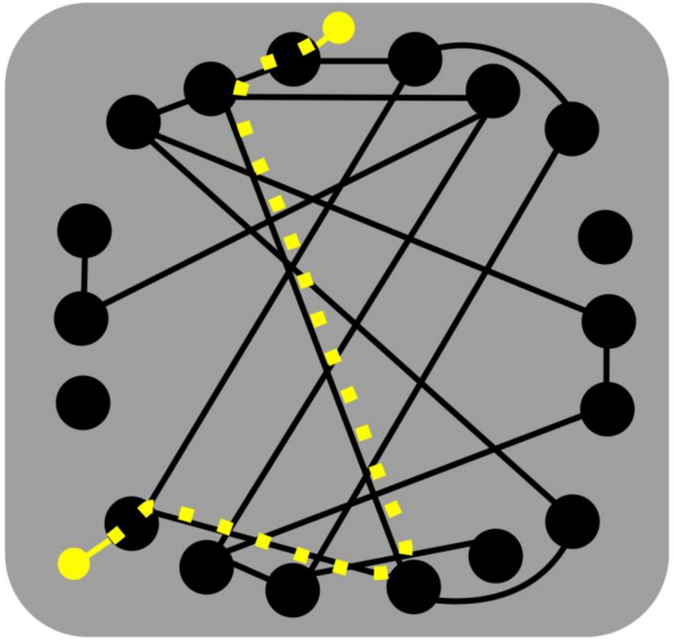


Problem: this is all static. In real life, we need some adaptivity to react to congestion...

Layer 1: “Almost”-
-shortest path: 3 hops



Layer 2: “Almost”-
-shortest path: 3 hops



...

FATPATHS ARCHITECTURE: LAYERED ROUTING PROTOCOL

Part 3.1

(a.1) Divide links into subsets (layers).

(a ... re ...

...inimally in each ...mal route in one ...en non-minimal ...idering all links

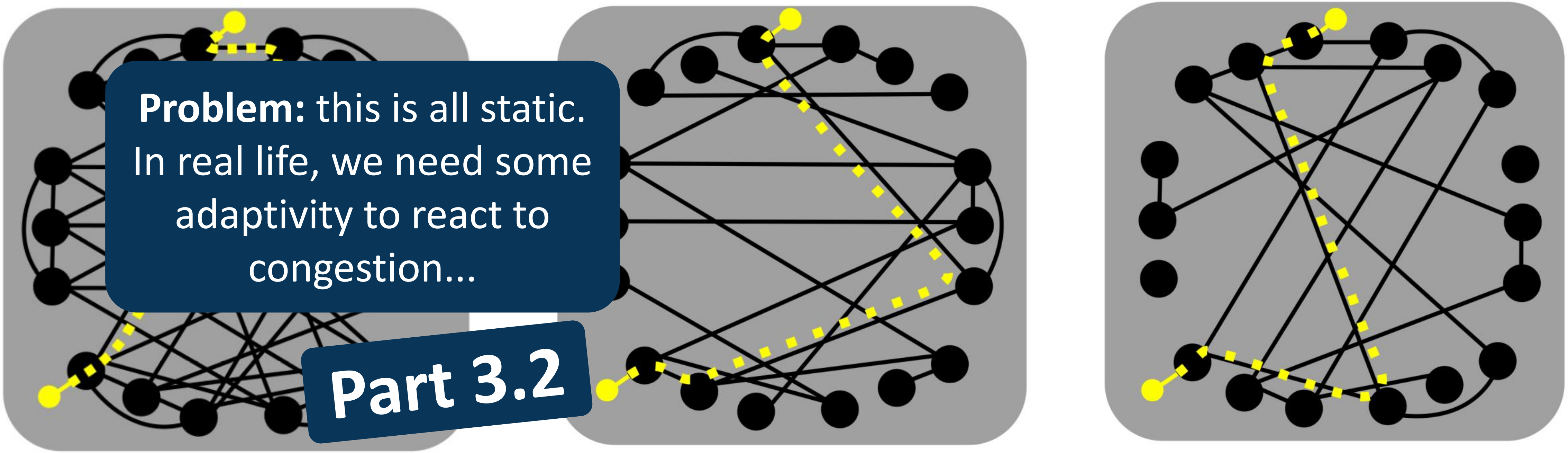
Design justification: Different layers enable multipathing

Default topology
Shortest path: 2 hops

Layer 1: "Almost"-
-shortest path: 3 hops

Default topology

Layer 2: "Almost"-
-shortest path: 3 hops



...

Problem: this is all static. In real life, we need some adaptivity to react to congestion...

Part 3.2

FatPaths architecture: load balancing with flowlets

FatPaths architecture: load balancing with flowlets

How to load balance traffic over different paths of different lenghts?

FatPaths Architecture: Load Balancing with Flowlets

A flowlet [1,2] is a sequence of packets within one flow, separated from other flowlets by sufficient time gaps, which prevents packet reordering at the receiver.

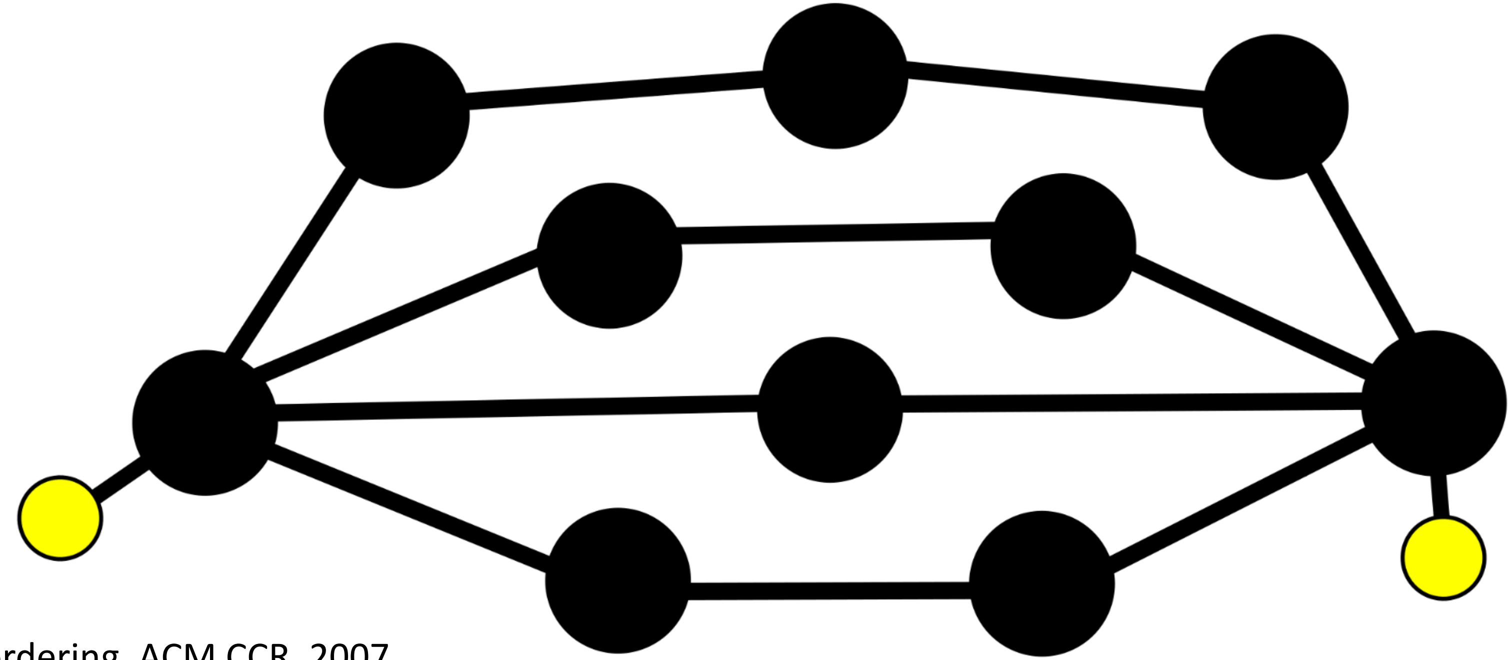
How to load balance traffic over different paths of different lenghts?

[1] S. Kandula et al. Dynamic load balancing without packet reordering. ACM CCR. 2007.
[2] E. Vanini et al. Let It Flow: Resilient Asymmetric Load Balancing with Flowlet Switching. NSDI’17.

FatPaths Architecture: Load Balancing with Flowlets

A flowlet [1,2] is a sequence of packets within one flow, separated from other flowlets by sufficient time gaps, which prevents packet reordering at the receiver.

How to load balance traffic over different paths of different lenghts?

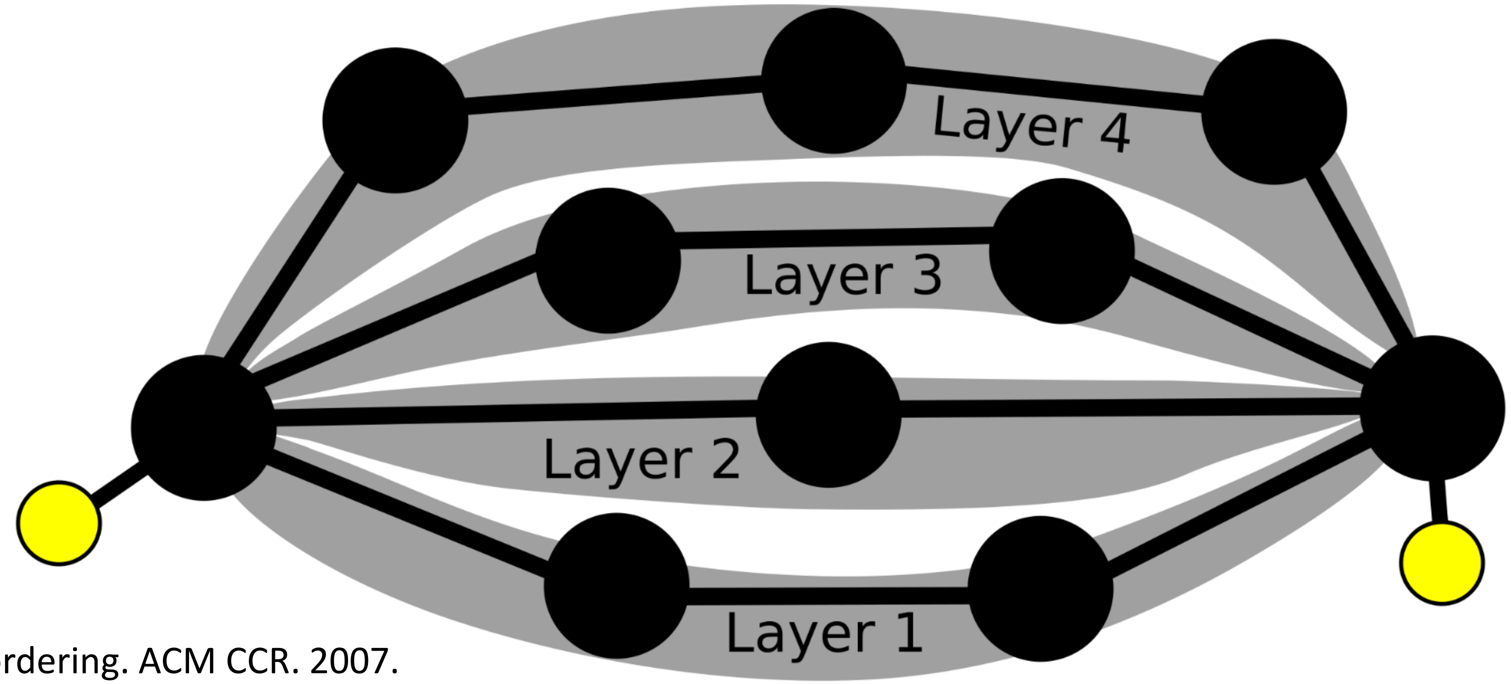


[1] S. Kandula et al. Dynamic load balancing without packet reordering. ACM CCR. 2007.
[2] E. Vanini et al. Let It Flow: Resilient Asymmetric Load Balancing with Flowlet Switching. NSDI'17.

FatPaths Architecture: Load Balancing with Flowlets

A flowlet [1,2] is a sequence of packets within one flow, separated from other flowlets by sufficient time gaps, which prevents packet reordering at the receiver.

How to load balance traffic over different paths of different lenghts?

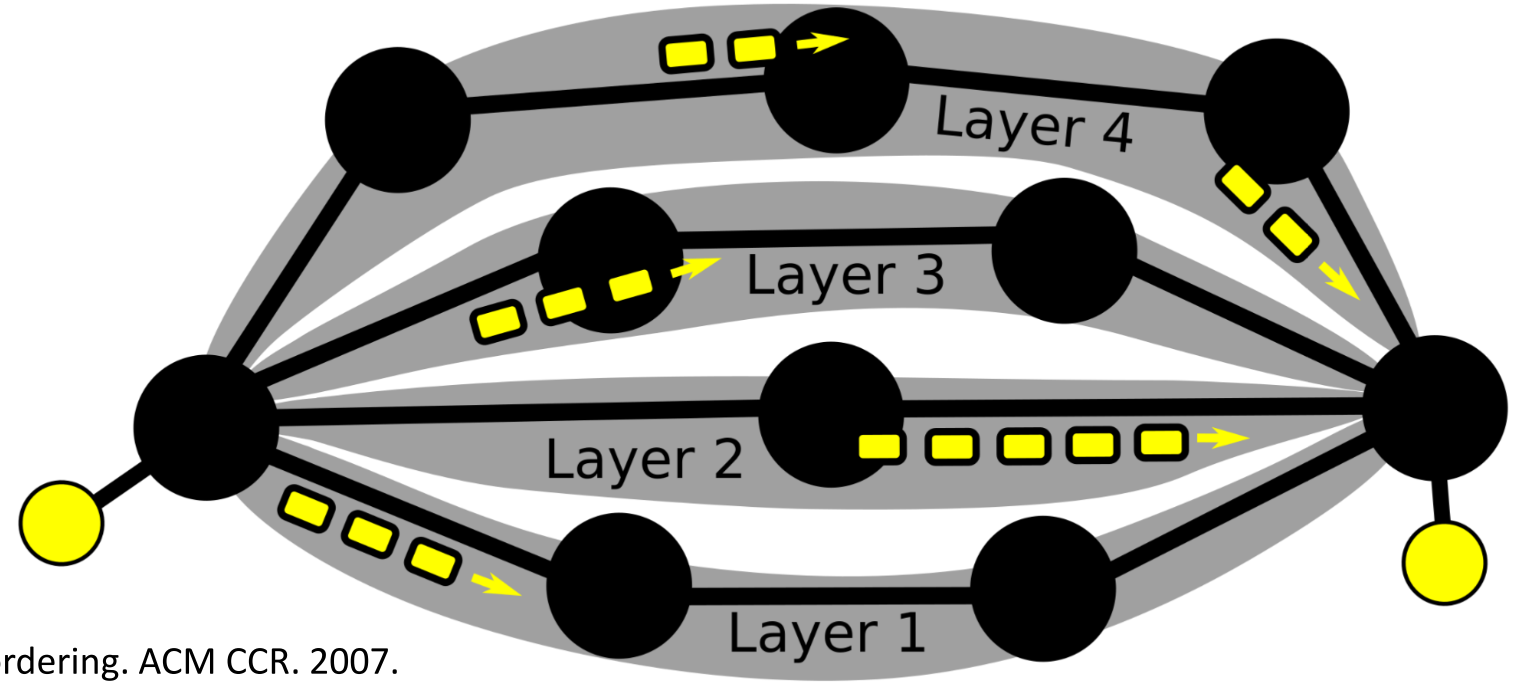


[1] S. Kandula et al. Dynamic load balancing without packet reordering. ACM CCR. 2007.
[2] E. Vanini et al. Let It Flow: Resilient Asymmetric Load Balancing with Flowlet Switching. NSDI'17.

FATPATHS ARCHITECTURE: LOAD BALANCING WITH FLOWLETS

A flowlet [1,2] is a sequence of packets within one flow, separated from other flowlets by sufficient time gaps, which prevents packet reordering at the receiver.

How to load balance traffic over different paths of different lenghts?



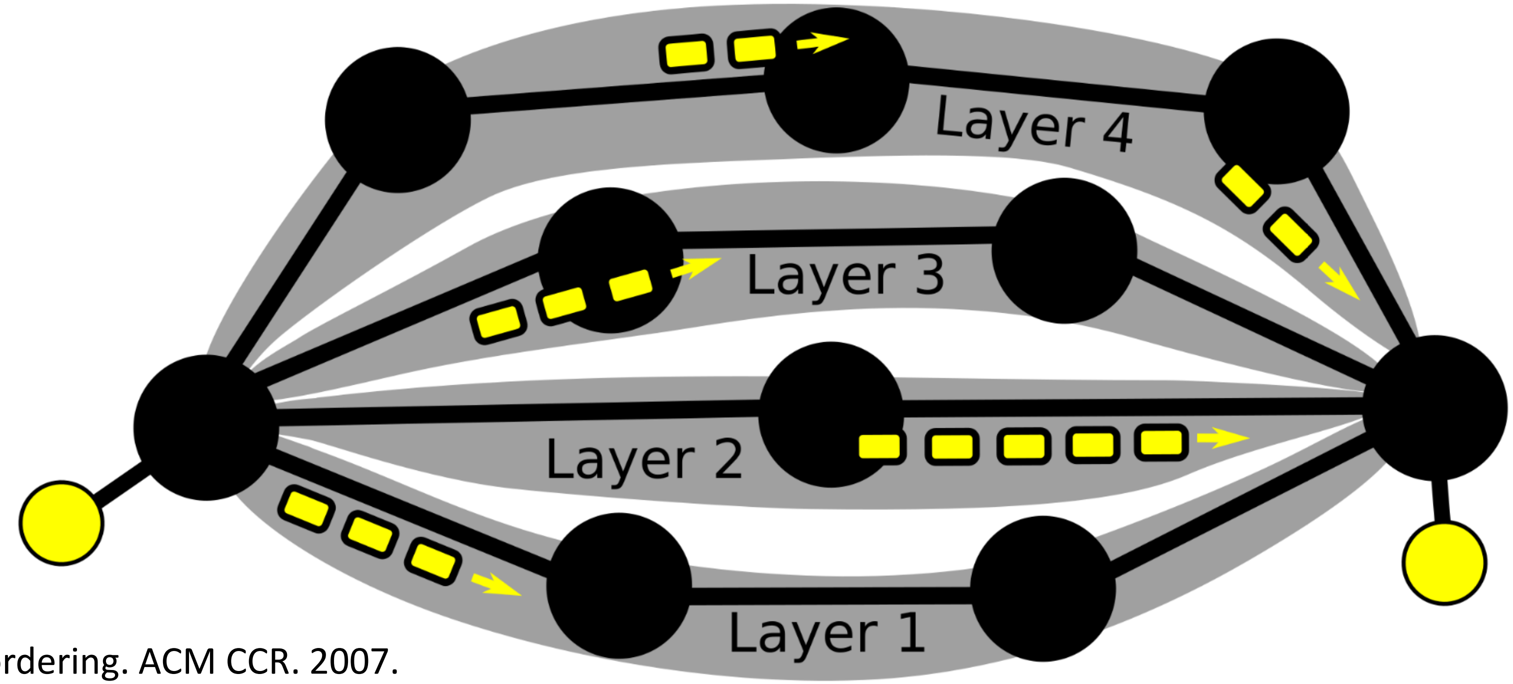
[1] S. Kandula et al. Dynamic load balancing without packet reordering. ACM CCR. 2007.
[2] E. Vanini et al. Let It Flow: Resilient Asymmetric Load Balancing with Flowlet Switching. NSDI'17.

FatPaths Architecture: Load Balancing with Flowlets

A flowlet [1,2] is a sequence of packets within one flow, separated from other flowlets by sufficient time gaps, which prevents packet reordering at the receiver.

How to load balance traffic over different paths of different lenghts?

Very simple load balancing: when one detects congestion, a router just picks a random path (layer) for the packets to be sent (they become a new flowlet)



[1] S. Kandula et al. Dynamic load balancing without packet reordering. ACM CCR. 2007.
[2] E. Vanini et al. Let It Flow: Resilient Asymmetric Load Balancing with Flowlet Switching. NSDI'17.

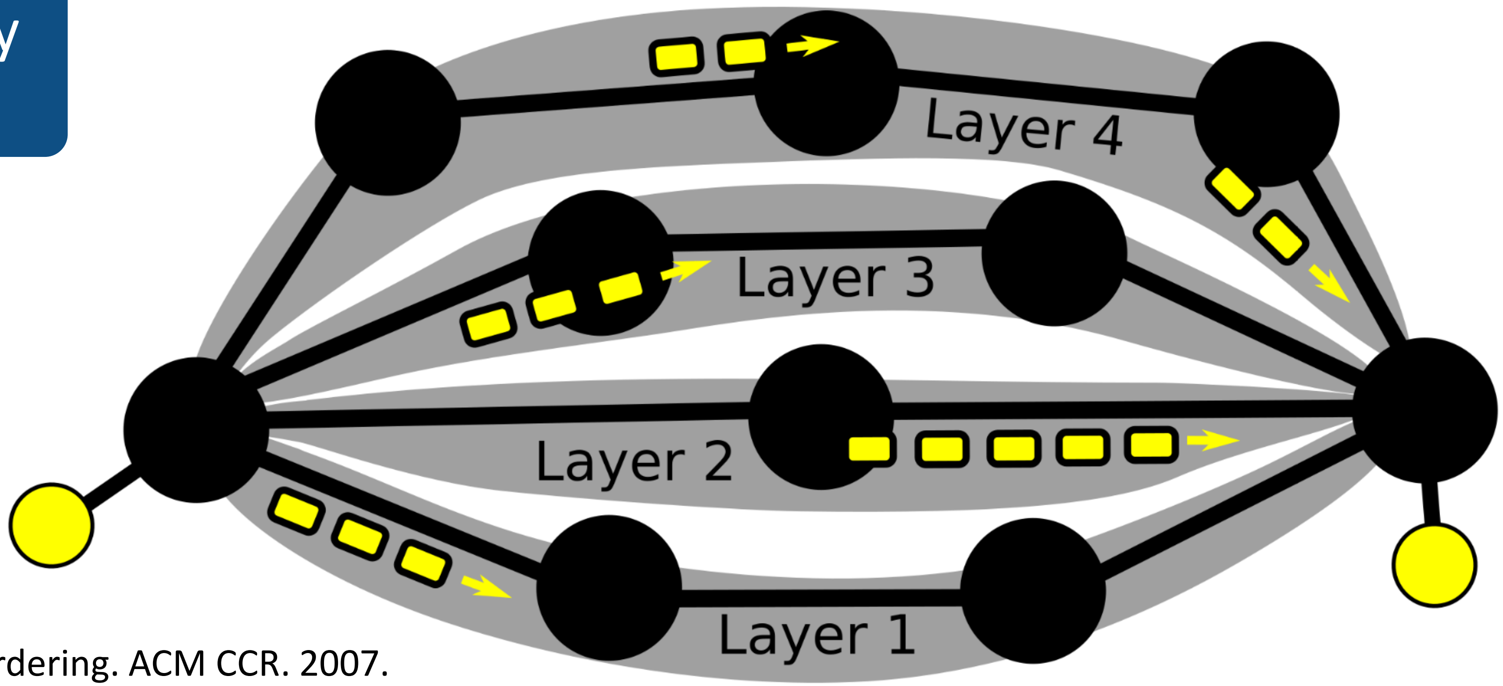
FatPaths architecture: load balancing with flowlets

A flowlet [1,2] is a sequence of packets within one flow, separated from other flowlets by sufficient time gaps, which prevents packet reordering at the receiver.

How to load balance traffic over different paths of different lenghts?

Size of flowlets changes automatically based on conditions in the network

Very simple load balancing: when one detects congestion, a router just picks a random path (layer) for the packets to be sent (they become a new flowlet)



[1] S. Kandula et al. Dynamic load balancing without packet reordering. ACM CCR. 2007.
[2] E. Vanini et al. Let It Flow: Resilient Asymmetric Load Balancing with Flowlet Switching. NSDI'17.

FatPaths architecture

Design details

FatPaths architecture

Design details

```
L = {E} //Init a set of layers L; we start with E that corresponds to σ_1.
Π = {π_1(V), ..., π_{n-1}(V)} //Generate n-1 random permutations of vertices.

//Init a matrix W containing weights of edges (u,v) ∈ E.
W = {[w_uv] | ∀u,v ∈ V : w_uv = 0}

foreach π ∈ Π do: //One iteration of the main loop derives one layer.
  E' = create_layer(π, E, W, L_min, L_max) //Generate a layer.
  L = L ∪ {E'} //Record the layer

//Derive a layer that corresponds to some σ_i.
create_layer(π, E, W, L_min, L_max):
  //A condition "π(u) < π(v)" ensures layer's acyclicity.
  V = {(u,v) ∈ V × V : π(u) < π(v)}
  //Init a priority queue Q. One queue element is a pair of vertices from V.
  Q.init(V)
  incidence_G =incidence_matrix(G) //Generate an incidence matrix of G.
  p_cnt = 0 //Init path count variable p_cnt.
  while (V ≠ ∅ and p_cnt < M) do:
    (u,v) = Q.pop() //Get a pair of vertices with lowest priority (lowest number of added paths).
    path = (v_1, v_2 ..., v_d) ←find_path(u, v, W, incidence_G, L_min, L_max) //Find path from u to v, with path of
         length ∈ [L_min, L_max], minimizing the sum of edge weights, such that for i < j: π(v_i) < π(v_j) and the
         available edges are given by the incidence_G matrix.
      th exists: //Check if u and v are connected, in the graph given by the incidence_G matrix.
        = p_cnt + 1
      each link ∈ path do:
        = E' ∪ {link} //Add each edge from path to the current layer.
      ach v_i, v_j ∈ path where |i - j| > 1 do:
        idence_G[v_i][v_j] = 0 //Exclude all edges, which will force the traffic from v_1 to v_d to use a
          different path than one that was found.
      each v_i, v_j ∈ path where j - i < L_min do:
        V = V \ (v_i, v_j) // Even if an additional path from v_i to v_j of length at least L_min will be added,
          there will still exist this shorter path between these vertices (the one contained by the
          path (v_1, v_2, ... v_d)), therefore the pair (v_i, v_j) should be further excluded from the V set.

ath(src, dst, W, incidence_G, L_min, L_max):
  path = null //Init variable containing the path with lowest cost.
   //Init an empty queue.
  (src)
  Q ≠ ∅ do:
  = Q.pop()
  th.last() == dst then:
```

Creating layers

```
L = {E} //Init a s
P = {π_1(V), ..., π_n
foreach π ∈ P do: //One iter
  E' = {}; foreac
    //P
             edge to the layer
```

Deploying layers (VLAN, ...)

FatPaths architecture

Design details

```
L = {E} //Init a set of layers L; we start with E that corresponds to σ1.
Π = {π1(V), ..., πn−1(V)} //Generate n−1 random permutations of vertices.

//Init a matrix W containing weights of edges (u,v) ∈ E.
W = {[wuv] | ∀u,v ∈ V : wuv = 0}
```

Creating layers

Deploying layers (VLAN, ...)

```
L = {E} //Init a s...
P = {π1(V), ..., πn−...
foreach π ∈ P do: //One iter...
  E' = {}; foreach ...
```

Populating forwarding entries

```
//Input: E'_i, i ∈ {1,...,n}: the specification of each layer. E'_i are router-router links in each layer i,
//produced by FatPaths layer construction algorithms specified in Listing 1 or Listing 2.
//Output: F_{i,s}: a forwarding table in router s ∈ V within layer i (1 ≤ i ≤ n).
//F_{i,s}[t] ∈ {1,...,k'} is a port number (in router s ∈ V) that leads to router t ∈ V (within layer i (1 ≤ i ≤ n)).

//Compute shortest paths between routers in each layer i. First, initialize the function σ_i.
//σ_i(s,t) is a port number (in router s) that leads to router t (in layer i); d is an auxiliary structure.
foreach i ∈ {1,...,n} do: //For each layer...
  foreach (s,t) ∈ V × V do: //For each router pair...
    if s == t then:
      d_st = 0 //The distance between u and v is zero, if u == v.
      σ_i(s,t) = s
    else if (s,t) ∈ E'_i then:
      d_st = 1 //The distance between directly connected routers is 1.
      σ_i(s,t) = t
    else:
      d_st = +∞ //The initial distance between non-adjacent routers is infinity.
      σ_i(s,t) = null

  //For each layer, derive the routing functions:
  foreach z ∈ |V| do:
    foreach s ∈ |V| do:
      foreach t ∈ |V| do:
        if d_st > d_sz + d_zt then:
          d_st = d_sz + d_zt
          σ_i(s,t) = σ_i(s,z)

  //Once σ_i is computed, populate a f...
  foreach s ∈ V do: //For every router...
    F_s = {} //Initialize the forwarding...

  //Build a forwarding table within la... ...the previously devised structures:
  foreach s ∈ V do:
    foreach t ∈ V, t ≠ s do:
      F_s[t] = σ_i(s,t)
```

FatPaths architecture

Design details

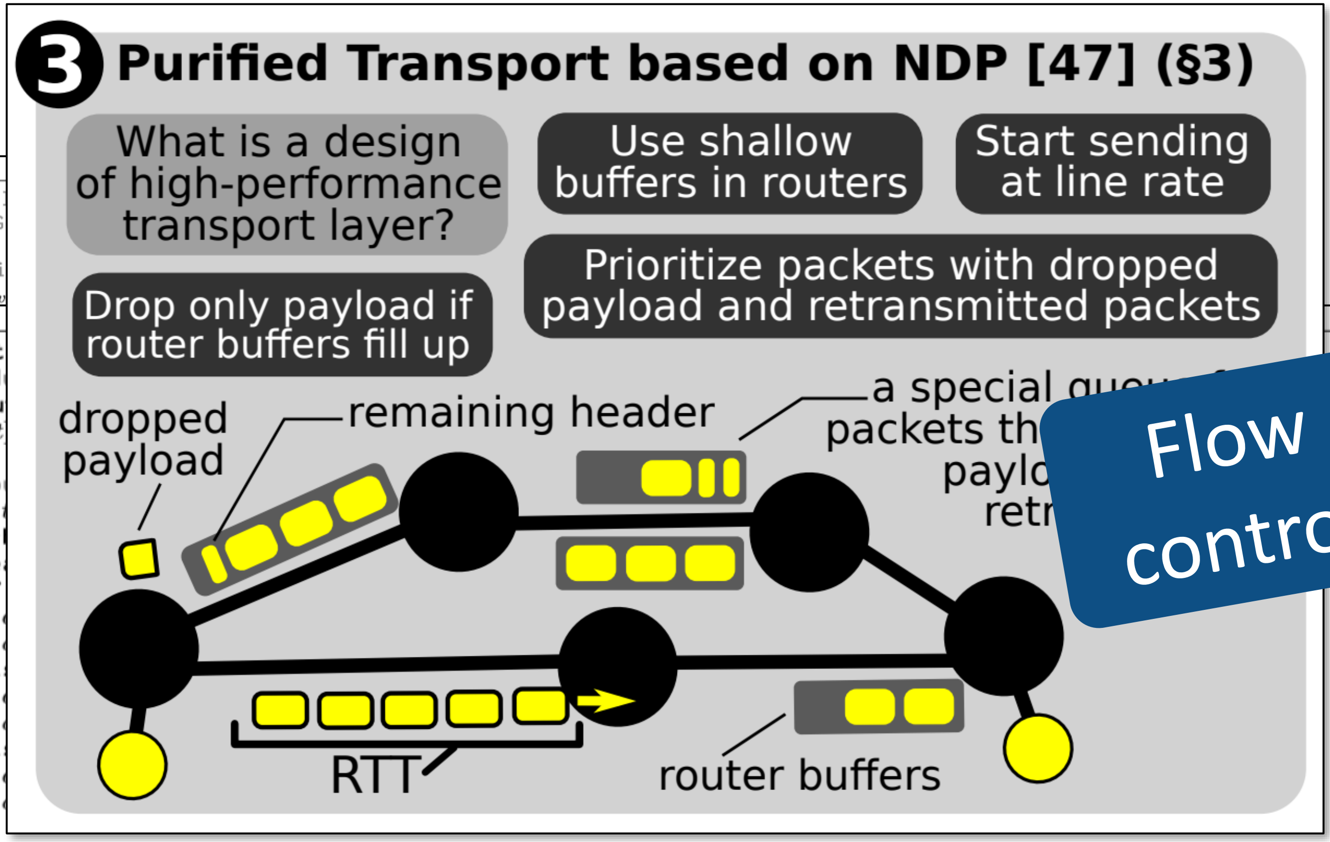
Creating layers

Deploying layers (VLAN, ...)

Populating forwarding entries

Flow control

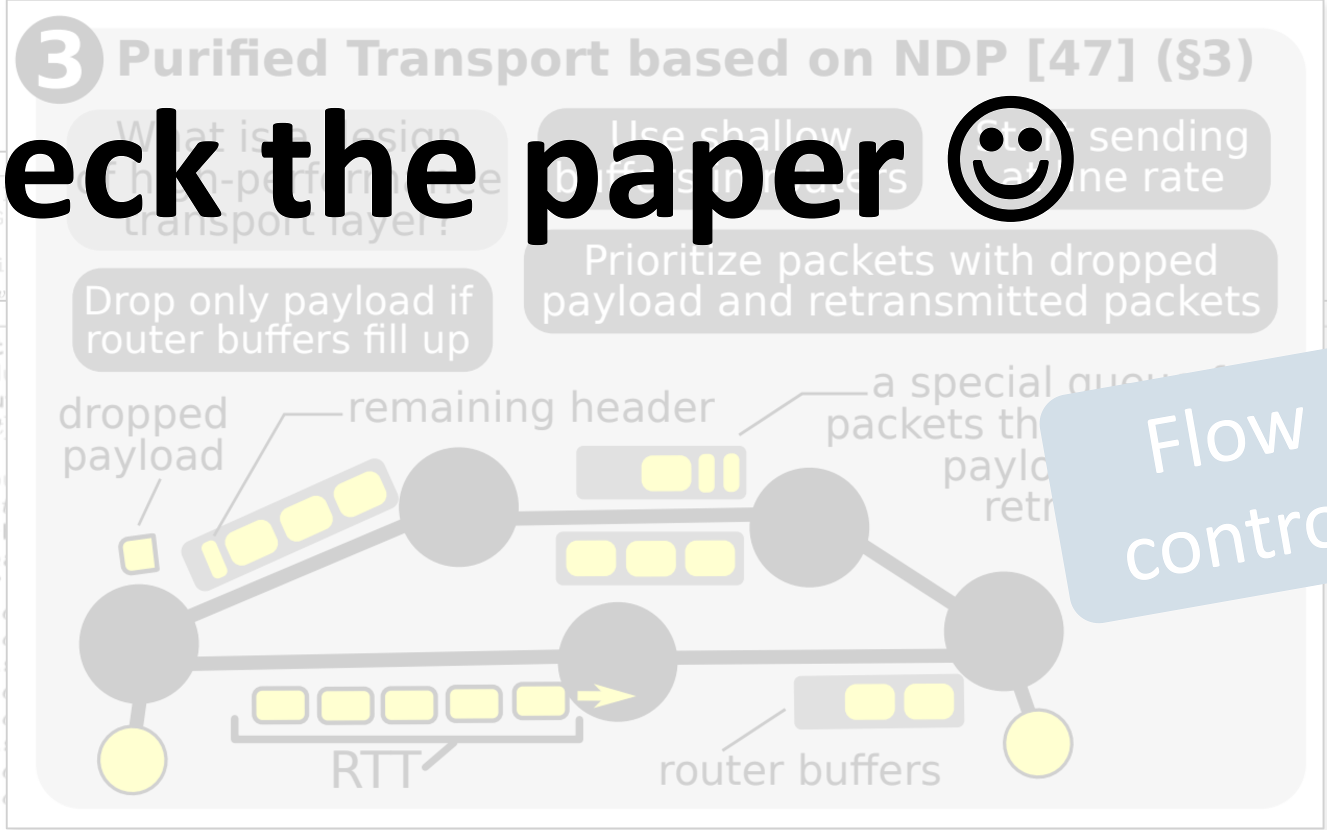
```
//For each layer, derive the routing functions:
foreach z ∈ |V| do:
  foreach s ∈ |V| do:
    foreach t ∈ |V| do:
      if d_st > d_sz + d_zt then:
        d_st = d_sz + d_zt
        σ_i(s,t) = σ_i(s,z)

//Once σ_i is computed, populate a f
foreach s ∈ V do: //For every router
  F_s = {} //Initialize the forwarding

//Build a forwarding table within la      the previously devised structures:
foreach s ∈ V do:
  foreach t ∈ V, t ≠ s do:
    F_s[t] = σ_i(s,t)
```



FatPaths architecture

Design details

Check the paper ☺



Flow control

Creating layers

Deploying layers (VLAN, ...)

Populating forwarding entries

FatPaths architecture

Design details

Check the paper ☺

Key design insight: Layered routing enables (1) easy encoding of the diversity of multiple minimal and non-minimal paths, and (2) enables simple and robust load balancing

EVALUATION

EVALUATION

<u>Question 1:</u> Can we get more performance on low-diameter networks than on Fat trees?

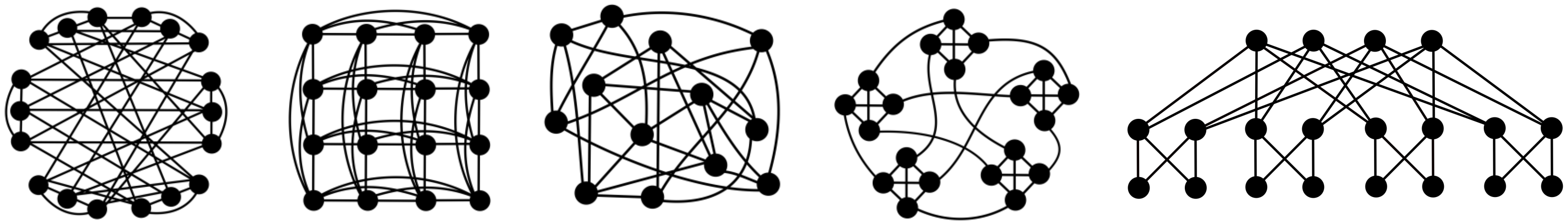
EVALUATION

<u>Question 1:</u> Can we get more performance on low-diameter networks than on Fat trees?

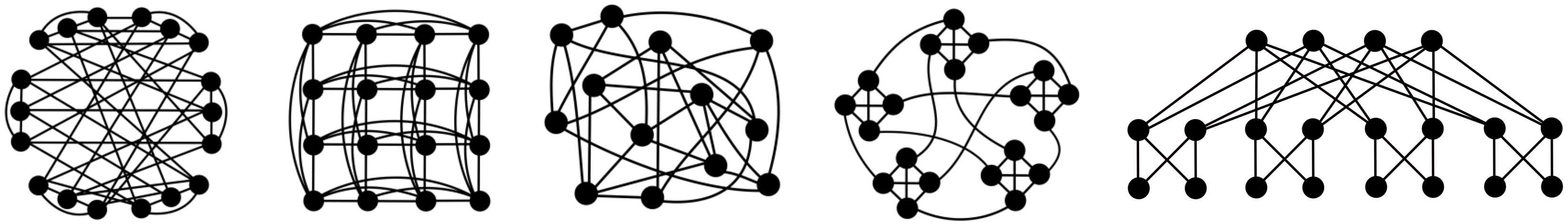
<u>Question 2:</u> Does Fatpaths give more performance than existing routing schemes?

EVALUATION

EVALUATION

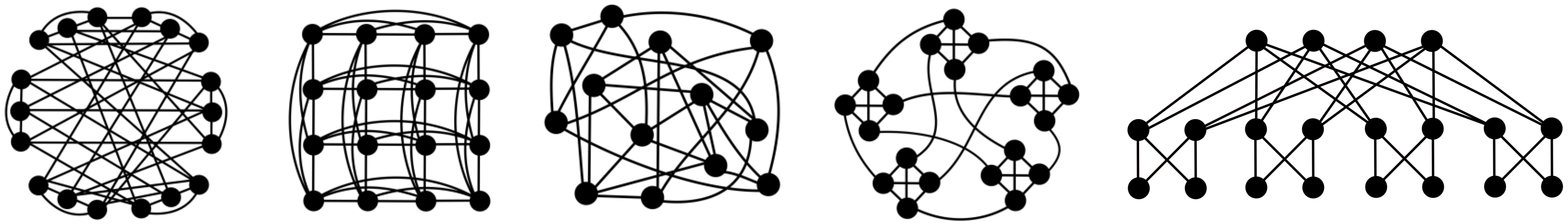


Evaluation



We pick Ethernet as a setting to accelerate as many Top500 systems as possible

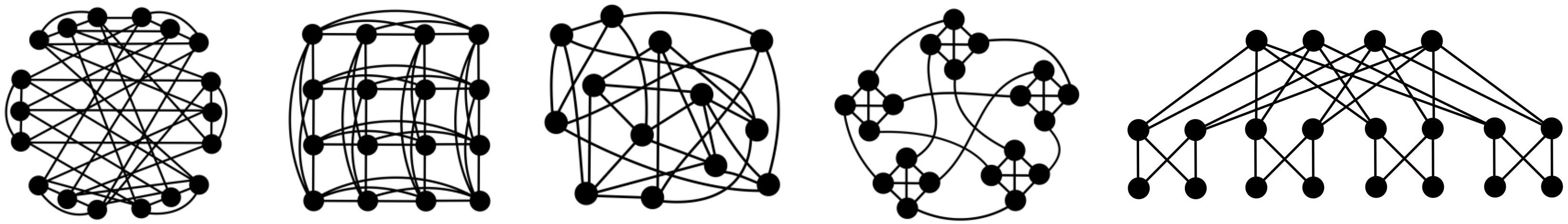
EVALUATION



<u>We pick Ethernet</u> as a setting to accelerate as many Top500 systems as possible

(more than 50% of systems in Top500 use Ethernet. However, they are not as efficient as InfiniBand and others (details in the paper)

EVALUATION



<u>We pick Ethernet</u> as a setting to accelerate as many Top500 systems as possible

(more than 50% of systems in Top500 use Ethernet. However, they are not as efficient as InfiniBand and others (details in the paper)

<u>Setting 1 „Bare Ethernet”</u>

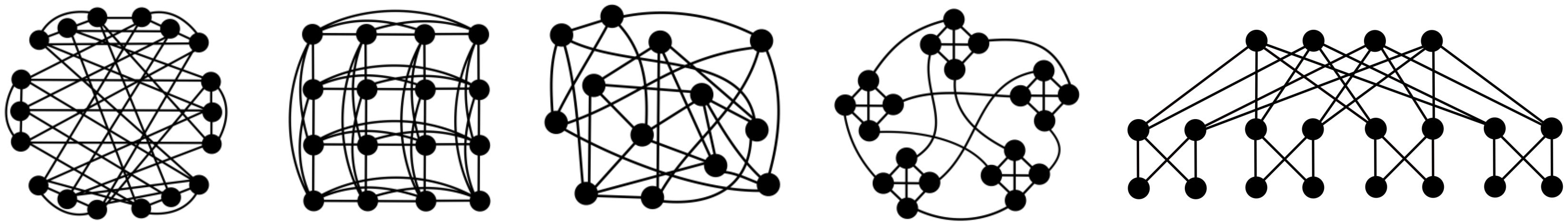
Ethernet, but no TCP.

Simulator: htsim [1].

NDP [1]: a very recent baseline, originally for fat trees.

[1] M. Handley et al. Re-architecting datacenter networks and stacks for low latency and high performance. SIGCOMM’17.

EVALUATION



We pick Ethernet as a setting to accelerate as many Top500 systems as possible

(more than 50% of systems in Top500 use Ethernet. However, they are not as efficient as InfiniBand and others (details in the paper)

Setting 1 „Bare Ethernet"
Ethernet, but no TCP.
Simulator: htsim [1].

NDP [1]: a very recent baseline, originally for fat trees.

Setting 2 „Full TCP"
Standard TCP and related.
Simulator: OMNeT++ [2].

ECMP [3]: traditional static load balancing.
LetFlow [4]: recent adaptive load balancing.

[1] M. Handley et al. Re-architecting datacenter networks and stacks for low latency and high performance. SIGCOMM'17.
[2] A. Varga et al. The OMNeT++ discrete event simulation system. ESM'01.
[3] C. Hopps. Analysis of an Equal-Cost Multi-Path Algorithm. RFC2992, 2000.
[4] E. Vanini et al. Let It Flow: Resilient Asymmetric Load Balancing with Flowlet Switching. NSDI'17.

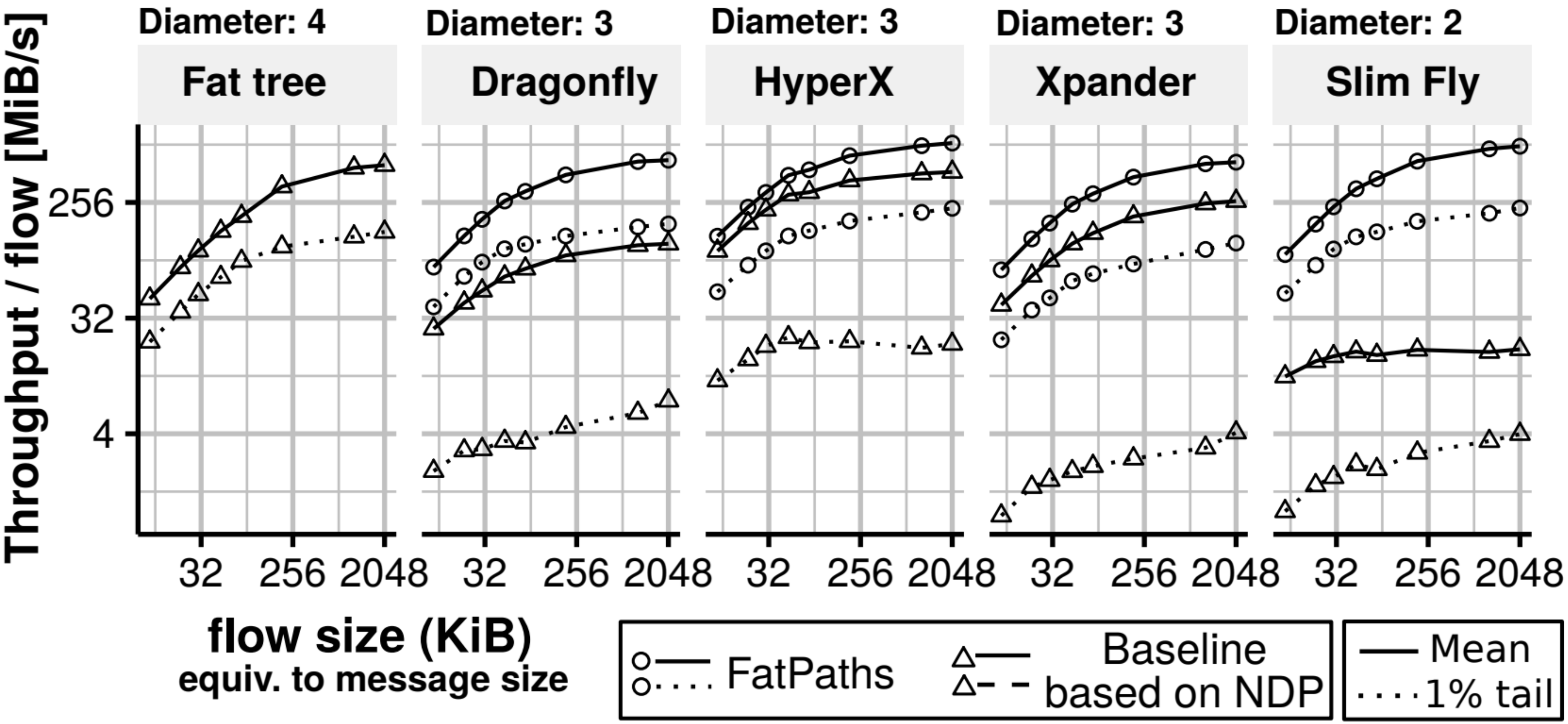
EVALUATION

N ≈ 10,000; comparable cost; stencil (no remapping); “bare Ethernet” setting

NDP: a very recent baseline for fat trees [47]

EVALUATION

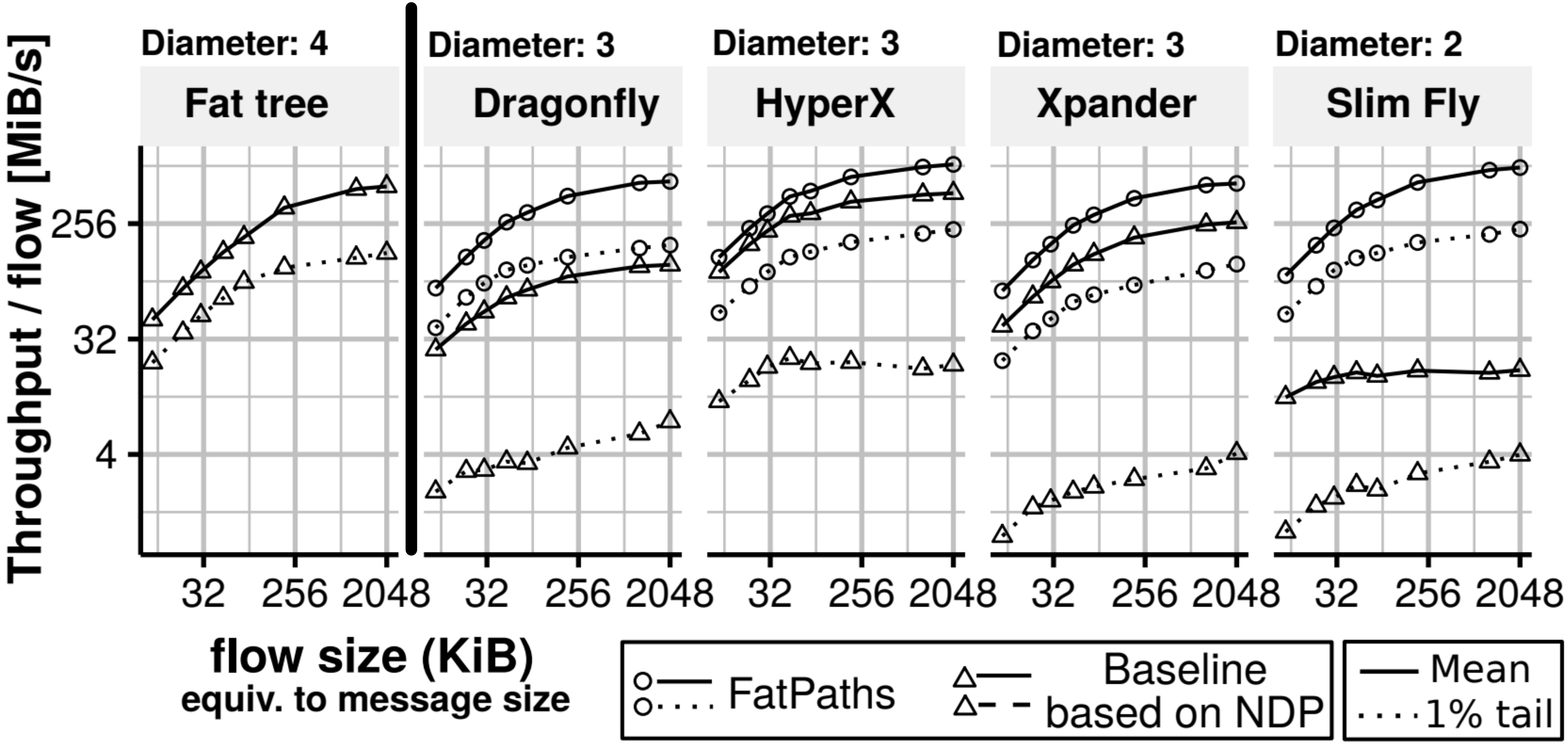
N ≈ 10,000; comparable cost; stencil (no remapping); “bare Ethernet” setting



NDP: a very recent baseline for fat trees [47]

EVALUATION

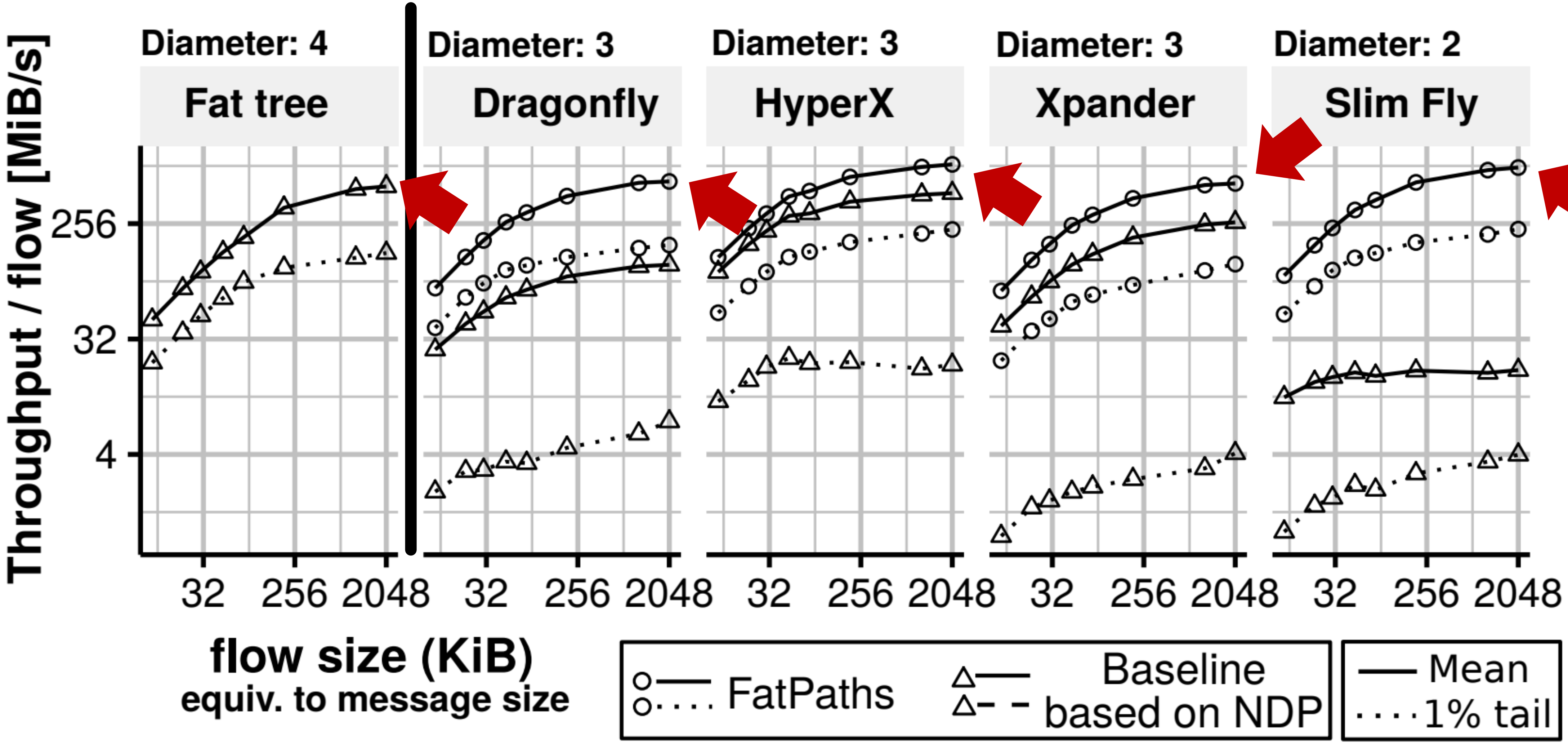
N ≈ 10,000; comparable cost; stencil (no remapping); “bare Ethernet” setting



NDP: a very recent baseline for fat trees [47]

EVALUATION

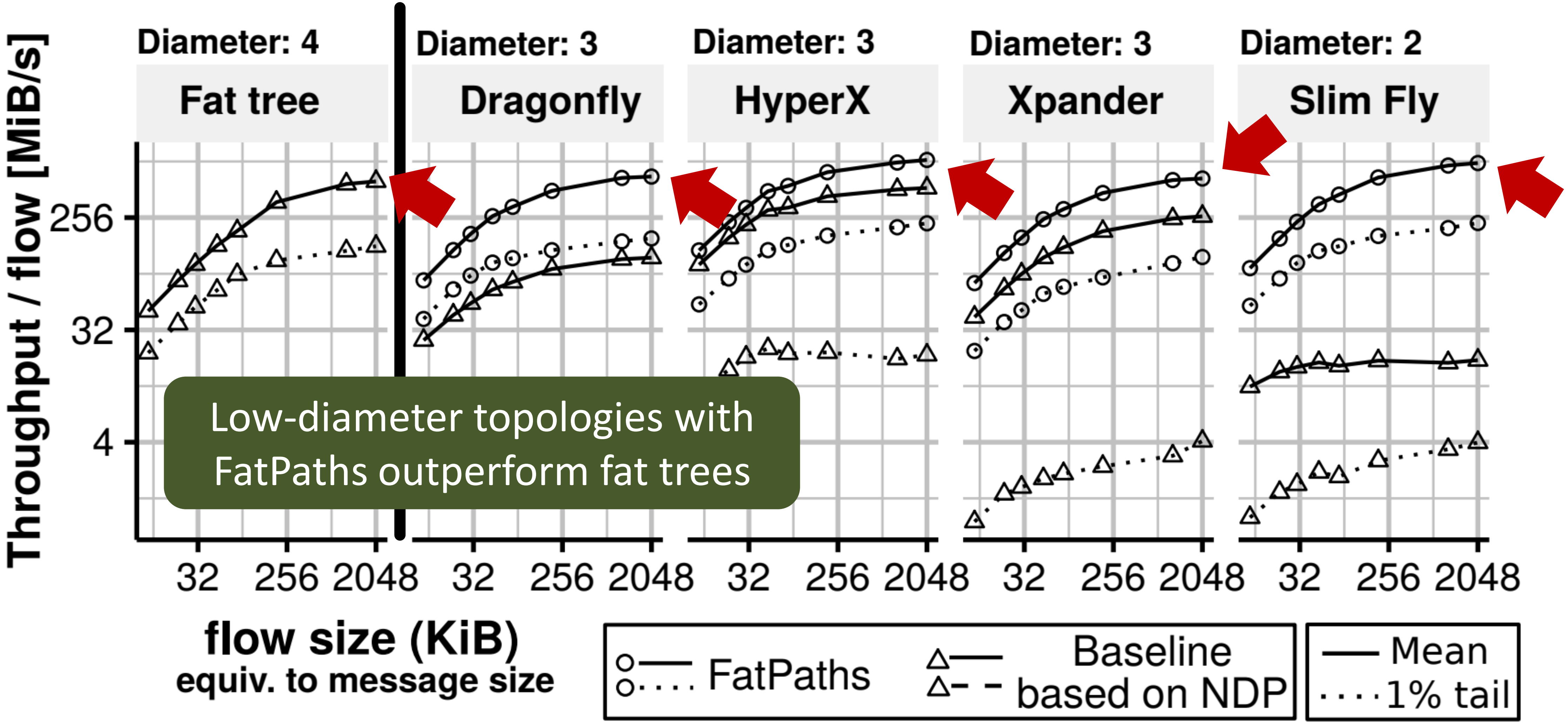
N ≈ 10,000; comparable cost; stencil (no remapping); "bare Ethernet" setting



NDP: a very recent baseline for fat trees [47]

EVALUATION

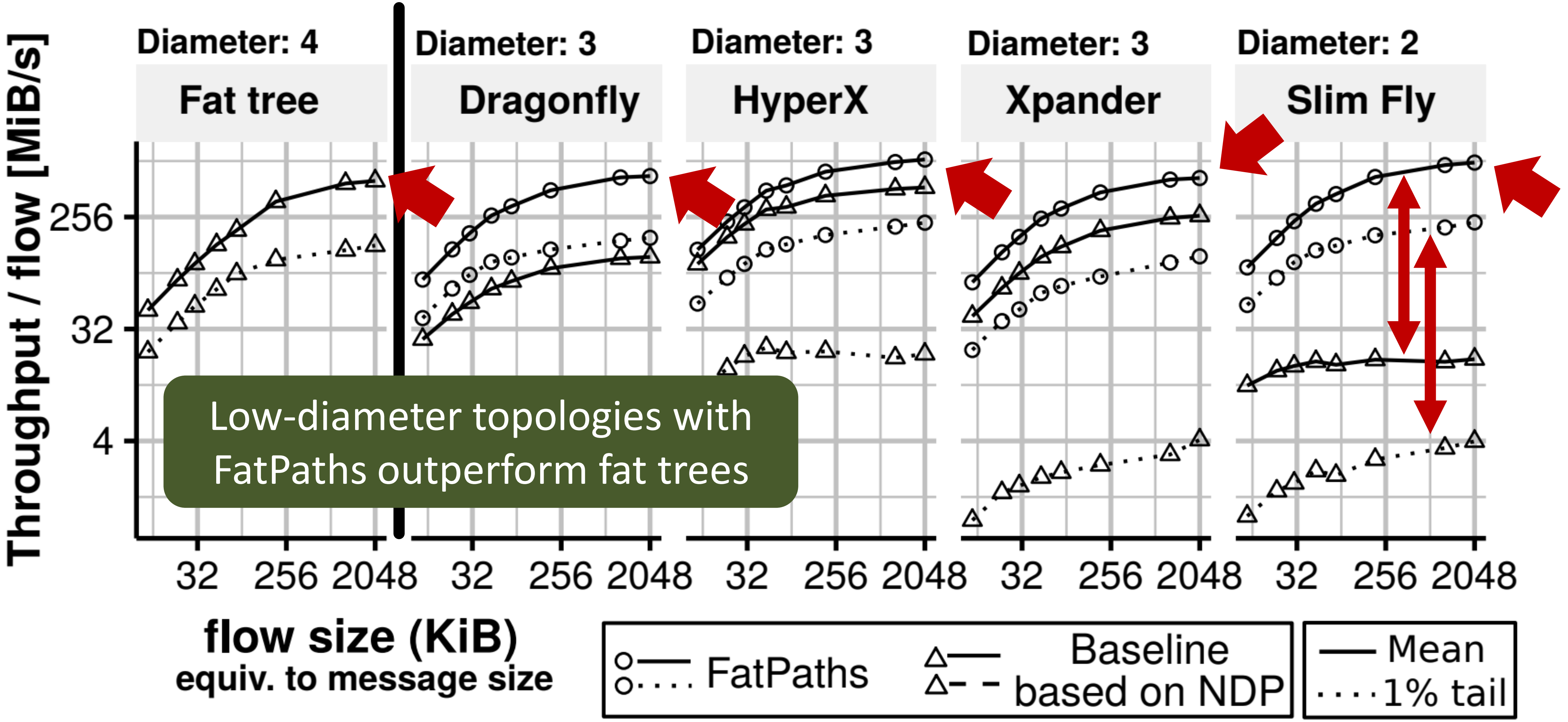
N ≈ 10,000; comparable cost; stencil (no remapping); "bare Ethernet" setting



NDP: a very recent baseline for fat trees [47]

EVALUATION

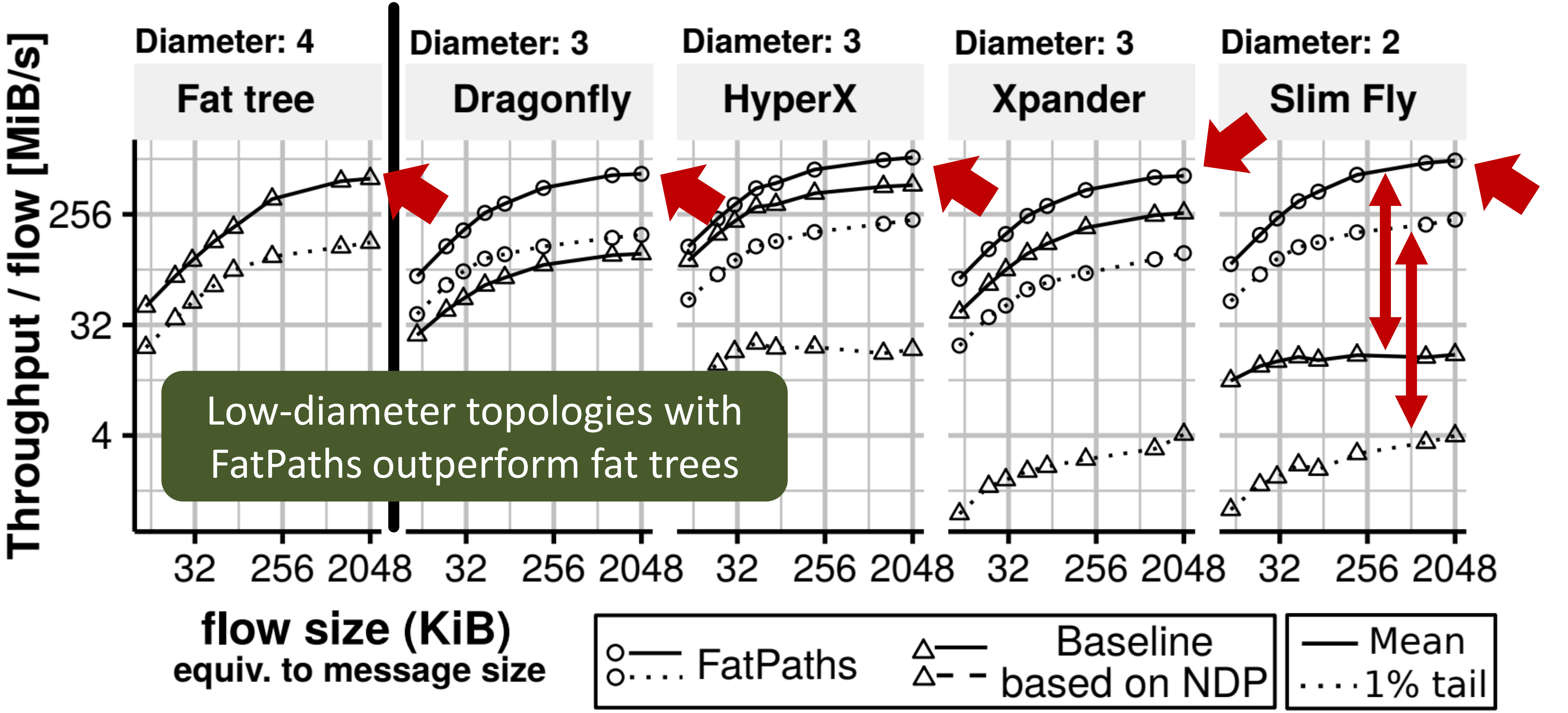
N ≈ 10,000; comparable cost; stencil (no remapping); "bare Ethernet" setting



NDP: a very recent baseline for fat trees [47]

EVALUATION

N ≈ 10,000; comparable cost; stencil (no remapping); "bare Ethernet" setting



NDP: a very recent baseline for fat trees [47]

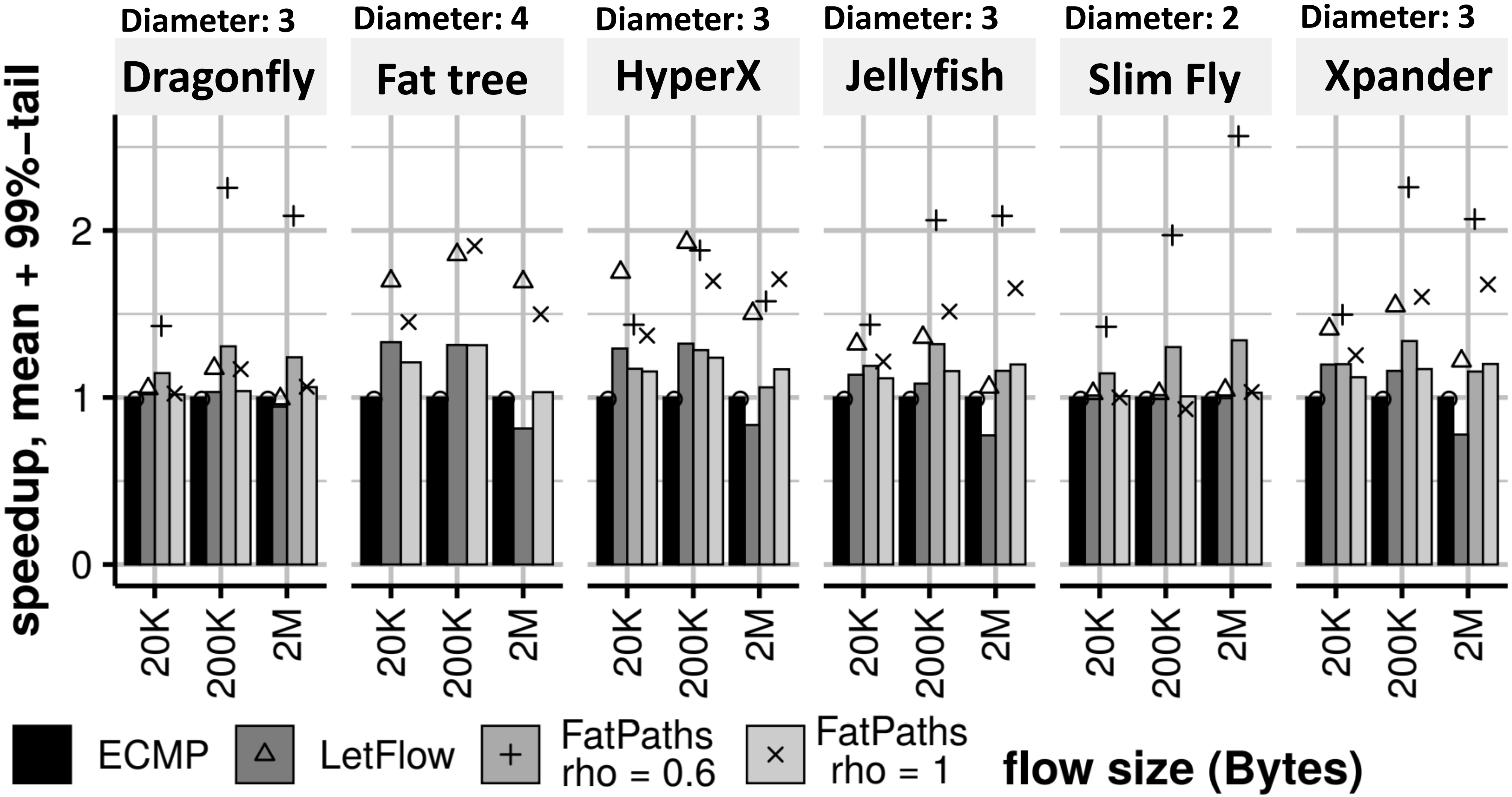
FatPaths outperforms simple NDP executed obliviously on low-diameter networks (up to 30x improvement)

EVALUATION

N ≈ 10,000; comparable cost; random uniform traffic; "full TCP" setting

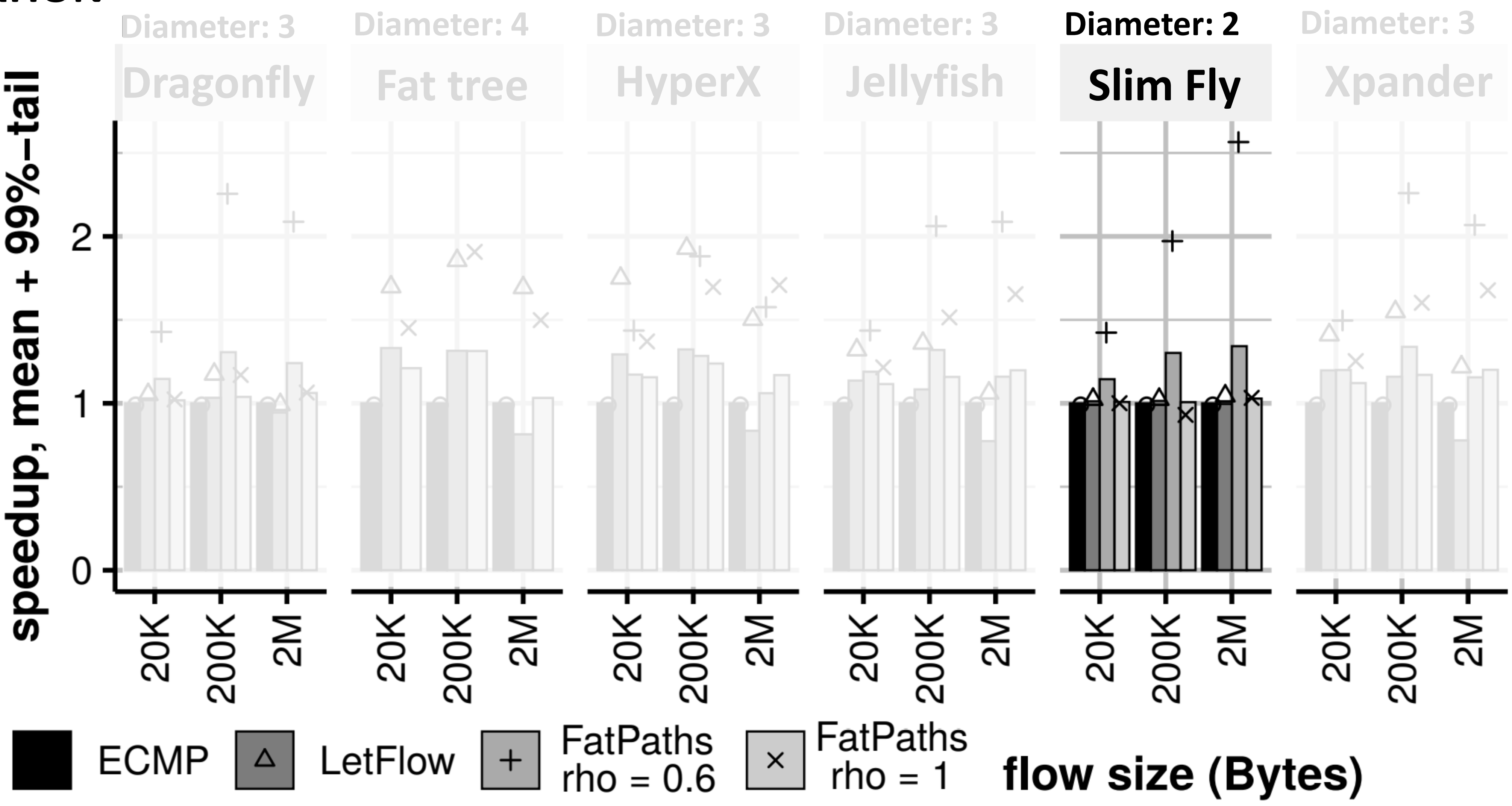
EVALUATION

N ≈ 10,000; comparable cost; random uniform traffic; “full TCP” setting



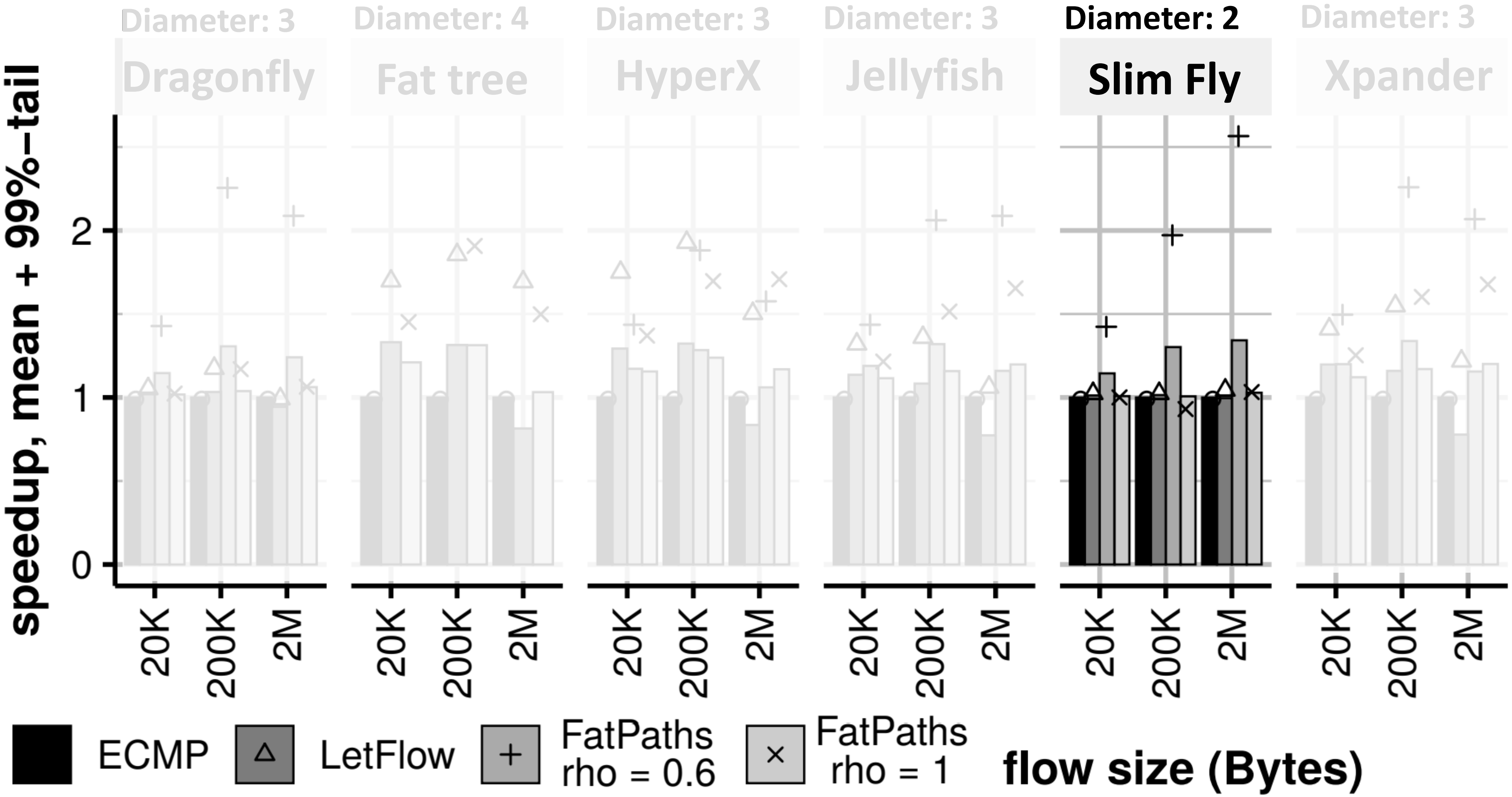
EVALUATION

N ≈ 10,000; comparable cost; random uniform traffic; “full TCP” setting



EVALUATION

N ≈ 10,000; comparable cost; random uniform traffic; “full TCP” setting

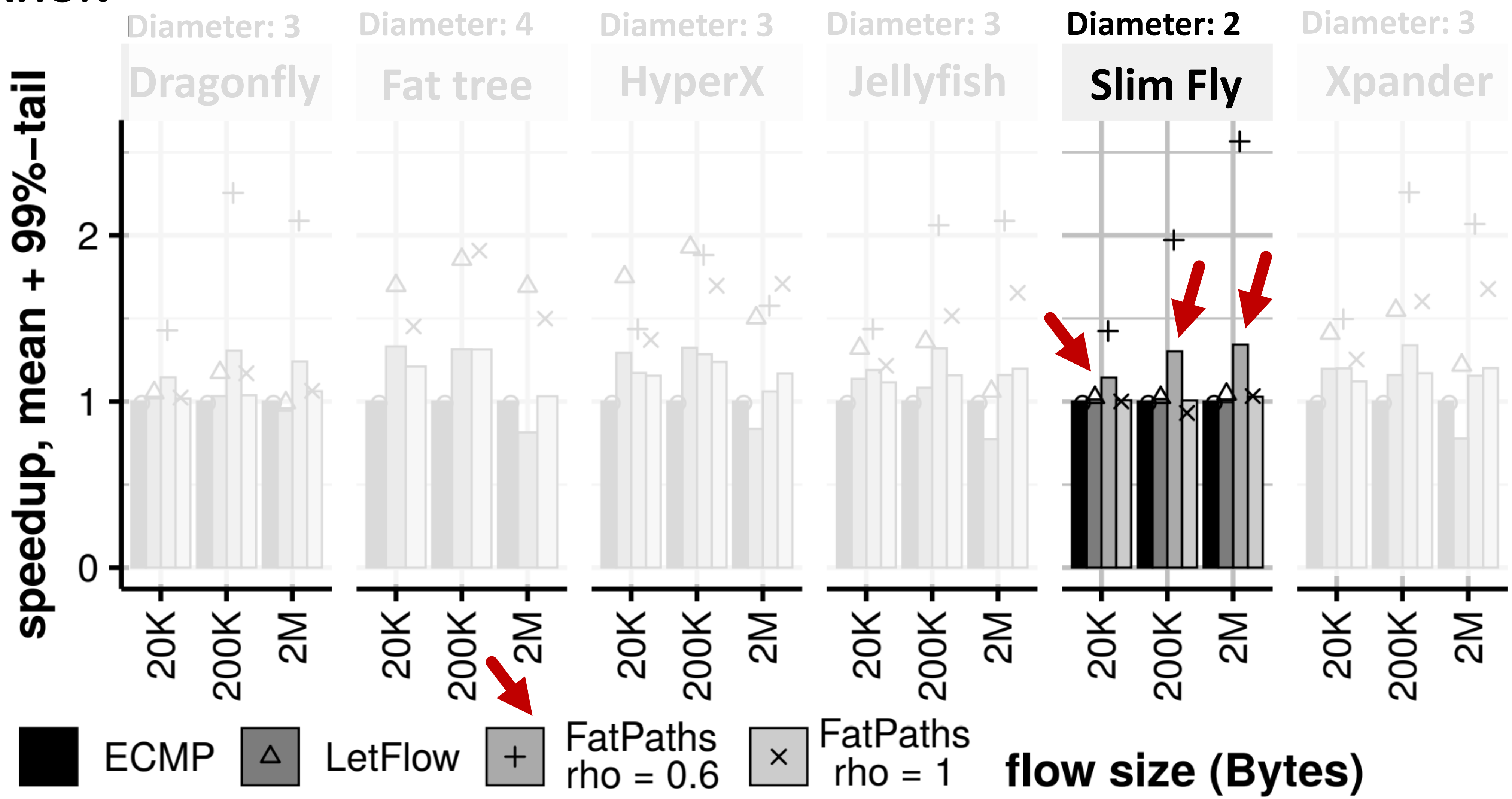


ECMP: traditional static load balancing
LetFlow: recent adaptive load balancing

Speedups are measured over ECMP
rho: fraction of links kept in a layer

EVALUATION

N ≈ 10,000; comparable cost; random uniform traffic; “full TCP” setting

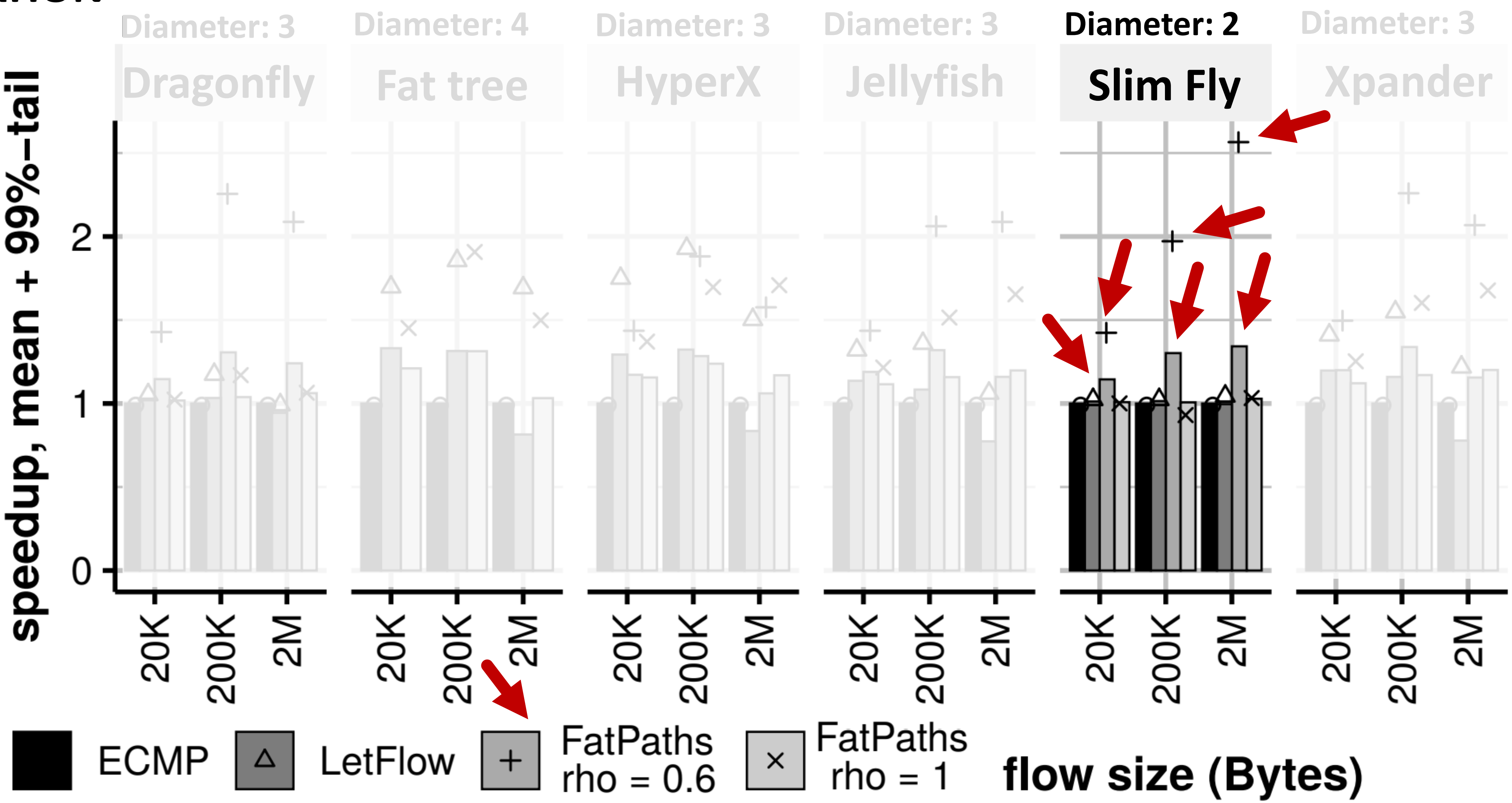


ECMP: traditional static load balancing
LetFlow: recent adaptive load balancing

Speedups are measured over ECMP
rho: fraction of links kept in a layer

EVALUATION

N ≈ 10,000; comparable cost; random uniform traffic; "full TCP" setting

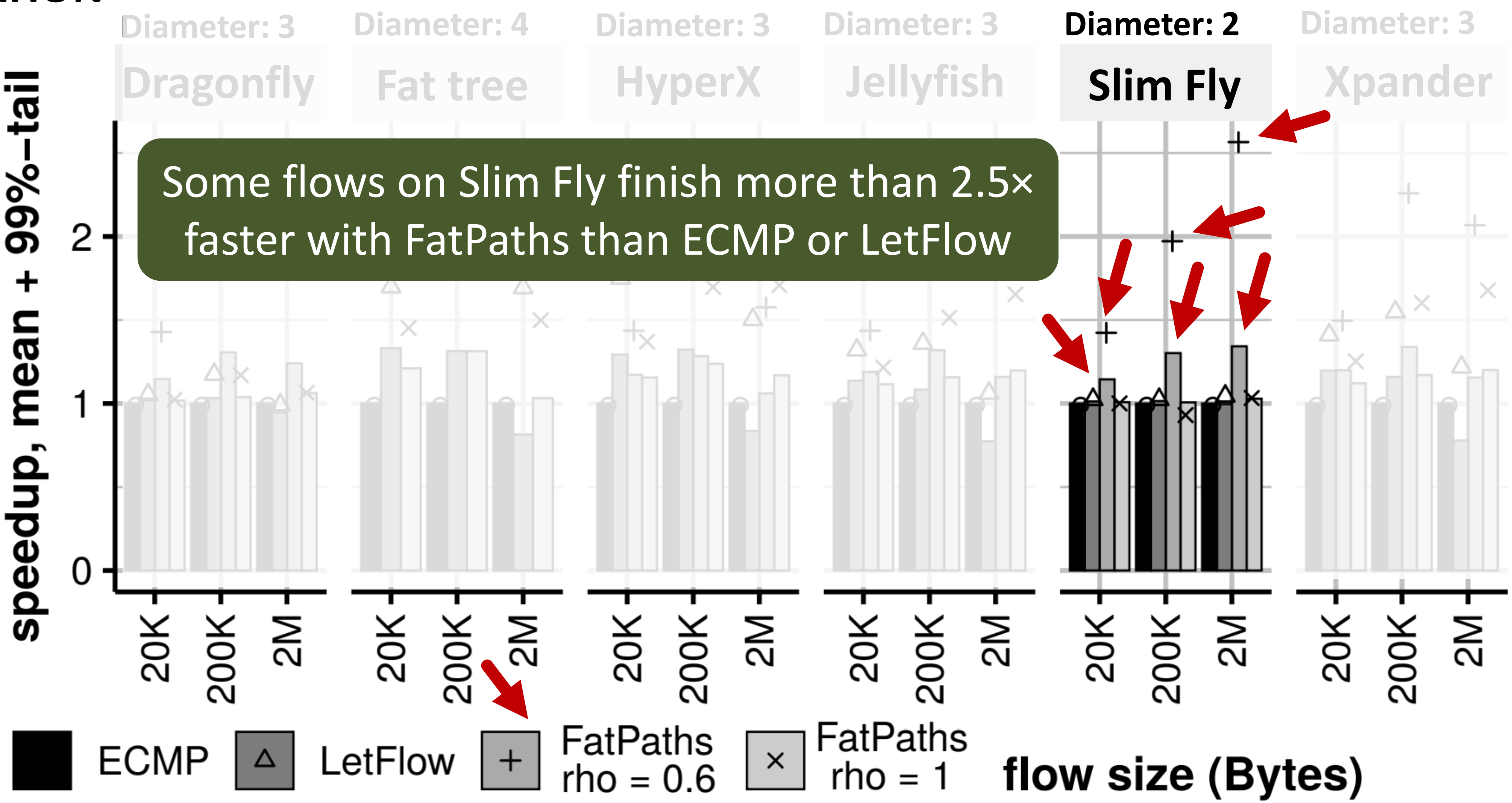


ECMP: traditional static load balancing
LetFlow: recent adaptive load balancing

Speedups are measured over ECMP
rho: fraction of links kept in a layer

EVALUATION

N ≈ 10,000; comparable cost; random uniform traffic; “full TCP” setting



ECMP: traditional static load balancing
LetFlow: recent adaptive load balancing

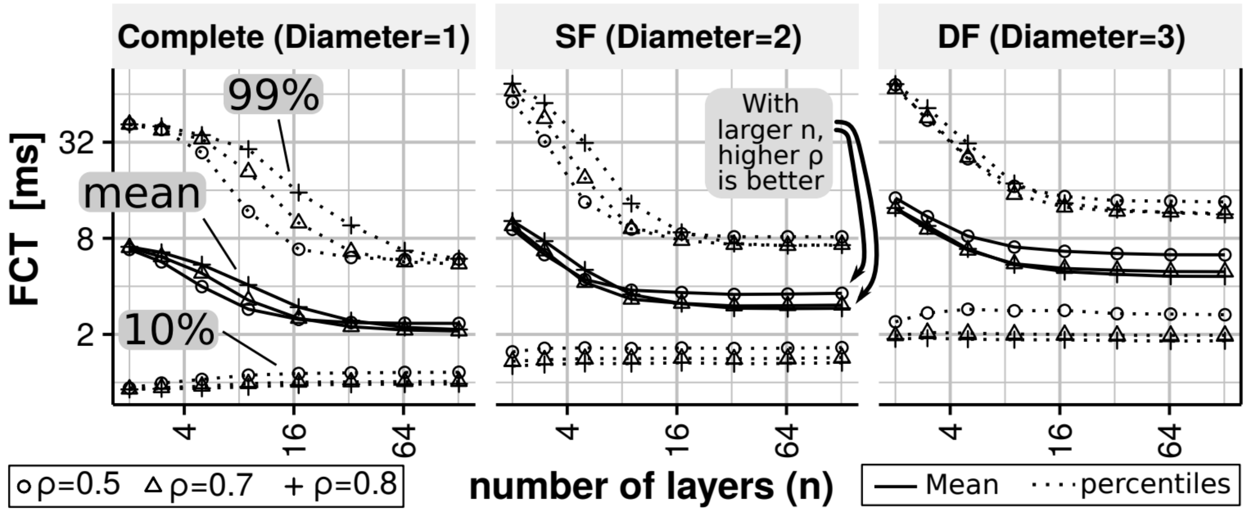
Speedups are measured over ECMP
rho: fraction of links kept in a layer

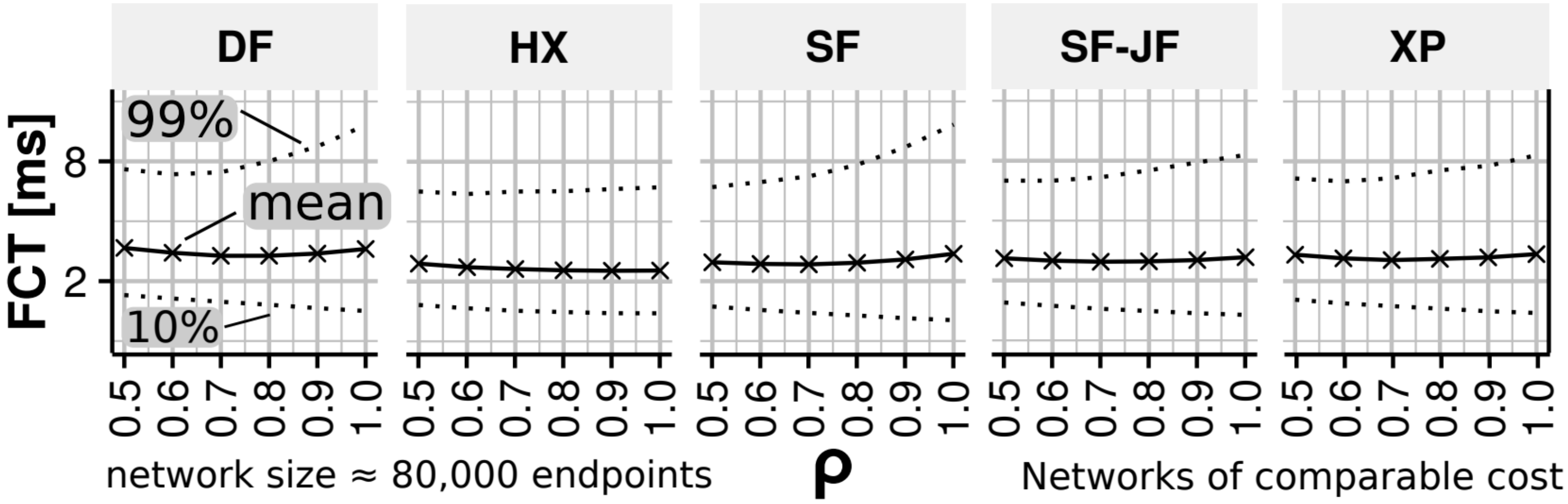
EVALUATION

Other analyzes

EVALUATION

Other analyzes



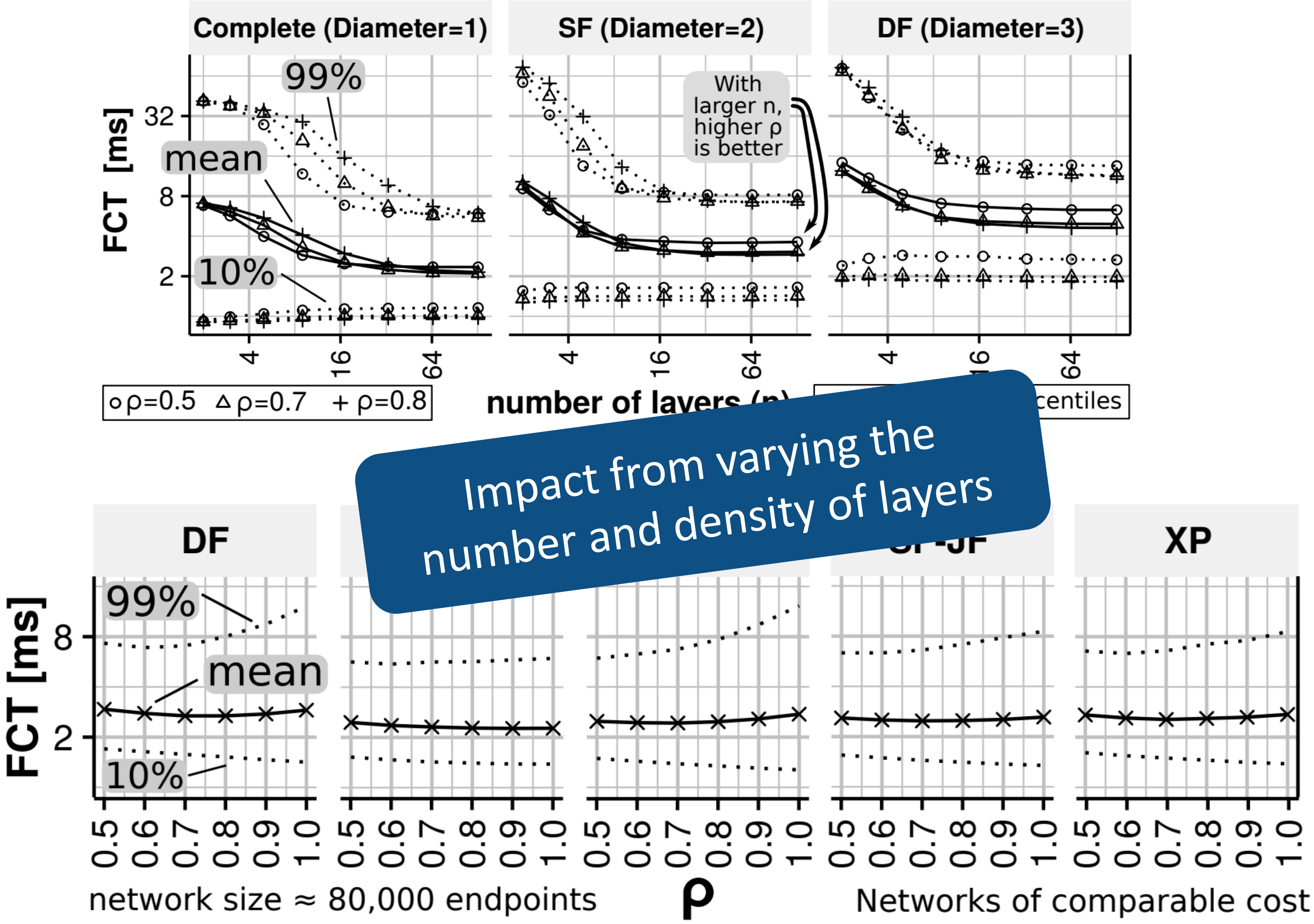


network size ≈ 80,000 endpoints

Networks of comparable cost

EVALUATION

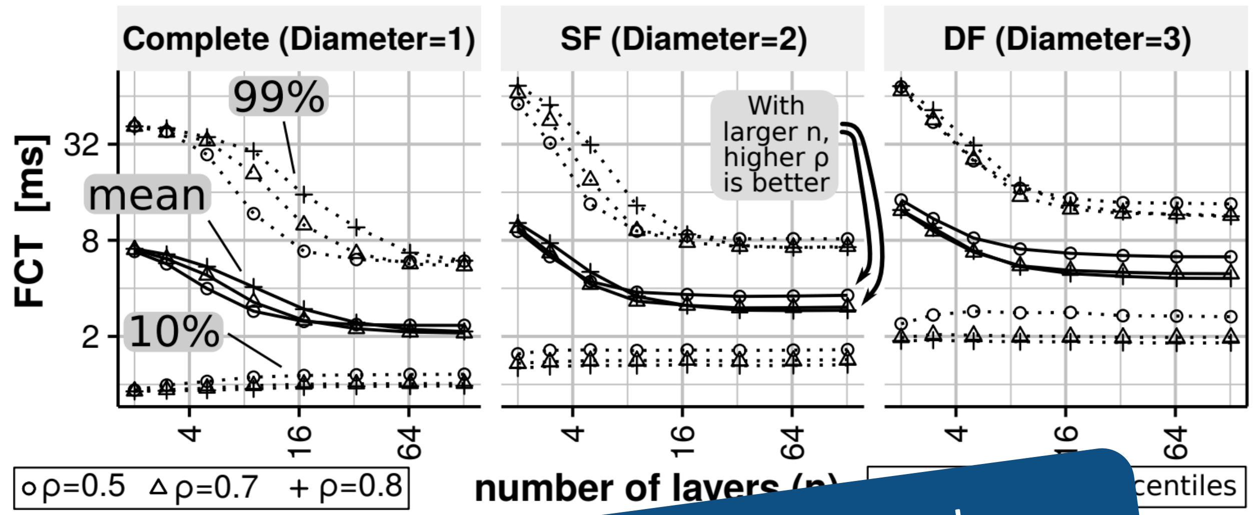
Other analyzes



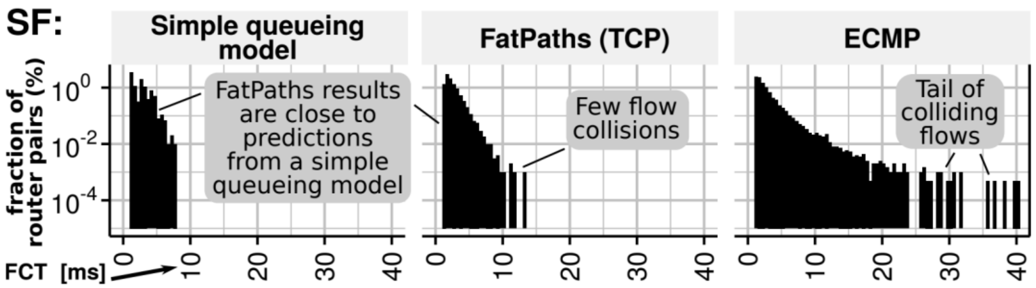
Impact from varying the number and density of layers

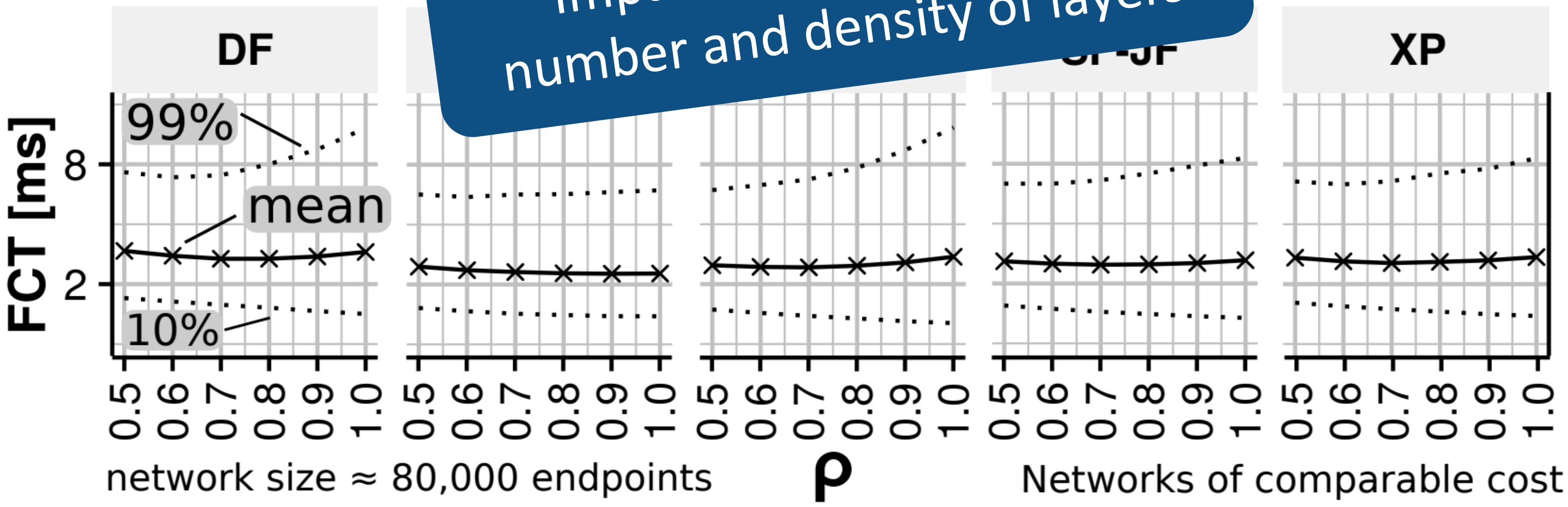
EVALUATION

Other analyzes



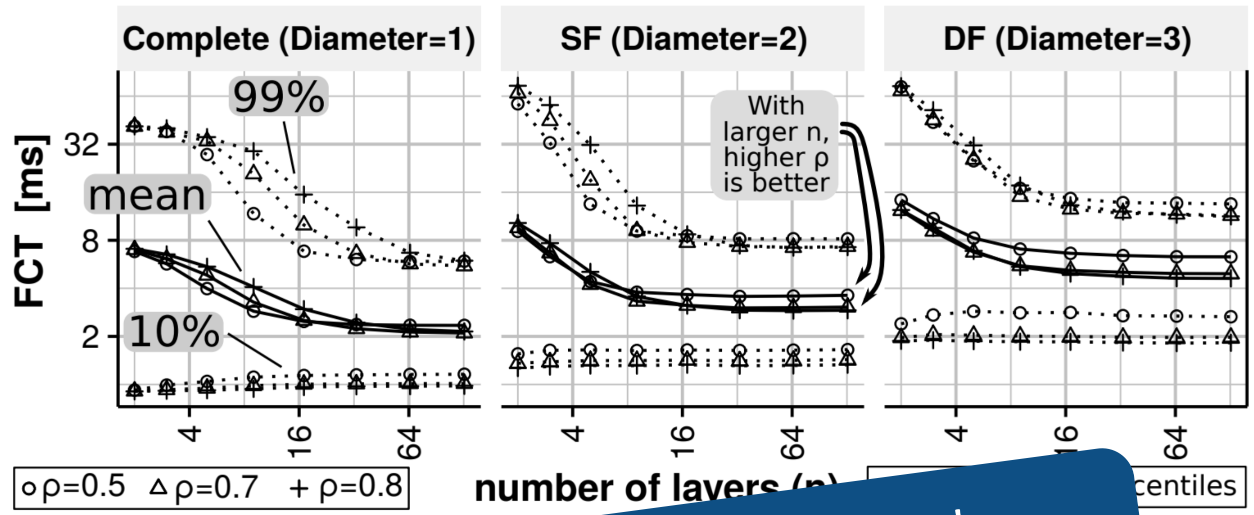
Impact from varying the number and density of layers

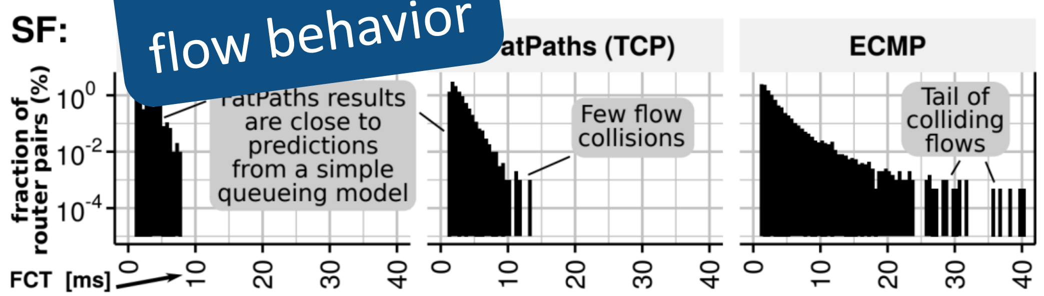




EVALUATION

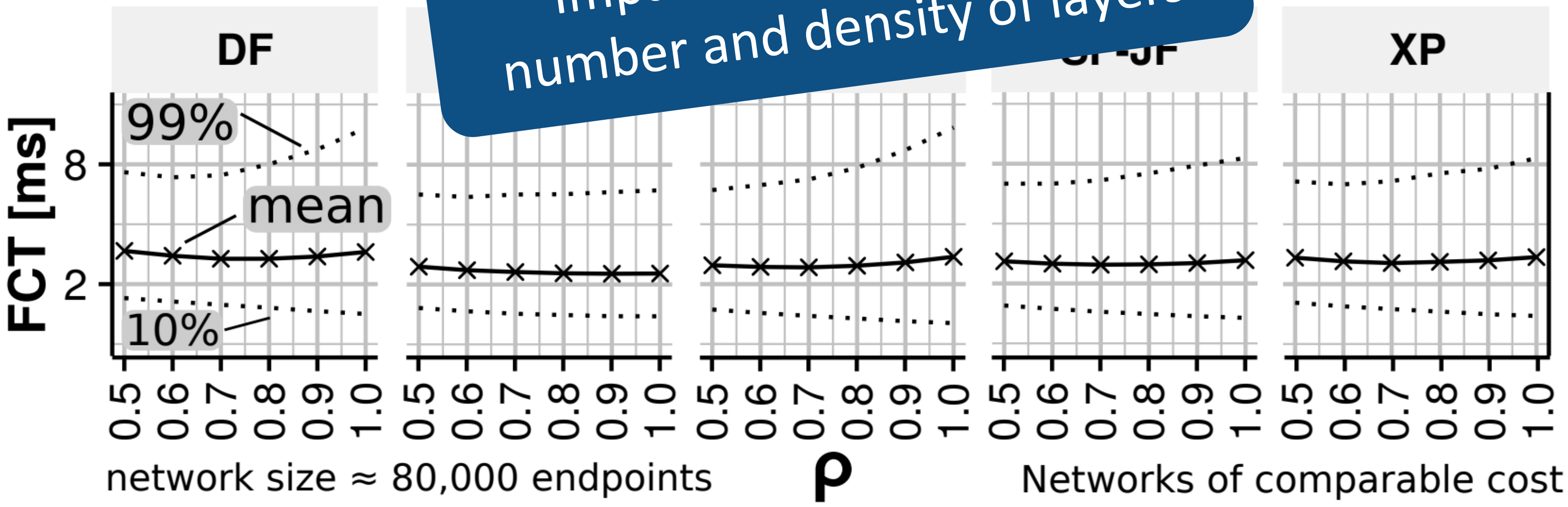
Other analyzes





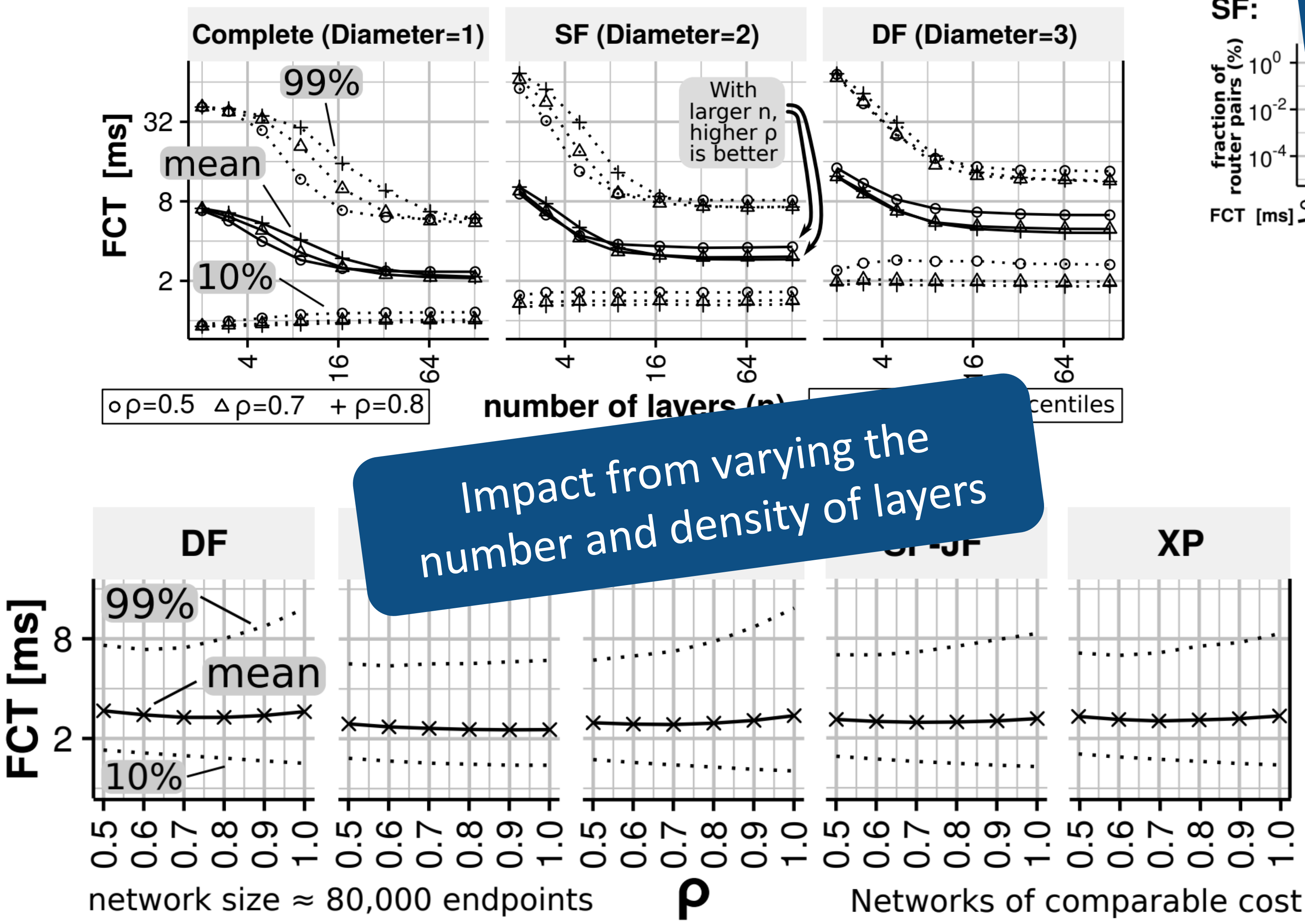
Details of flow behavior

Impact from varying the number and density of layers

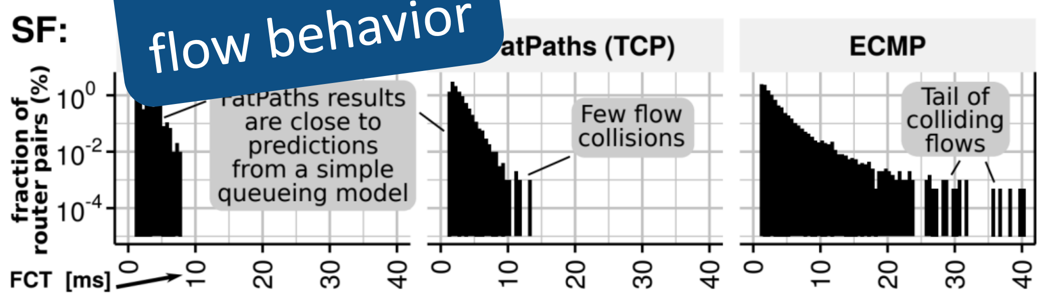


EVALUATION

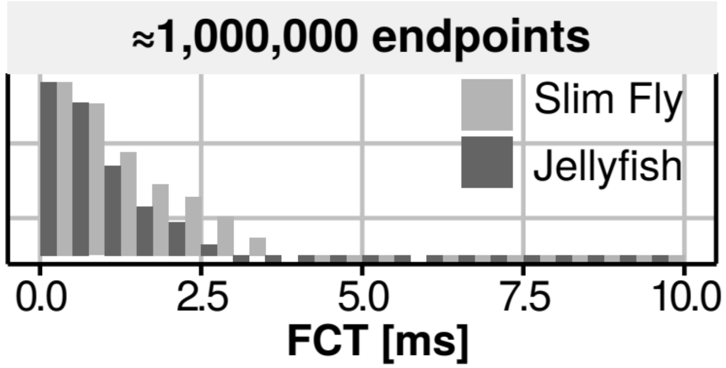
Other analyzes

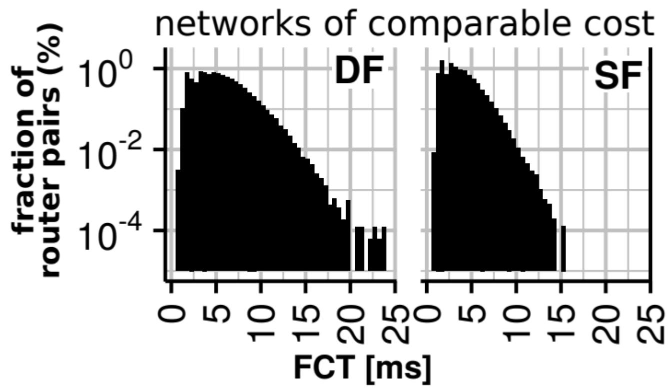


Impact from varying the number and density of layers



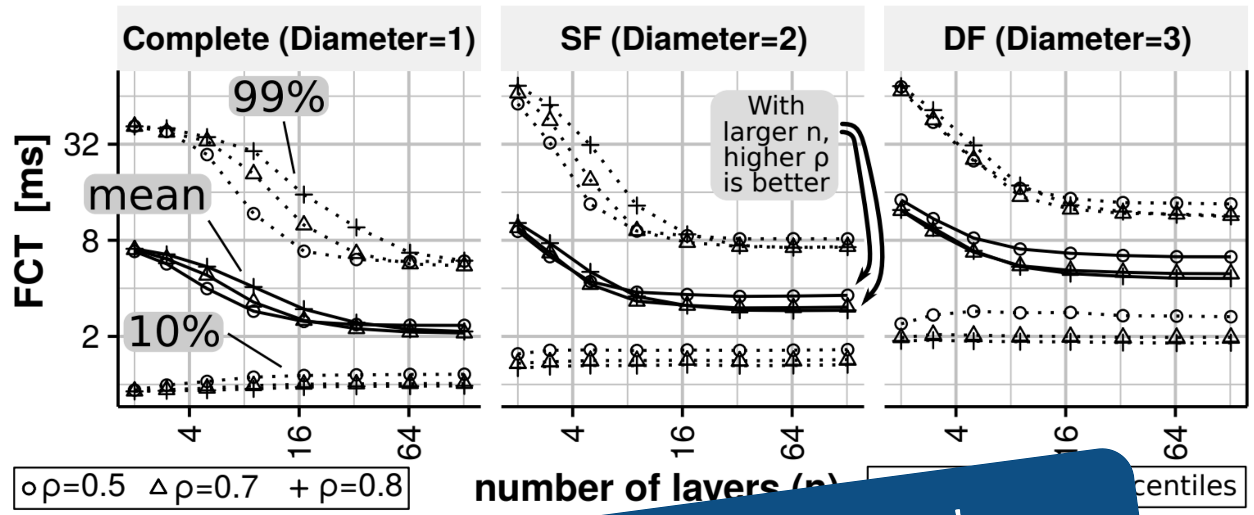
Details of flow behavior



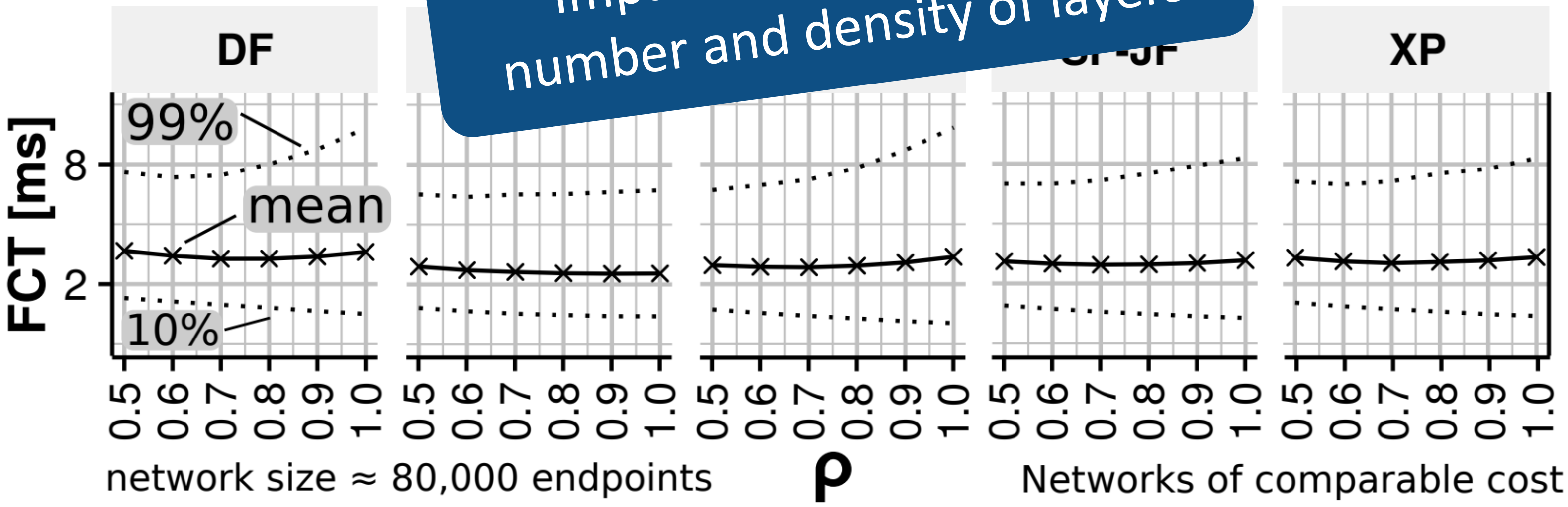


Evaluation

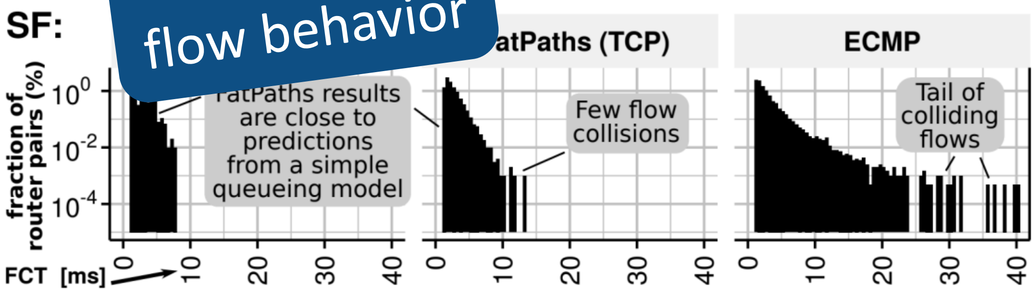
Other analyzes

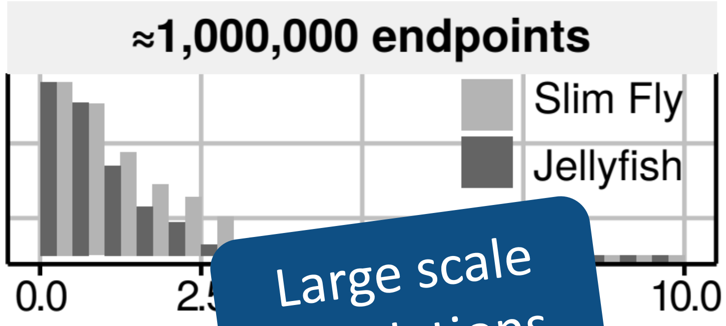


Impact from varying the number and density of layers

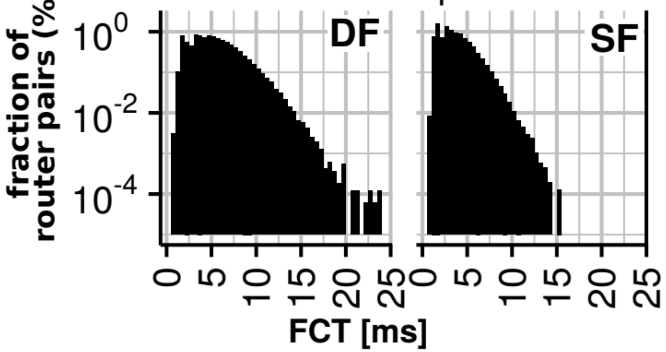


Details of flow behavior

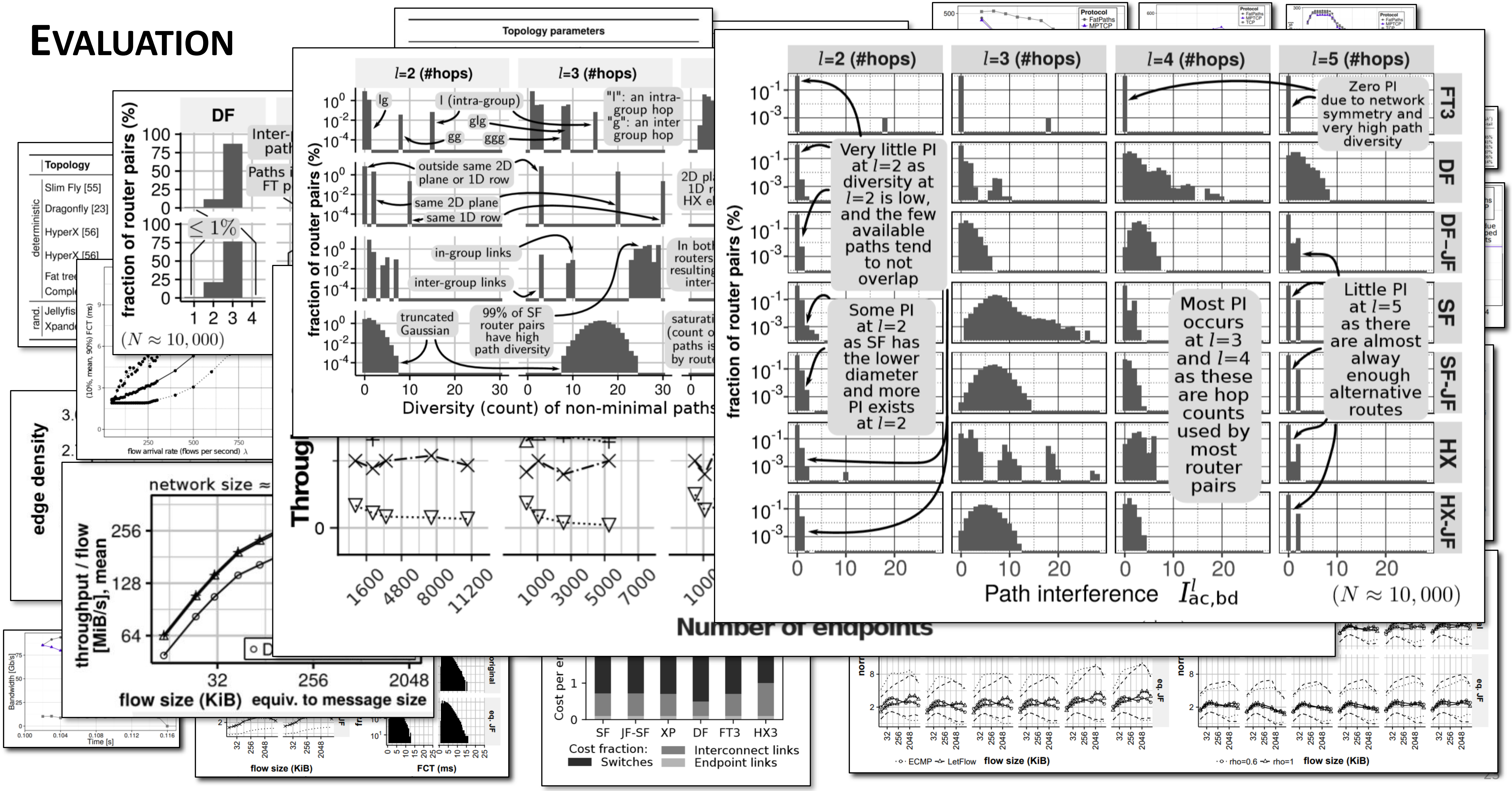




Large scale simulations



EVALUATION


l=2 (#hops)
l=3 (#hops)
l=4 (#hops)
l=5 (#hops)
FT3
DF
DF-JF
SF
SF-JF
HX
HX-JF
fraction of router pairs (%)
Zero PI due to network symmetry and very high path diversity
Very little PI at l=2 as diversity at l=2 is low, and the few available paths tend to not overlap
Some PI at l=2 as SF has the lower diameter and more PI exists at l=2
Most PI occurs at l=3 and l=4 as these are hop counts used by most router pairs
Little PI at l=5 as there are almost alway enough alternative routes
Path interference $I^l_{ac,bd}$
$(N \approx 10,000)$
Diversity (count) of non-minimal paths
truncated Gaussian
99% of SF router pairs have high path diversity
in-group links
inter-group links
outside same 2D plane or 1D row
same 2D plane
same 1D row
"l": an intra-group hop "g": an inter group hop
Topology parameters
Topology
Slim Fly [55]
Dragonfly [23]
HyperX [56]
deterministic
rand.
fraction of router pairs (%)
DF
$\leq 1\%$
$(N \approx 10,000)$
edge density
network size ≈
throughput / flow [MiB/s], mean
flow size (KiB) equiv. to message size
Number of endpoints
Cost fraction:
Switches
Interconnect links
Endpoint links
flow size (KiB)
FCT (ms)
flow arrival rate (flows per second) λ
ECMP
LetFlow
rho=0.6
rho=1

Evaluation

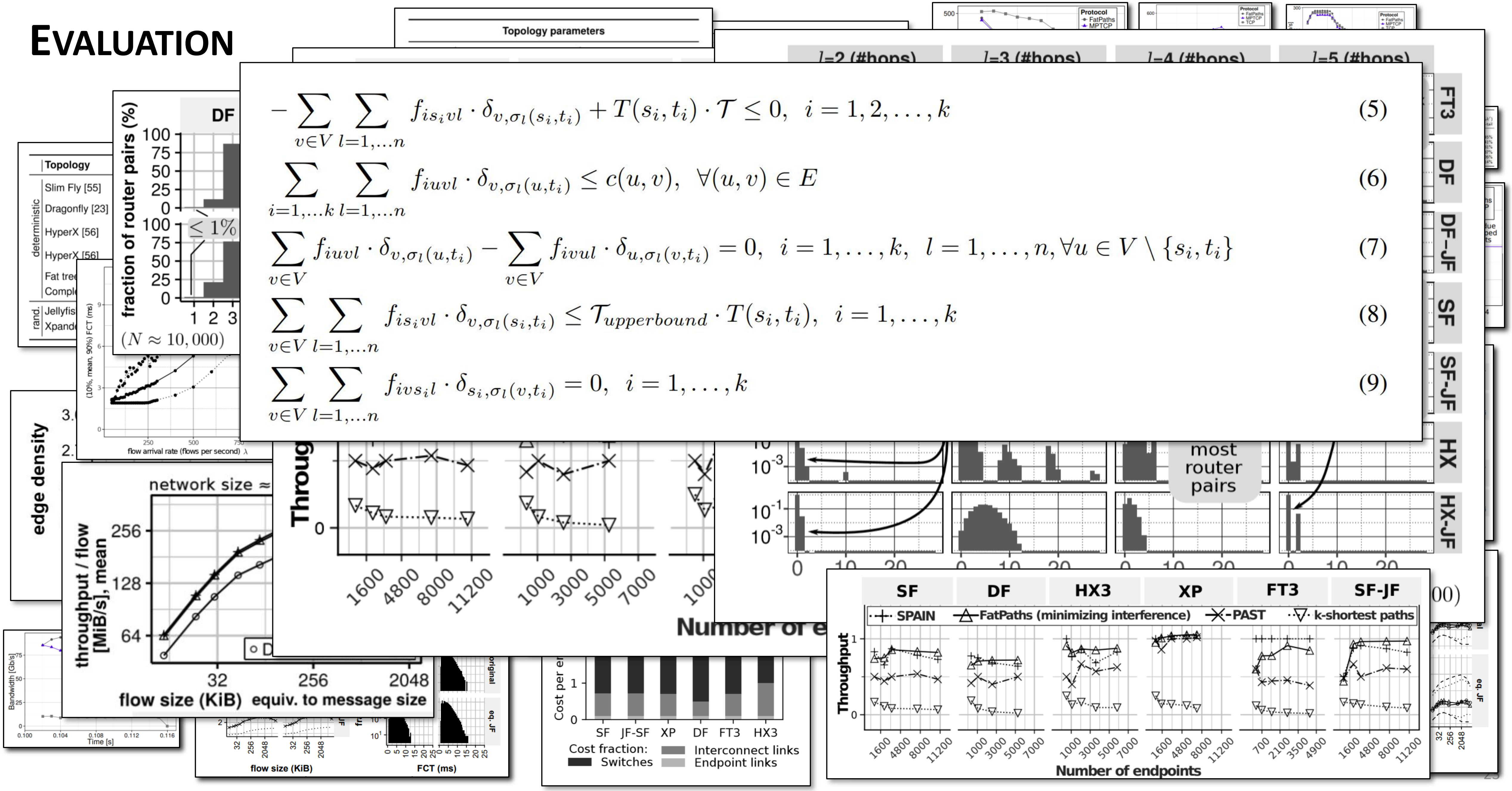
$$-\sum_{v \in V} \sum_{l=1,\ldots,n} f_{is_ivl} \cdot \delta_{v,\sigma_l(s_i,t_i)} + T(s_i,t_i) \cdot \mathcal{T} \leq 0, \quad i = 1, 2, \ldots, k \tag{5}$$

$$\sum_{i=1,\ldots k} \sum_{l=1,\ldots,n} f_{iuvl} \cdot \delta_{v,\sigma_l(u,t_i)} \leq c(u,v), \quad \forall (u,v) \in E \tag{6}$$

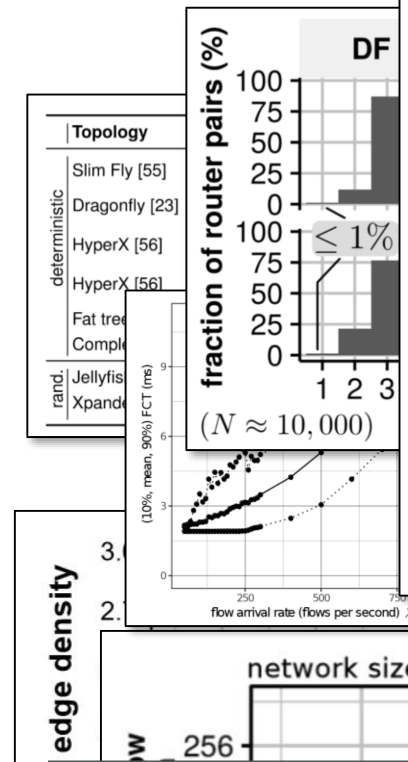
$$\sum_{v \in V} f_{iuvl} \cdot \delta_{v,\sigma_l(u,t_i)} - \sum_{v \in V} f_{ivul} \cdot \delta_{u,\sigma_l(v,t_i)} = 0, \quad i = 1, \ldots, k, \quad l = 1, \ldots, n, \forall u \in V \setminus \{s_i, t_i\} \tag{7}$$

$$\sum_{v \in V} \sum_{l=1,\ldots,n} f_{is_ivl} \cdot \delta_{v,\sigma_l(s_i,t_i)} \leq \mathcal{T}_{upperbound} \cdot T(s_i,t_i), \quad i = 1, \ldots, k \tag{8}$$

$$\sum_{v \in V} \sum_{l=1,\ldots,n} f_{ivs_il} \cdot \delta_{s_i,\sigma_l(v,t_i)} = 0, \quad i = 1, \ldots, k \tag{9}$$



Evaluation



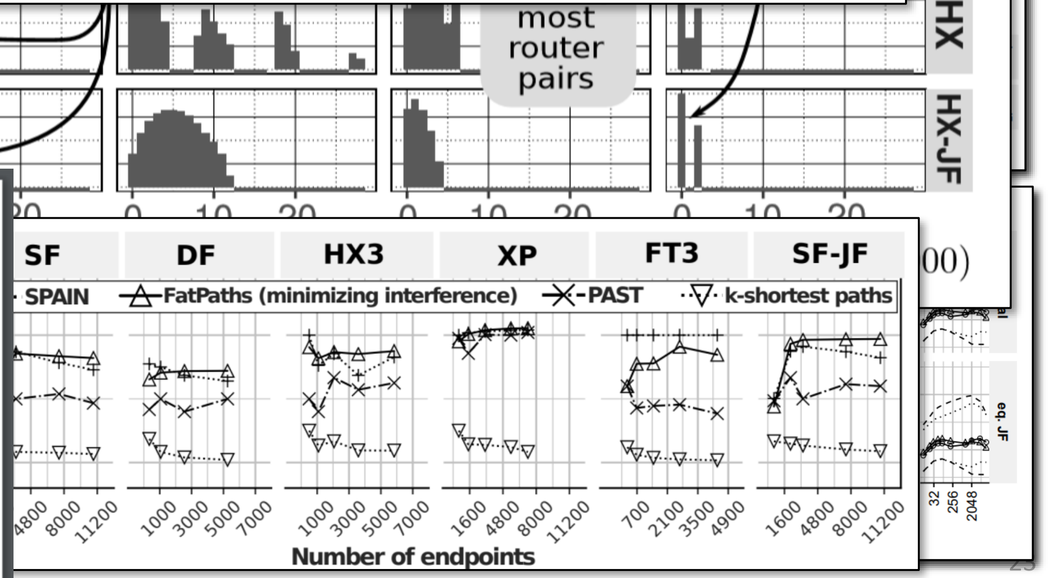
$$-\sum_{v\in V}\sum_{l=1,\dots,n} f_{is_ivl}\cdot\delta_{v,\sigma_l(s_i,t_i)} + T(s_i,t_i)\cdot\mathcal{T}\le 0,\ \ i=1,2,\dots,k \tag{5}$$

$$\sum_{i=1,\dots k}\sum_{l=1,\dots,n} f_{iuvl}\cdot\delta_{v,\sigma_l(u,t_i)}\le c(u,v),\ \ \forall(u,v)\in E \tag{6}$$

$$\sum_{v\in V} f_{iuvl}\cdot\delta_{v,\sigma_l(u,t_i)} - \sum_{v\in V} f_{ivul}\cdot\delta_{u,\sigma_l(v,t_i)} = 0,\ \ i=1,\dots,k,\ \ l=1,\dots,n,\forall u\in V\setminus\{s_i,t_i\} \tag{7}$$

$$\sum_{v\in V}\sum_{l=1,\dots,n} f_{is_ivl}\cdot\delta_{v,\sigma_l(s_i,t_i)}\le \mathcal{T}_{upperbound}\cdot T(s_i,t_i),\ \ i=1,\dots,k \tag{8}$$

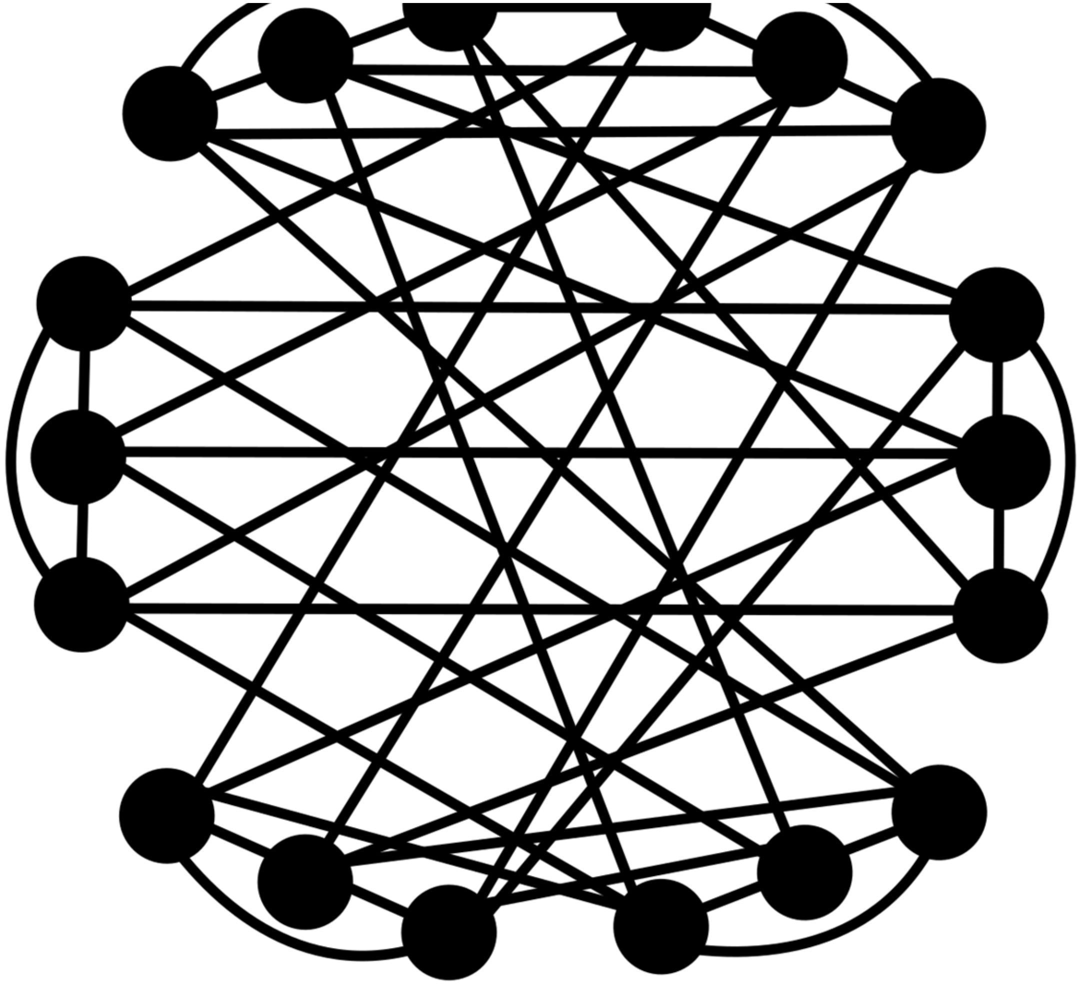
$$\sum_{v\in V}\sum_{l=1,\dots,n} f_{ivs_il}\cdot\delta_{s_i,\sigma_l(v,t_i)} = 0,\ \ i=1,\dots,k \tag{9}$$

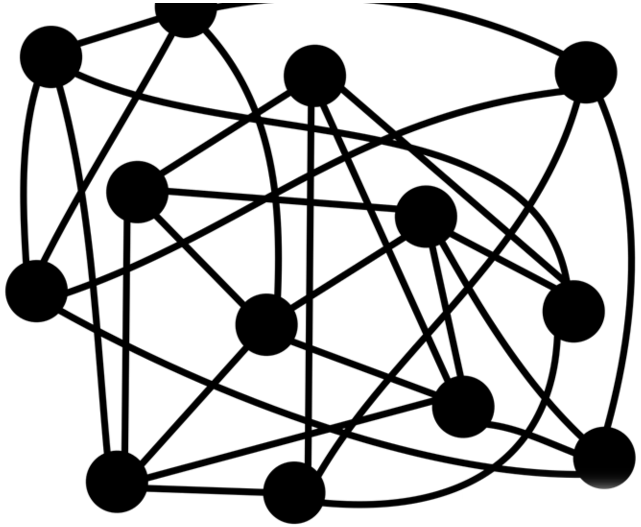


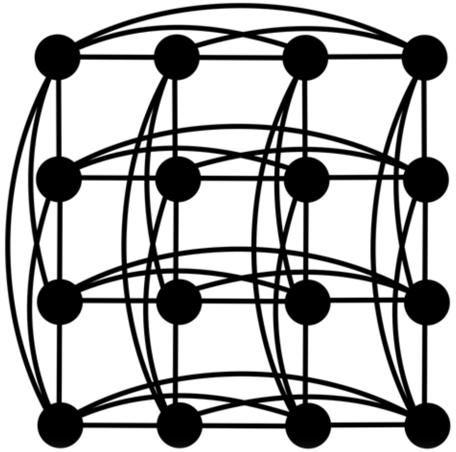
1

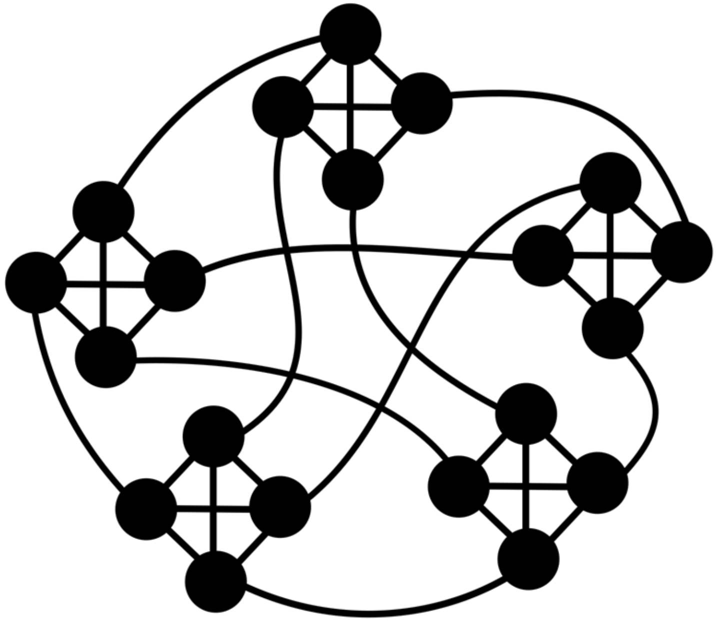
High-Performance Routing with Multipathing and Path Diversity in Supercomputers and Data Centers

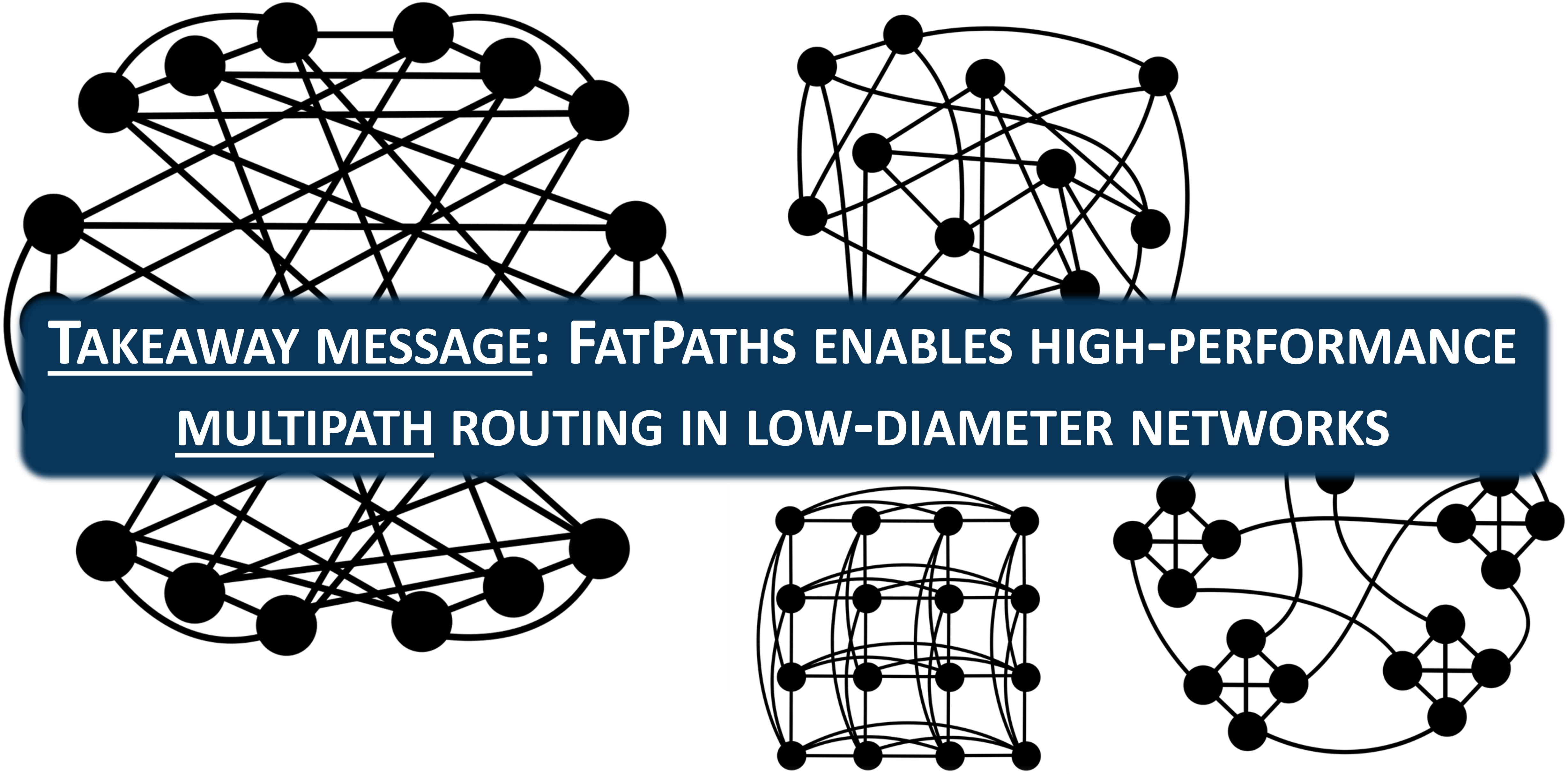
Maciej Besta[1], Jens Domke[2], Marcel Schneider[1], Marek Konieczny[3], Salvatore Di Girolamo[1], Timo Schneider[1], Ankit Singla[1], Torsten Hoefler[1]
[1]Department of Computer Science, ETH Zurich; [2]RIKEN Center for Computational Science (R-CCS)
[3]Department of Computer Science, Electronics and Telecommunications; AGH-UST





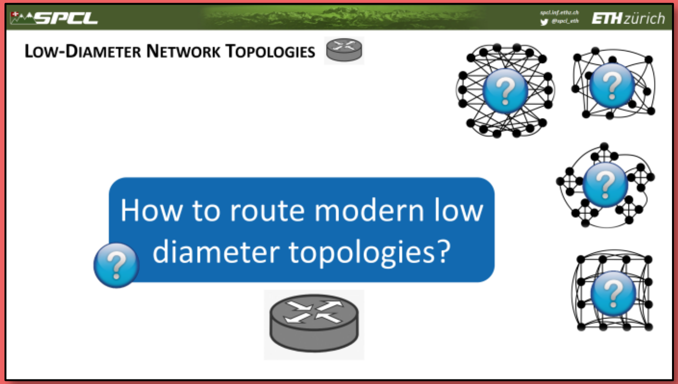


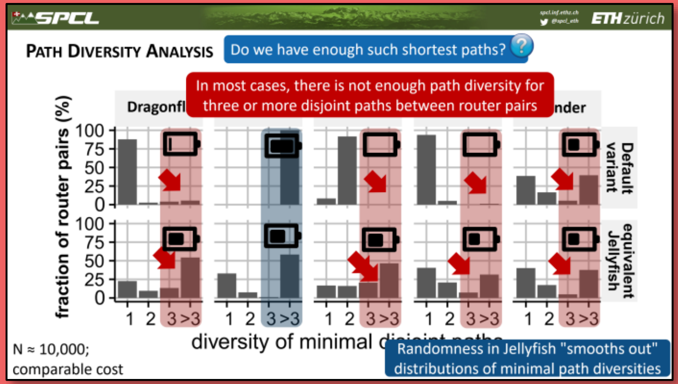


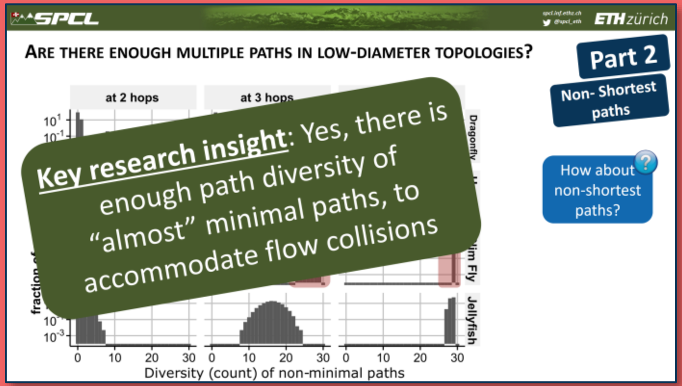


TAKEAWAY MESSAGE: FATPATHS ENABLES HIGH-PERFORMANCE MULTIPATH ROUTING IN LOW-DIAMETER NETWORKS

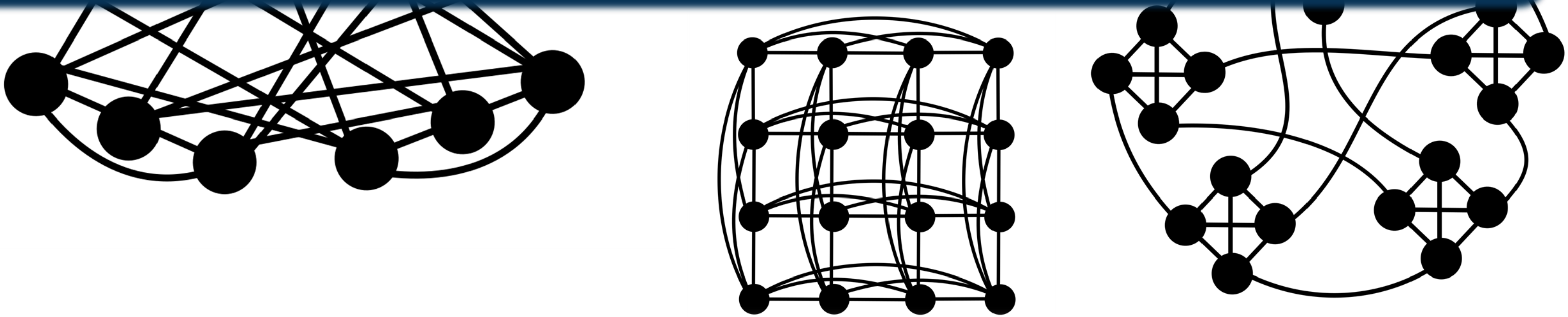
Potential for multipath routing in modern low-diameter networks



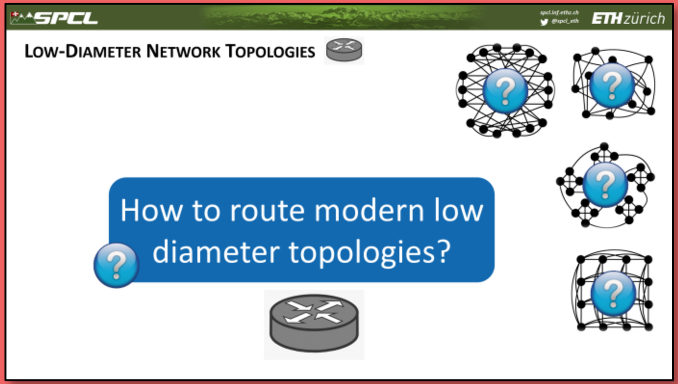


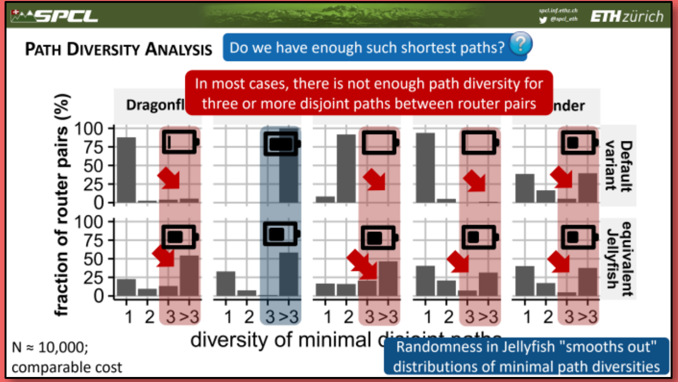


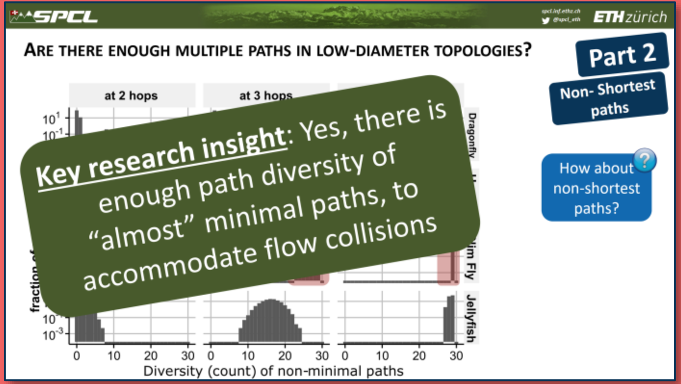
Takeaway message: FatPaths enables high-performance multipath routing in low-diameter networks



Potential for multipath routing in modern low-diameter networks

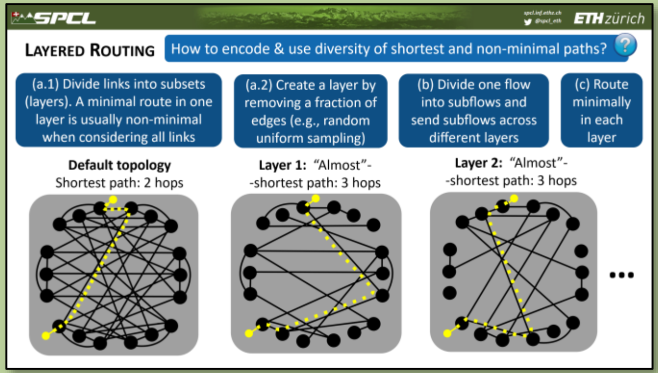


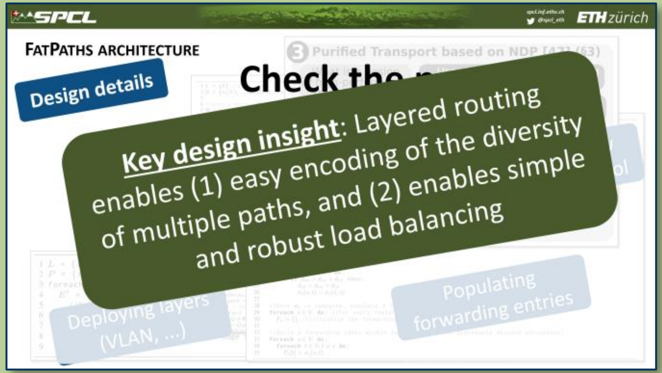




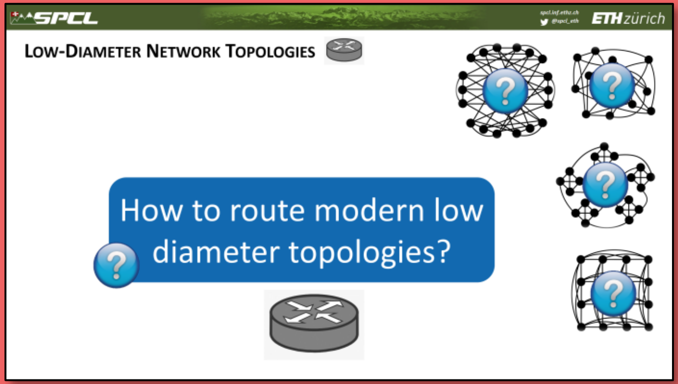
Takeaway message: FatPaths enables high-performance multipath routing in low-diameter networks

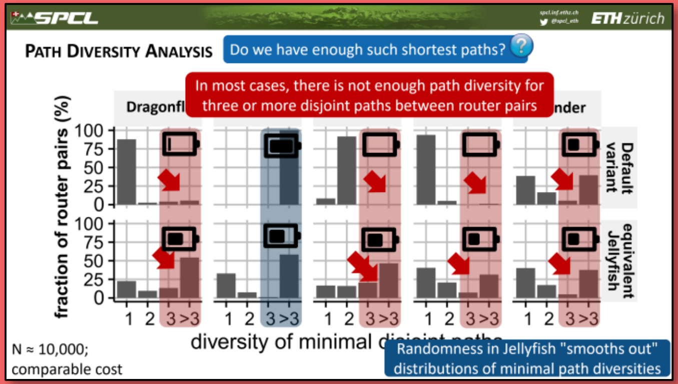
High-performance routing protocol & architecture

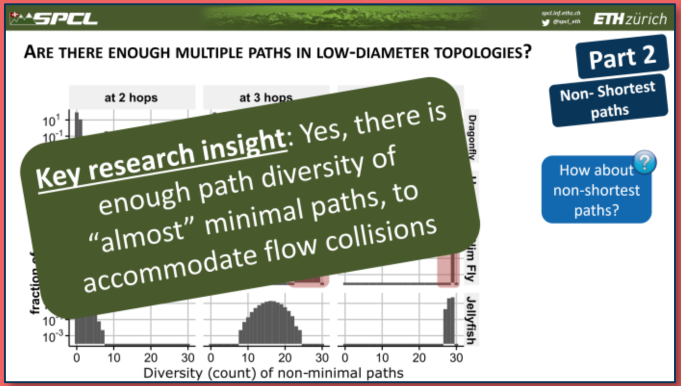




Potential for multipath routing in modern low-diameter networks

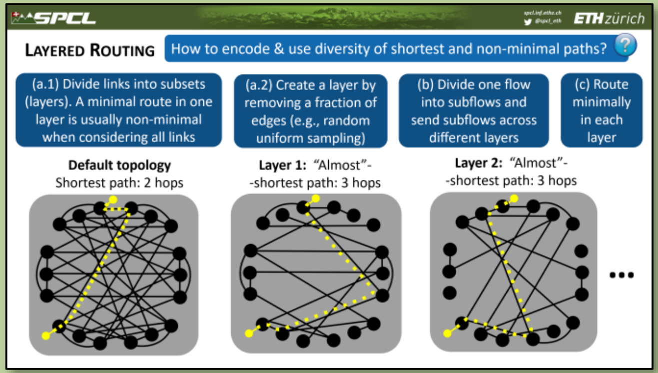


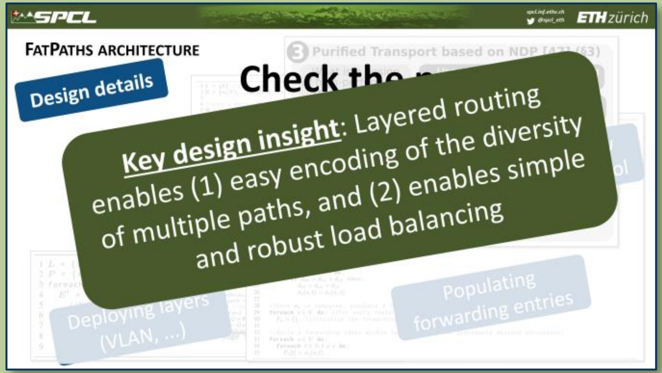




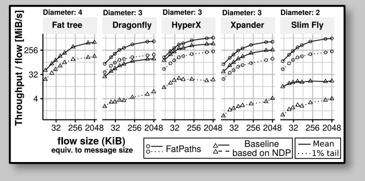
Takeaway message: FatPaths enables high-performance multipath routing in low-diameter networks

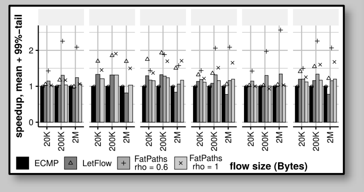
High-performance routing protocol & architecture

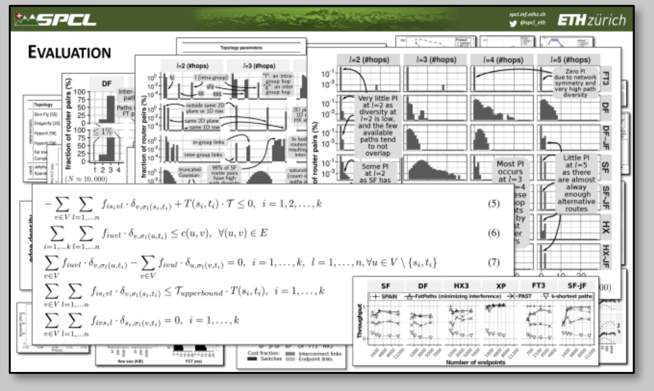




Rich evaluation & theoretical analysis

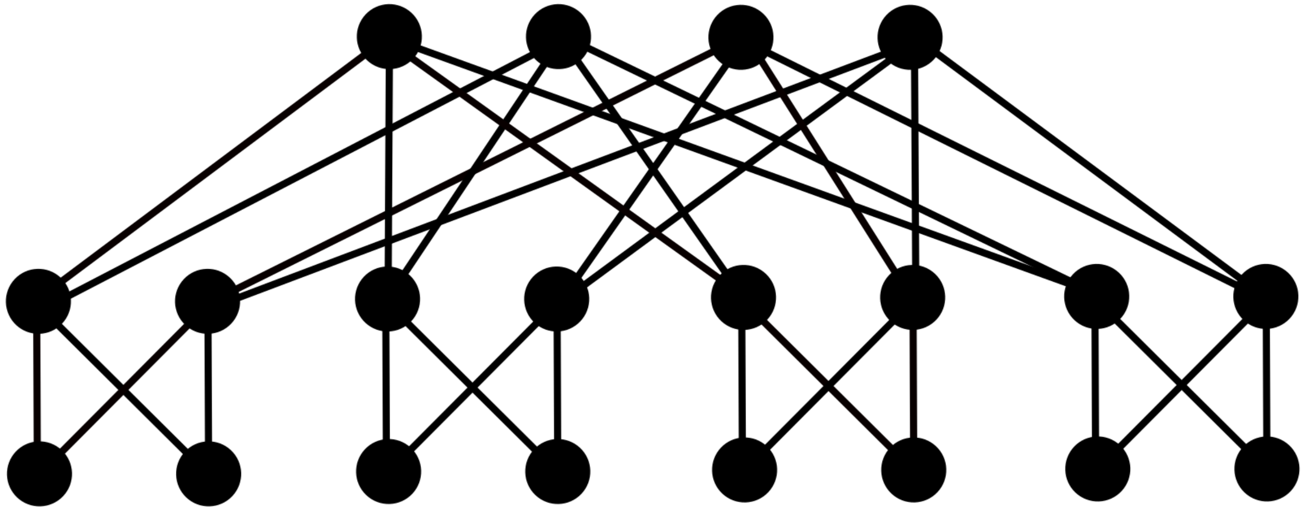






Backup Slides and Slides' Variants

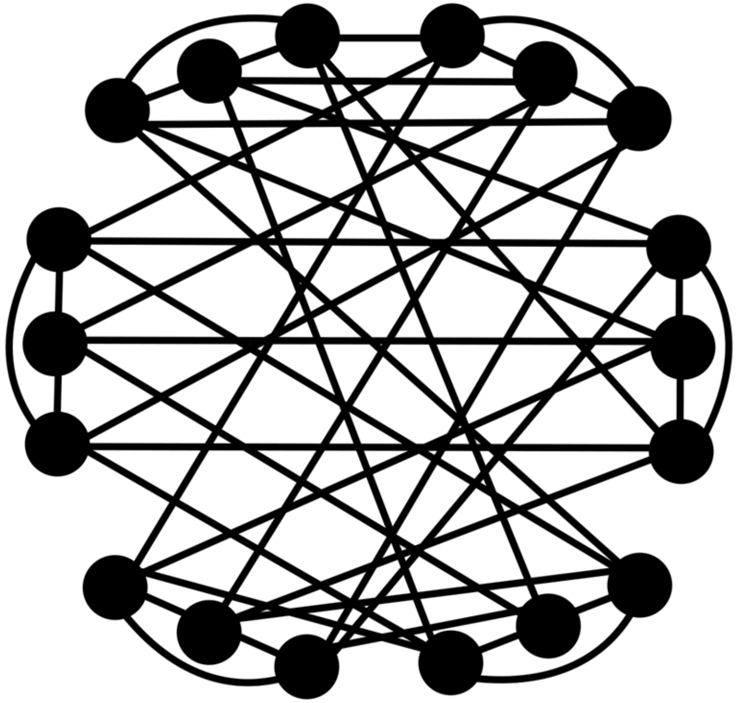
Low-Diameter Network Topologies vs Fat Trees



3-level Fat tree

Diameter = 4

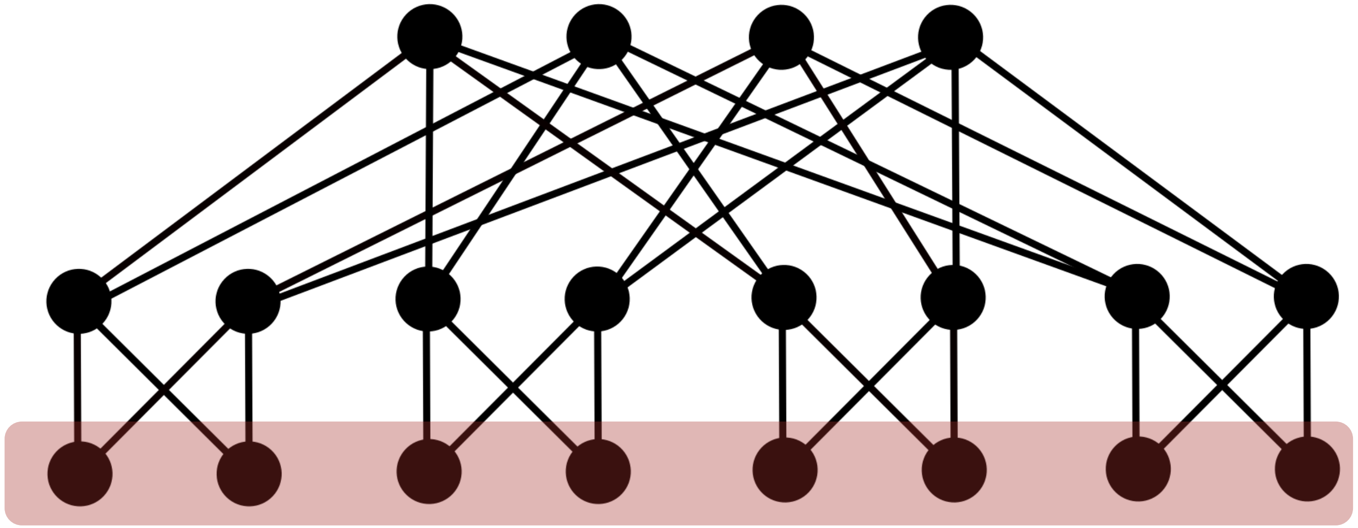
Example Slim Fly (based on the Hoffman-Singleton Graph [1])



Diameter = 2

[1] Hoffman, Alan J.; Singleton, Robert R. (1960), *Moore graphs with diameter 2 and 3*, IBM Journal of Research and Development

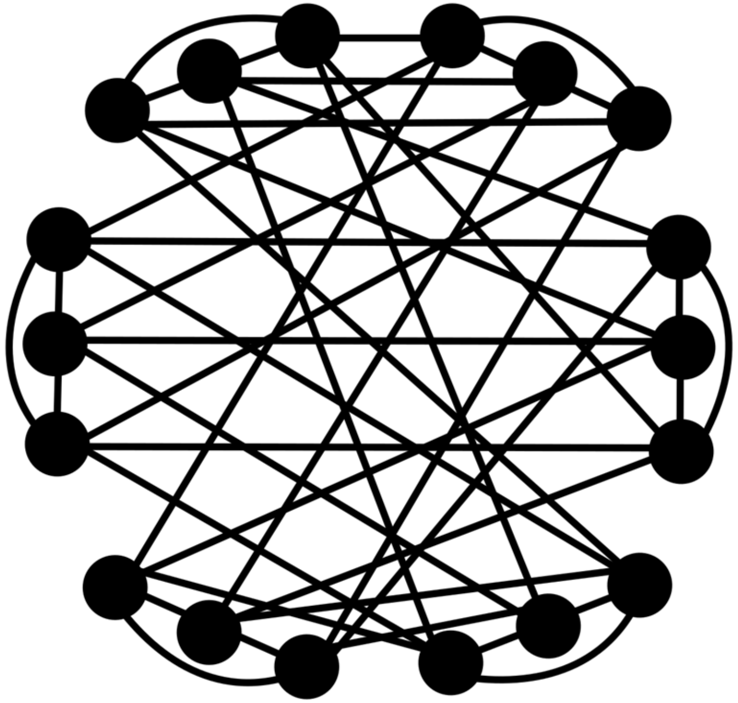
Low-Diameter Network Topologies vs Fat Trees



3-level Fat tree

Diameter = 4
Only edge (leaf) routers attach to endpoints

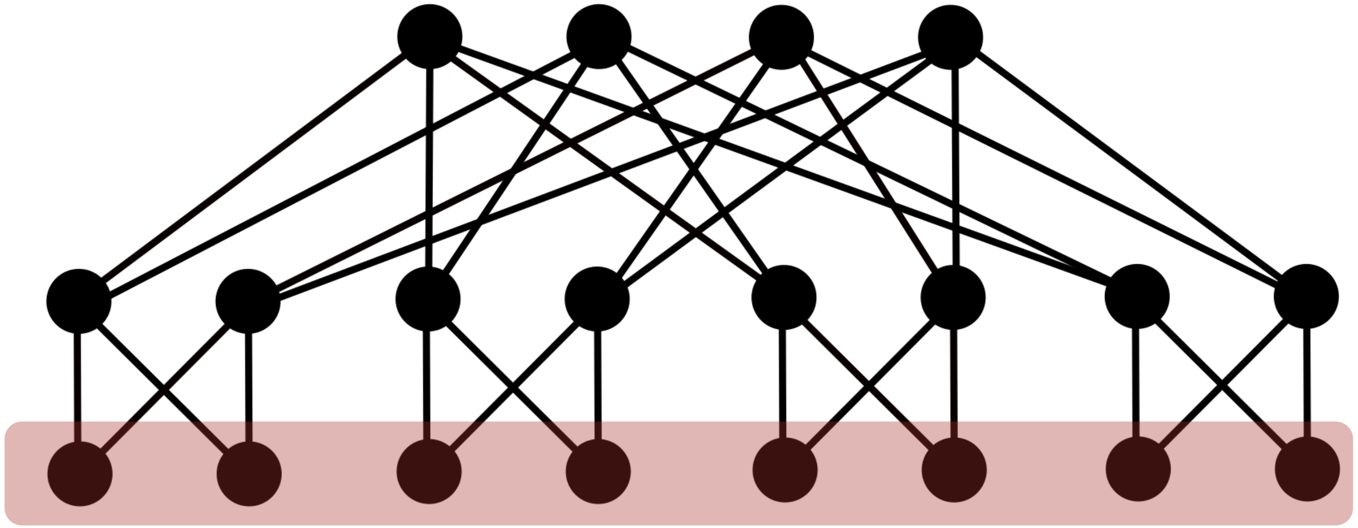
Example Slim Fly (based on the Hoffman-Singleton Graph [1])



Diameter = 2

[1] Hoffman, Alan J.; Singleton, Robert R. (1960), *Moore graphs with diameter 2 and 3*, IBM Journal of Research and Development

Low-Diameter Network Topologies vs Fat Trees



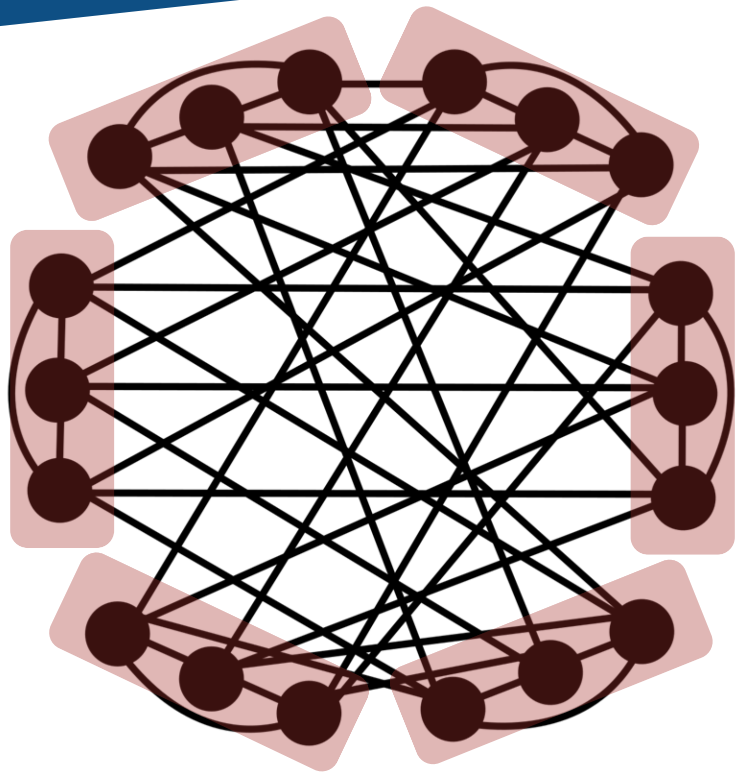
3-level Fat tree

Diameter = 4

Only edge (leaf) routers attach to endpoints

All routers attach to endpoints

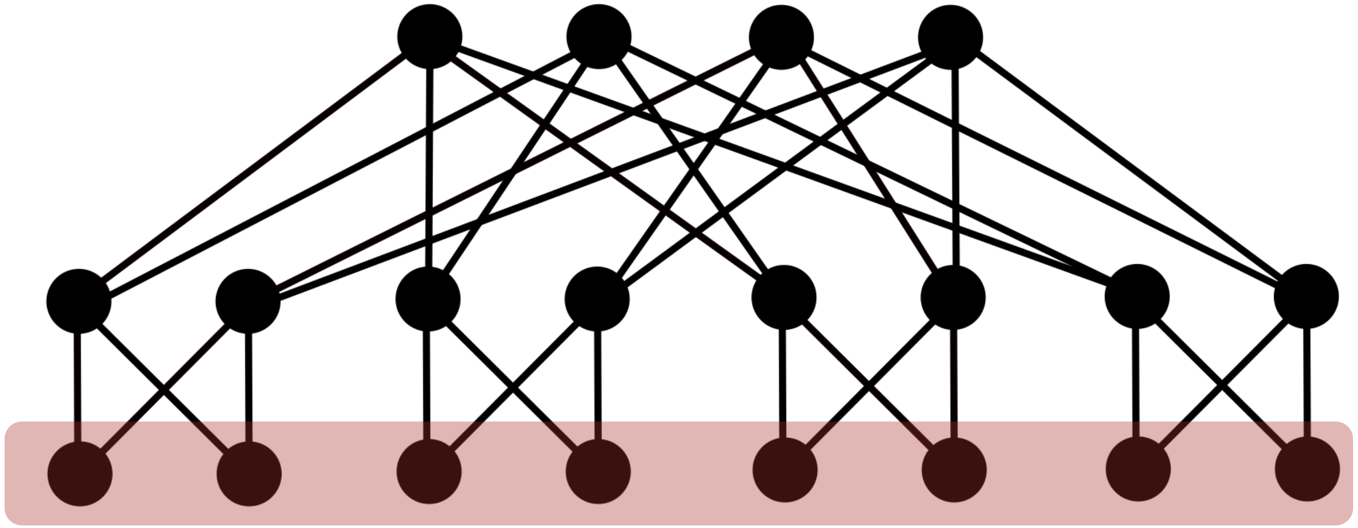
Example Slim Fly (based on the Hoffman-Singleton Graph [1])



Diameter = 2

[1] Hoffman, Alan J.; Singleton, Robert R. (1960), *Moore graphs with diameter 2 and 3*, IBM Journal of Research and Development

Low-Diameter Network Topologies vs Fat Trees



3-level Fat tree

Diameter = 4

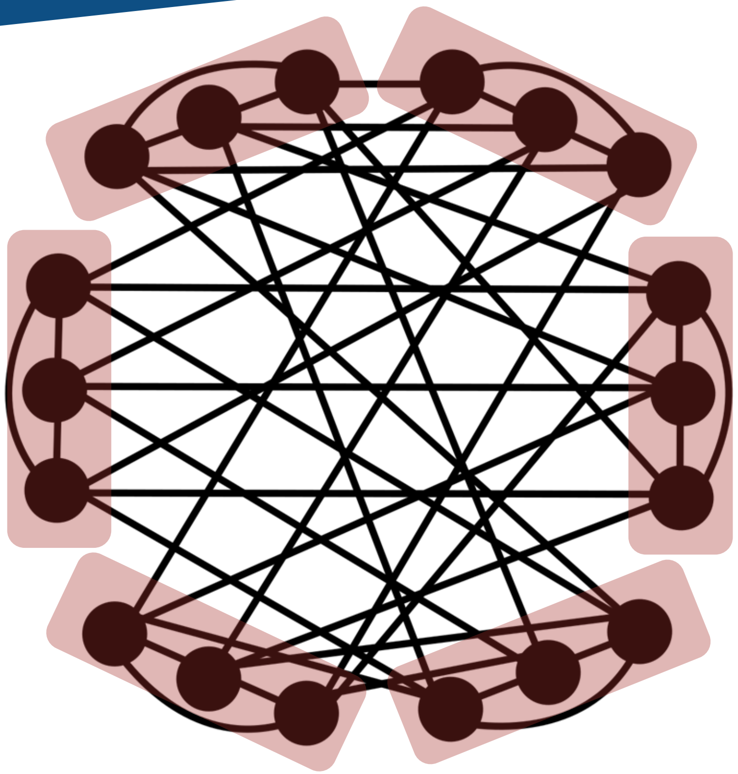
Only edge (leaf) routers attach to endpoints

Lower concentration

All routers attach to endpoints

Higher concentration

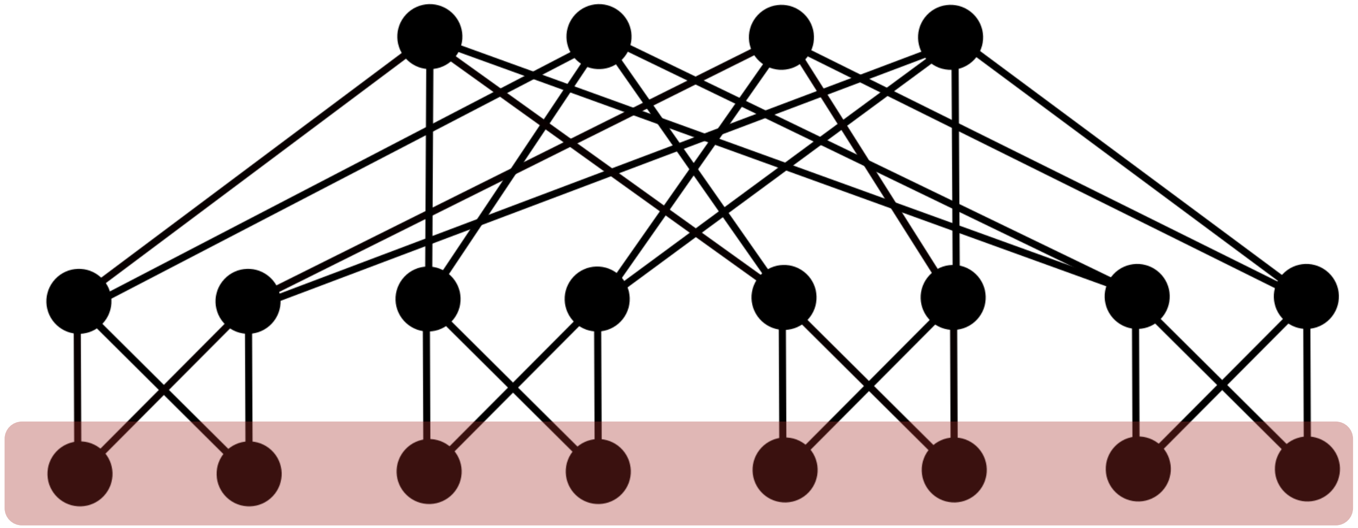
Example Slim Fly (based on the Hoffman-Singleton Graph [1])



Diameter = 2

[1] Hoffman, Alan J.; Singleton, Robert R. (1960), *Moore graphs with diameter 2 and 3*, IBM Journal of Research and Development

Low-Diameter Network Topologies vs Fat Trees

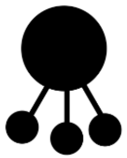


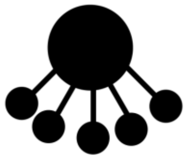
3-level Fat tree

Diameter = 4

Only edge (leaf) routers attach to endpoints

Lower concentration





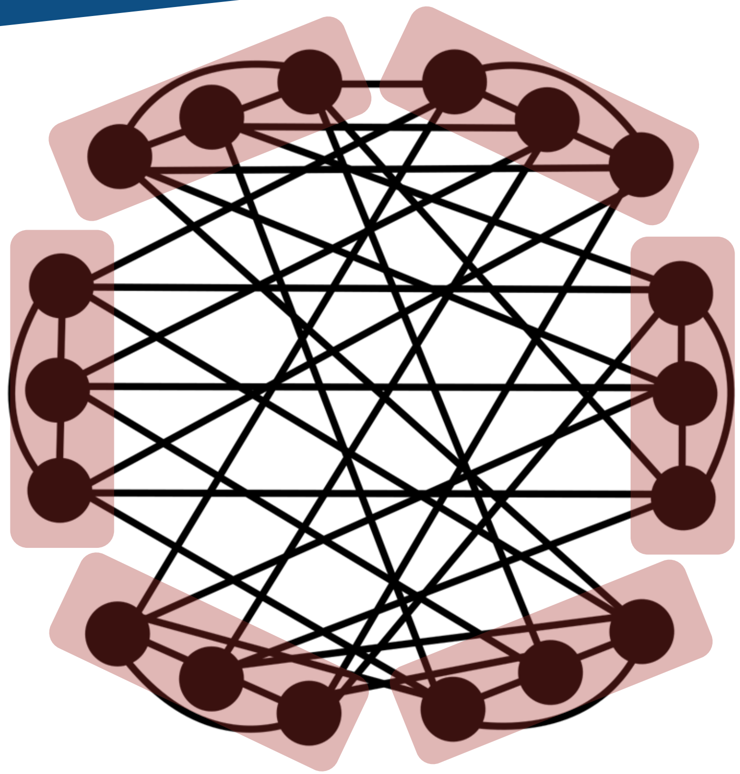
Higher concentration

All routers attach to endpoints

> ≈50% fewer routers

> ≈30% fewer cables

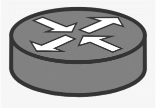
Example Slim Fly (based on the Hoffman-Singleton Graph [1])



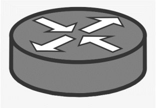
Diameter = 2

[1] Hoffman, Alan J.; Singleton, Robert R. (1960), *Moore graphs with diameter 2 and 3*, IBM Journal of Research and Development

MULTIPATH ROUTING: MOTIVATION



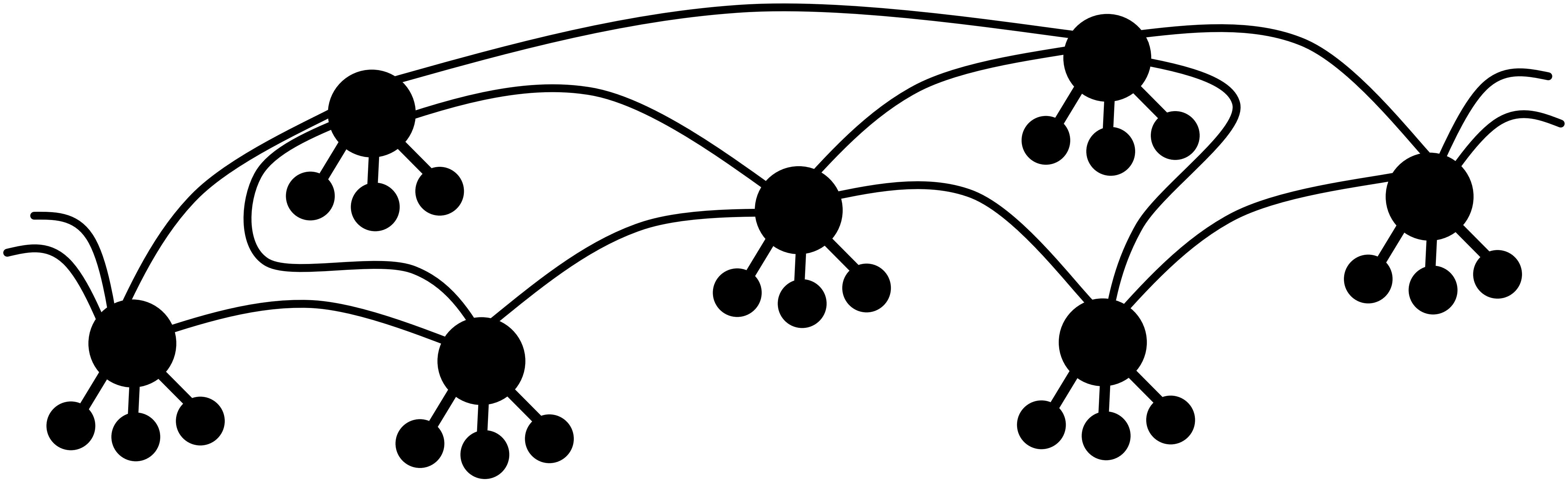
Multipath Routing: Motivation



What are the problems that we want to tackle with multipathing?

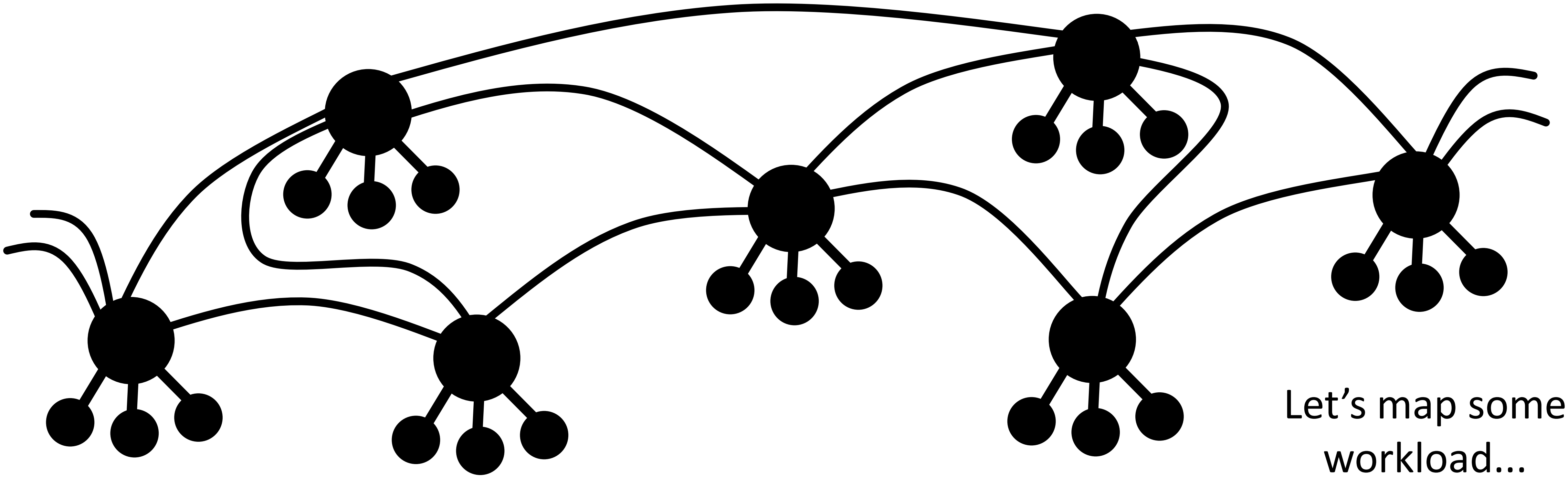
MULTIPATH ROUTING: MOTIVATION

What are the problems that we want to tackle with multipathing?



MULTIPATH ROUTING: MOTIVATION

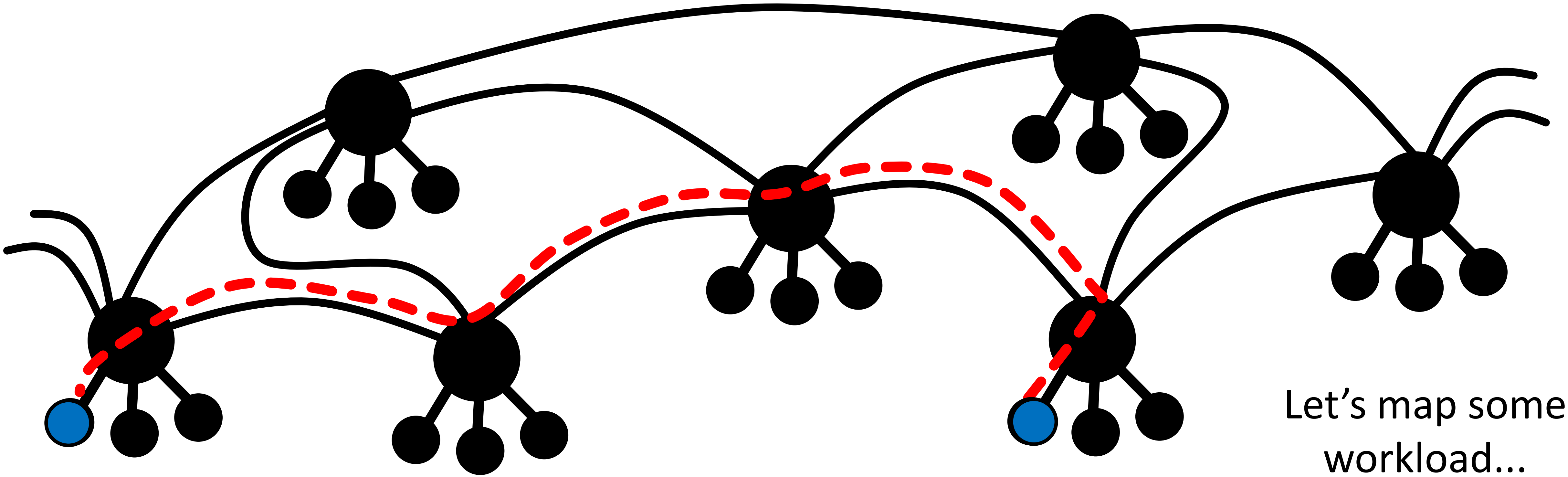
What are the problems that we want to tackle with multipathing?



Let’s map some workload...

Multipath Routing: Motivation

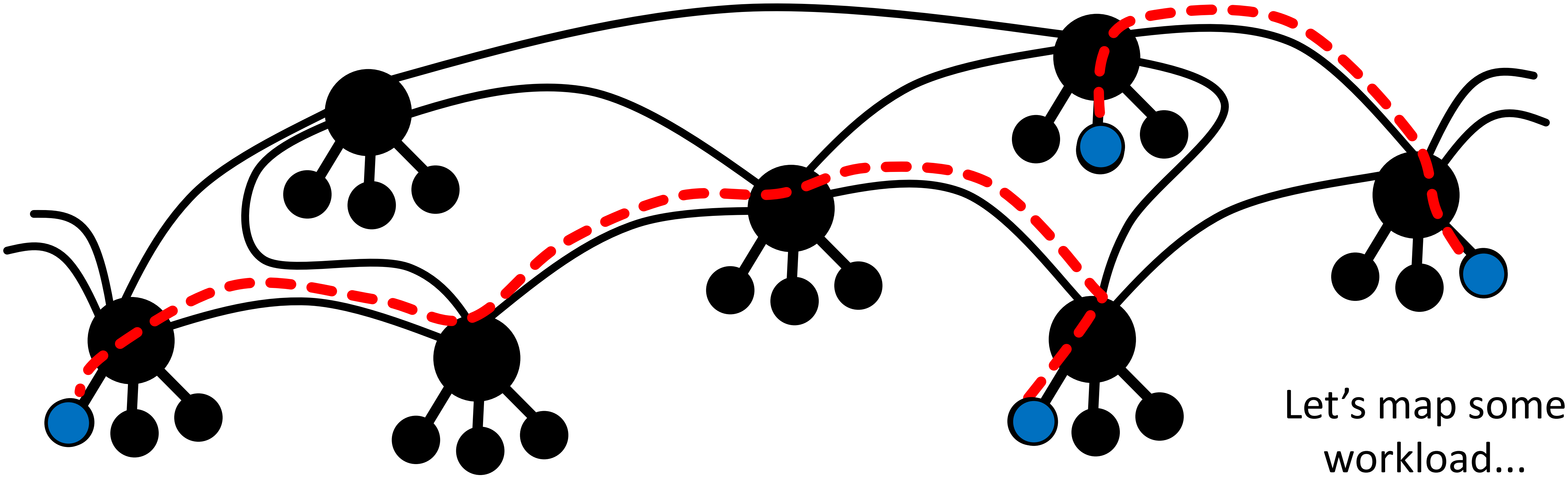
What are the problems that we want to tackle with multipathing?



Let's map some workload...

Multipath Routing: Motivation

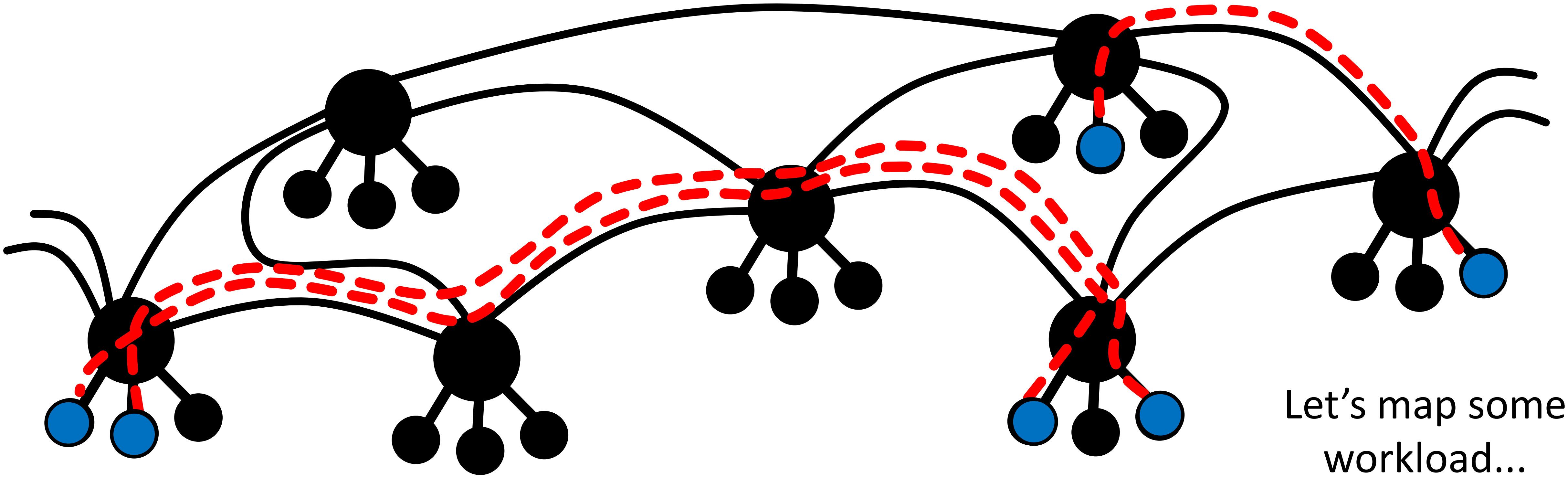
What are the problems that we want to tackle with multipathing?



Let's map some workload...

Multipath Routing: Motivation

What are the problems that we want to tackle with multipathing?

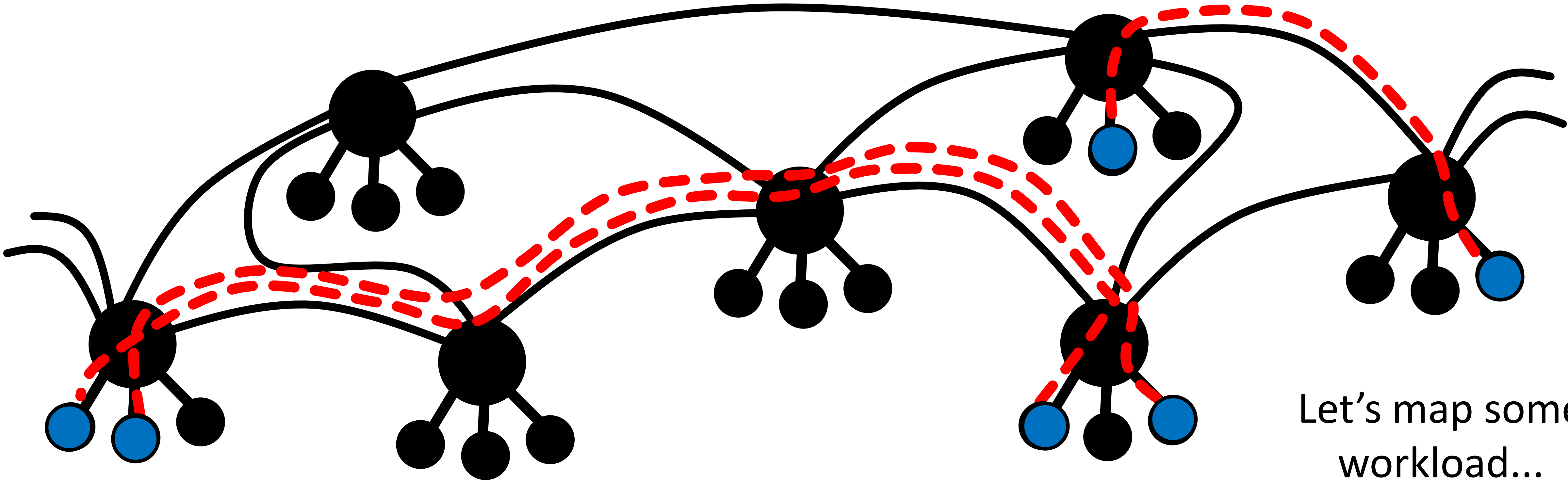


Let's map some workload...

MULTIPATH ROUTING: MOTIVATION

What are the problems that we want to tackle with multipathing?

Flows collide!

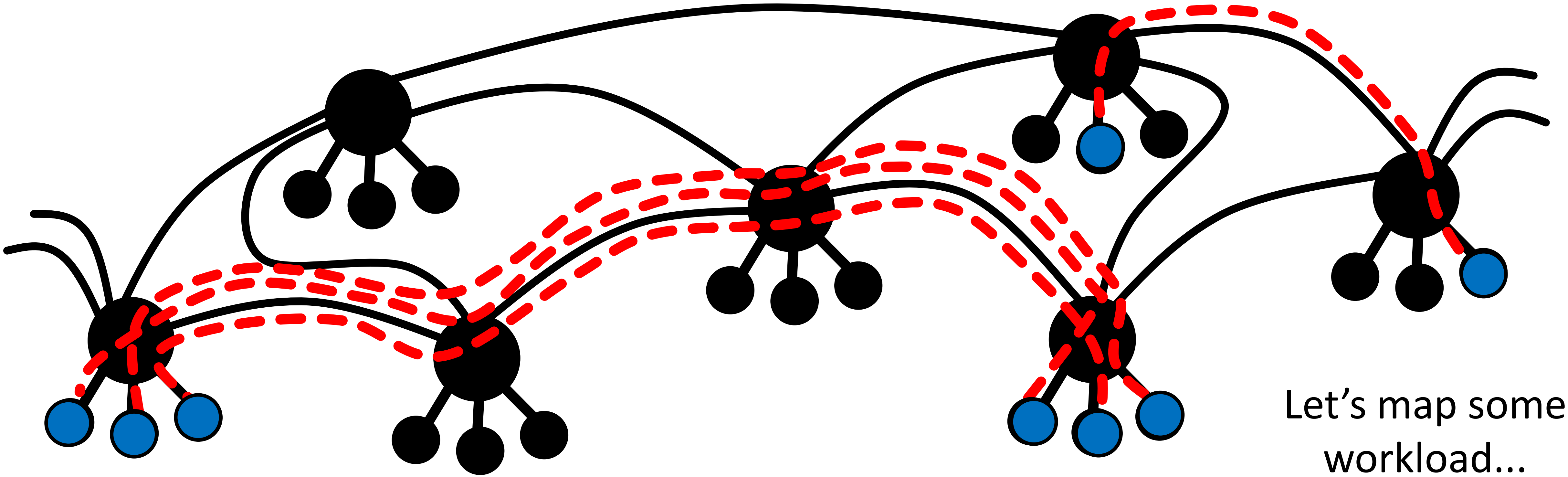


Let's map some workload...

Multipath Routing: Motivation

What are the problems that we want to tackle with multipathing?

Flows collide!

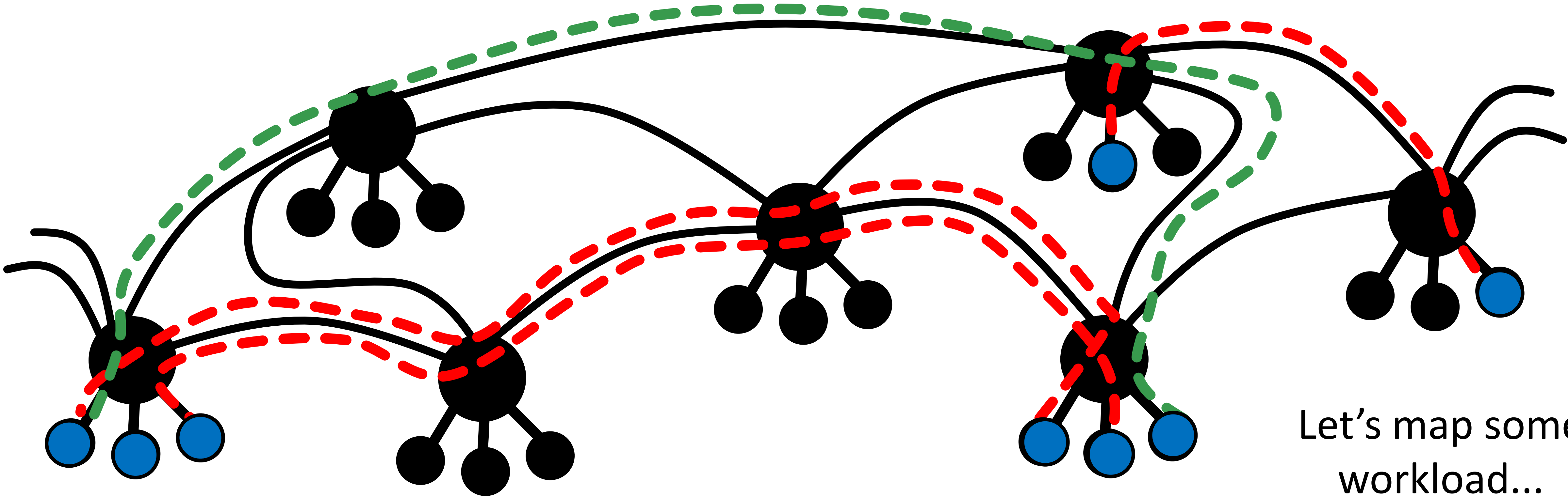


Let's map some workload...

MULTIPATH ROUTING: MOTIVATION

What are the problems that we want to tackle with multipathing?

Flows collide!



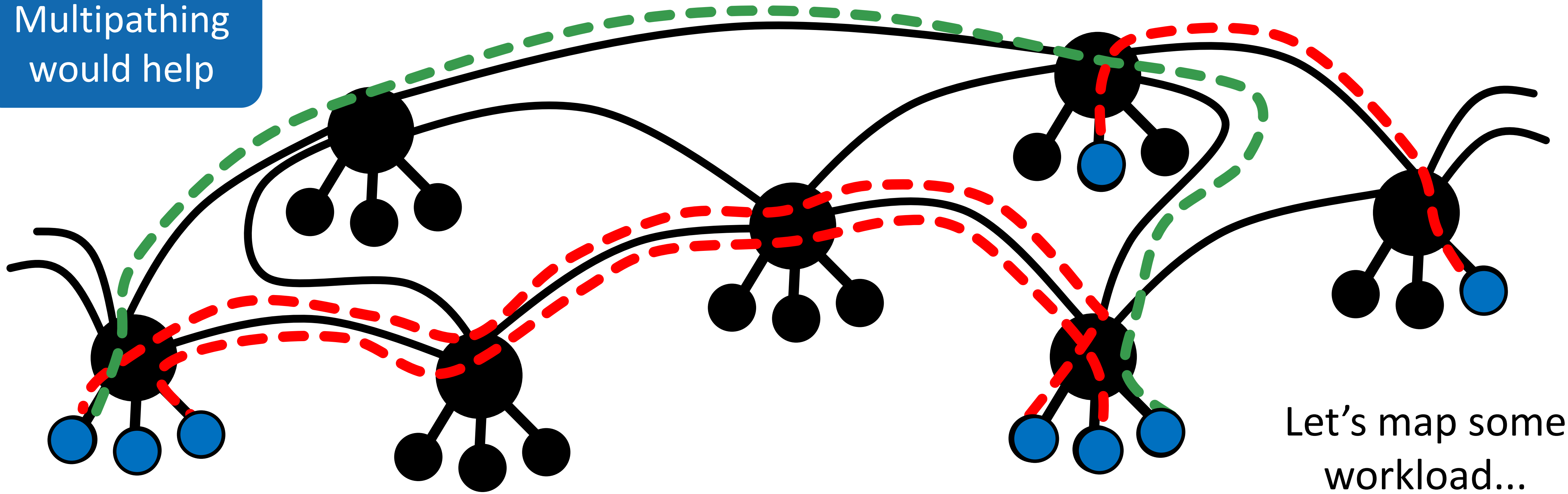
Let's map some workload...

MULTIPATH ROUTING: MOTIVATION

What are the problems that we want to tackle with multipathing?

Multipathing would help

Flows collide!



Let's map some workload...

MULTIPATH ROUTING: MOTIVATION

What are the problems that we want to tackle with multipathing?

Flows collide!

Multipathing would help

Let's map some workload...

Multipath Routing: Motivation

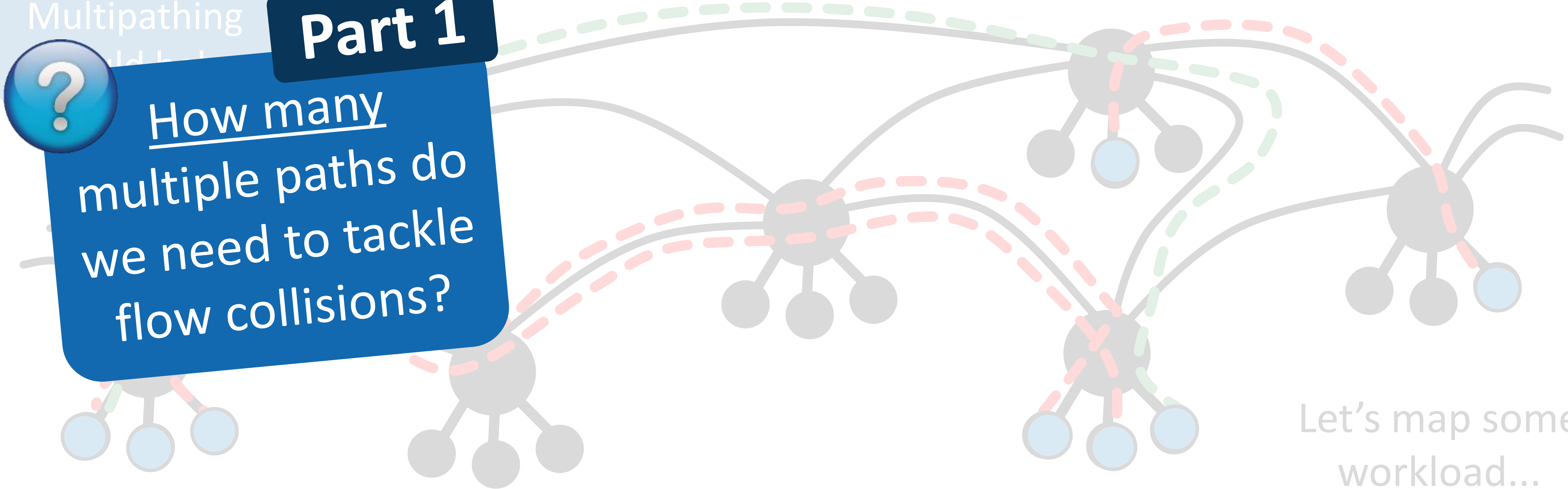
What are the problems that we want to tackle with multipathing?

Multipathing

Part 1

How many multiple paths do we need to tackle flow collisions?

Flows collide!



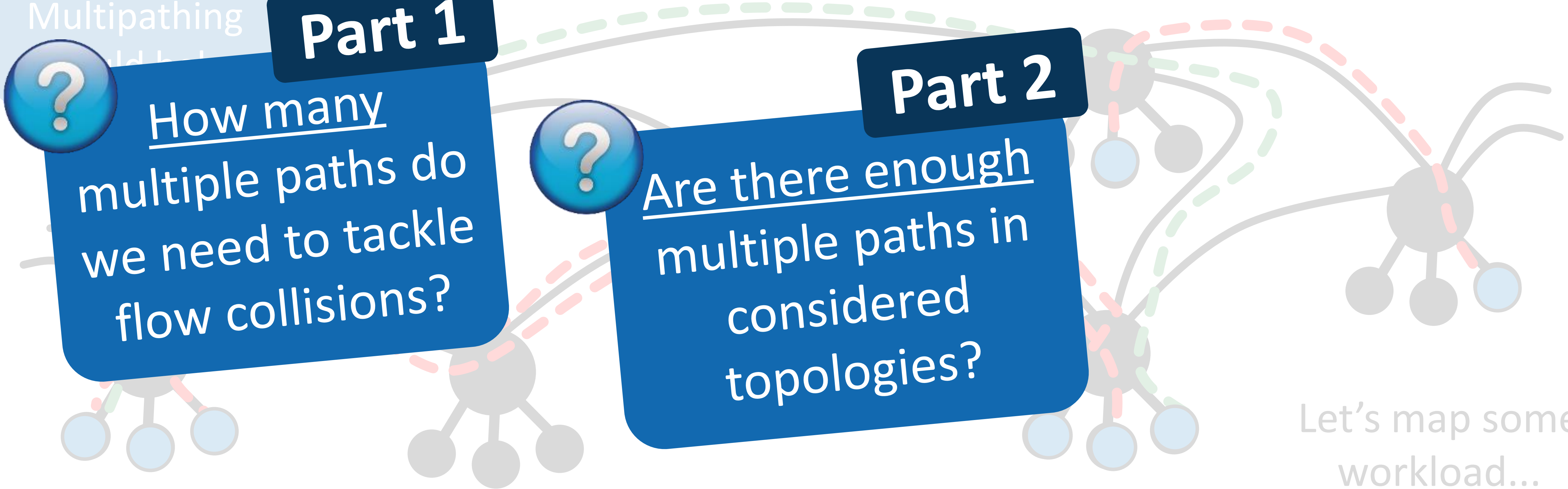
Let's map some workload...

Multipath Routing: Motivation

What are the problems that we want to tackle with multipathing?

Flows collide!

Multipathing



Part 1

How many multiple paths do we need to tackle flow collisions?

Part 2

Are there enough multiple paths in considered topologies?

Let's map some workload...

MULTIPATH ROUTING: MOTIVATION

What are the problems that we want to tackle with multipathing?

Flows collide!

Multipathing

Part 1

How many multiple paths do we need to tackle flow collisions?

Part 2

Are there enough multiple paths in considered topologies?

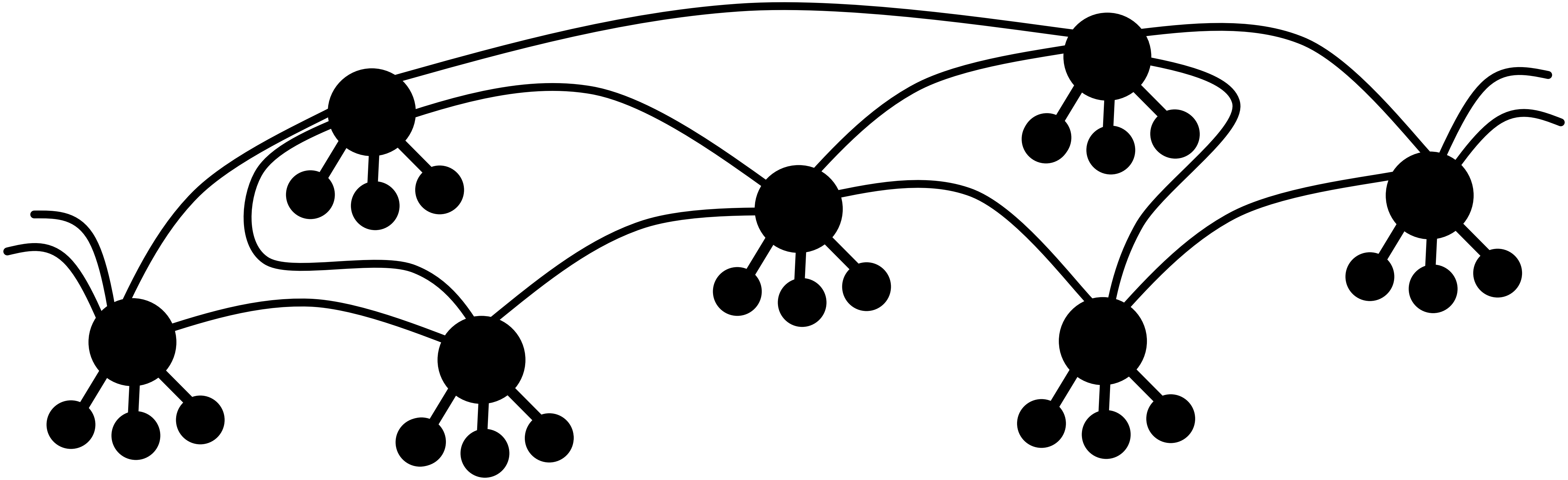
Part 3

How to use such multiple paths in considered topologies?

workload...

How many multiple paths do we need to tackle flow collisions?

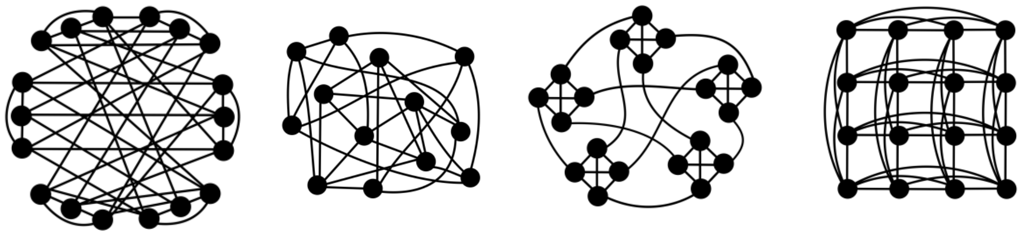
Part 1

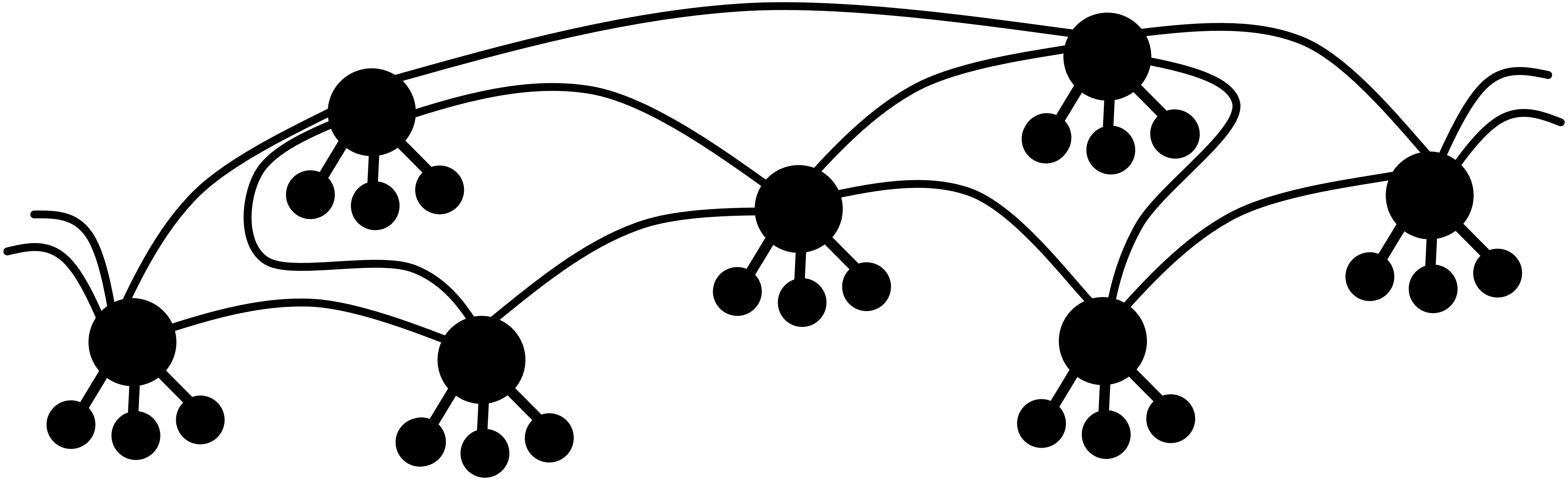


How many multiple paths do we need to tackle flow collisions?

Part 1

...For different topologies

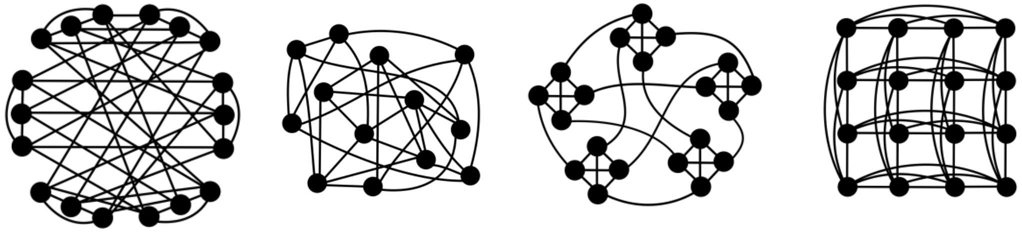




How many multiple paths do we need to tackle flow collisions?

Part 1

...For different topologies



...For different workloads

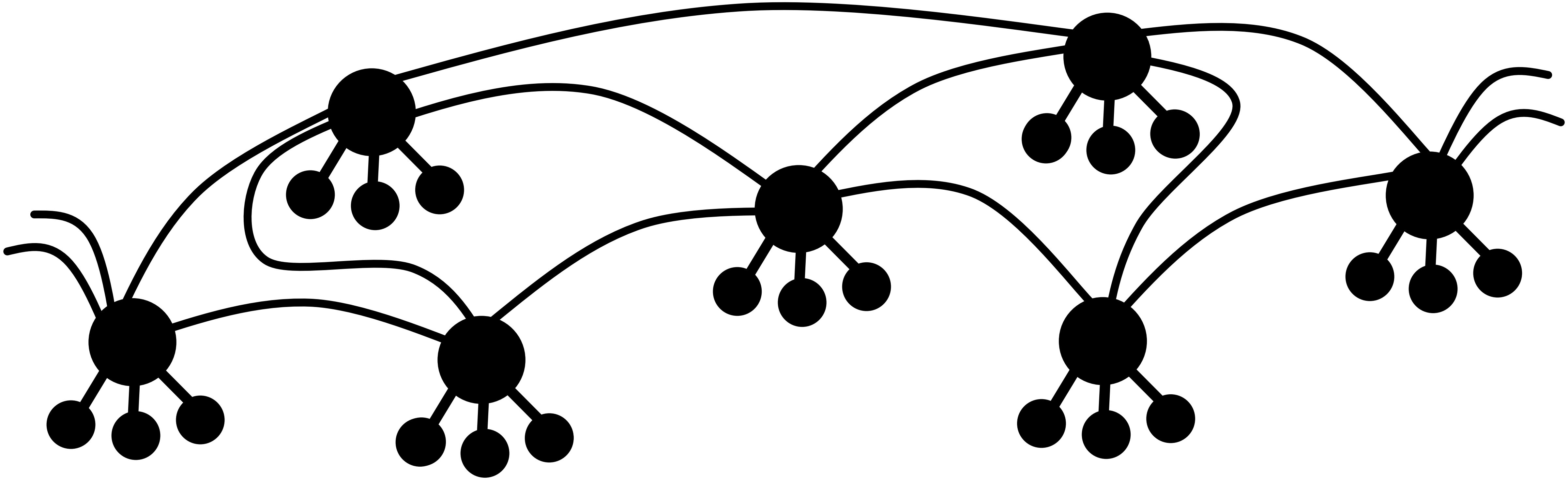
Random uniform

Off-diagonal

Stencil

Shuffle

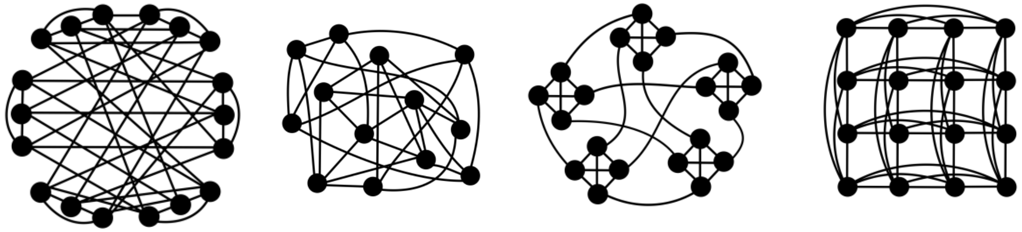
...



How many multiple paths do we need to tackle flow collisions?

Part 1

...For different topologies



...For different workloads

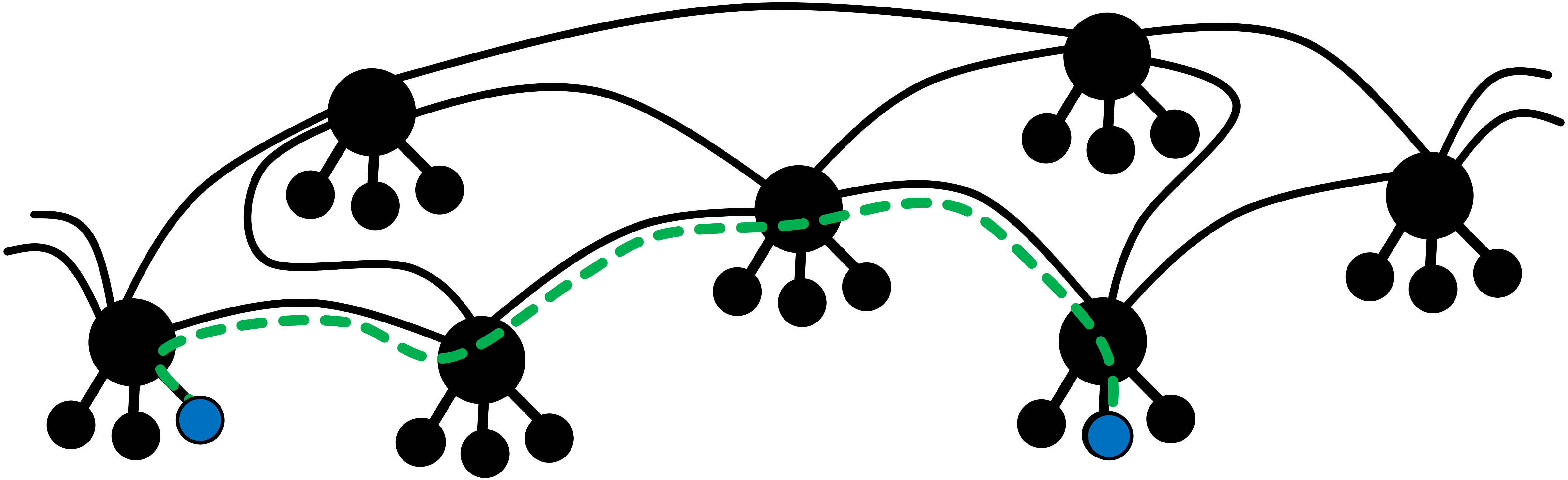
Random uniform

Off-diagonal

Stencil

Shuffle

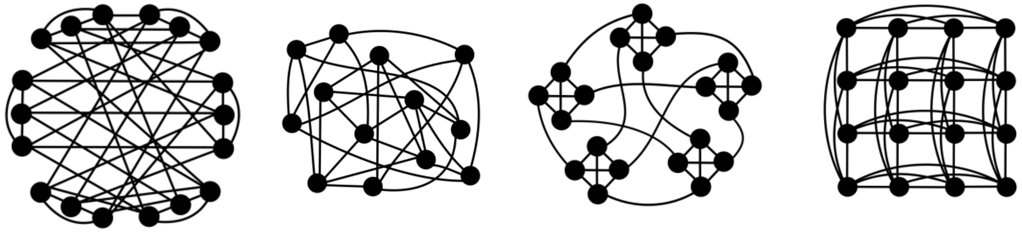
...



HOW MANY MULTIPLE PATHS DO WE NEED TO TACKLE FLOW COLLISIONS?

Part 1

...For different topologies



...For different workloads

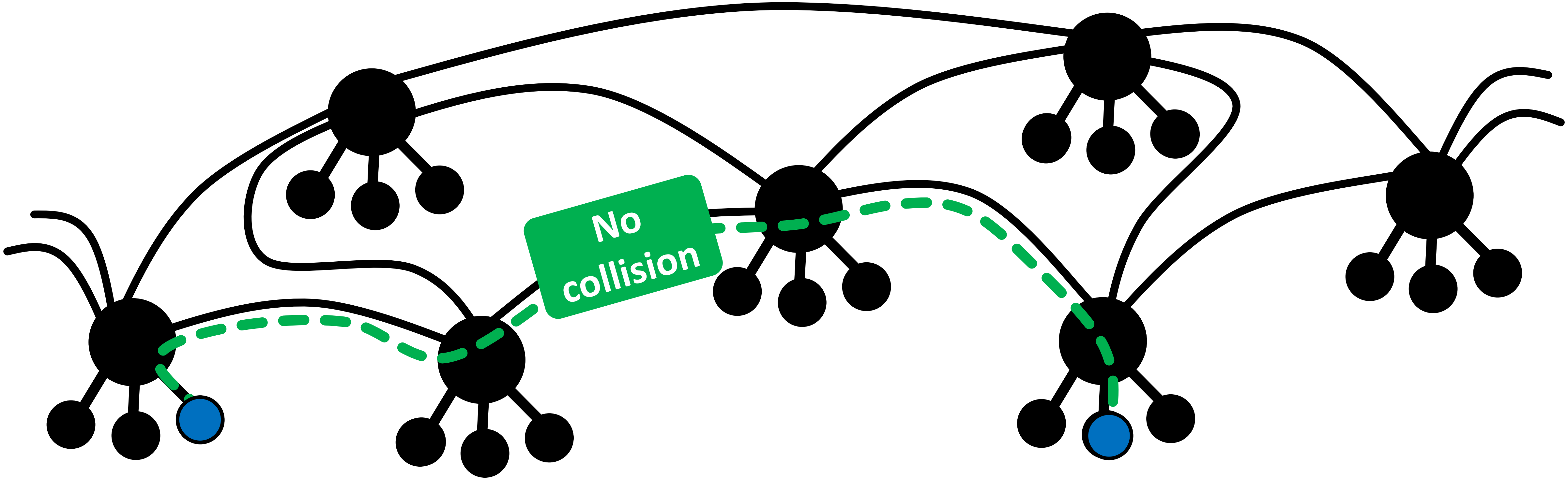
Random uniform

Off-diagonal

Stencil

Shuffle

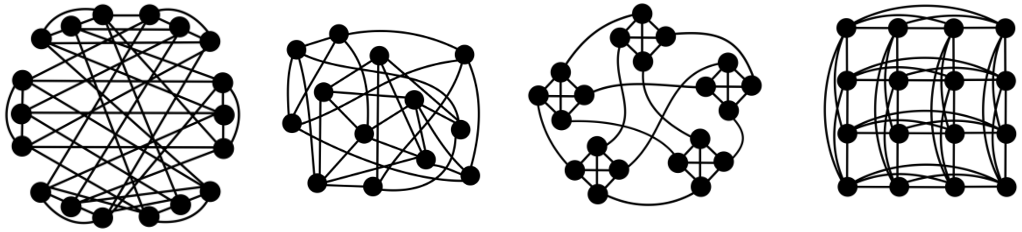
...



How many multiple paths do we need to tackle flow collisions?

Part 1

...For different topologies



...For different workloads

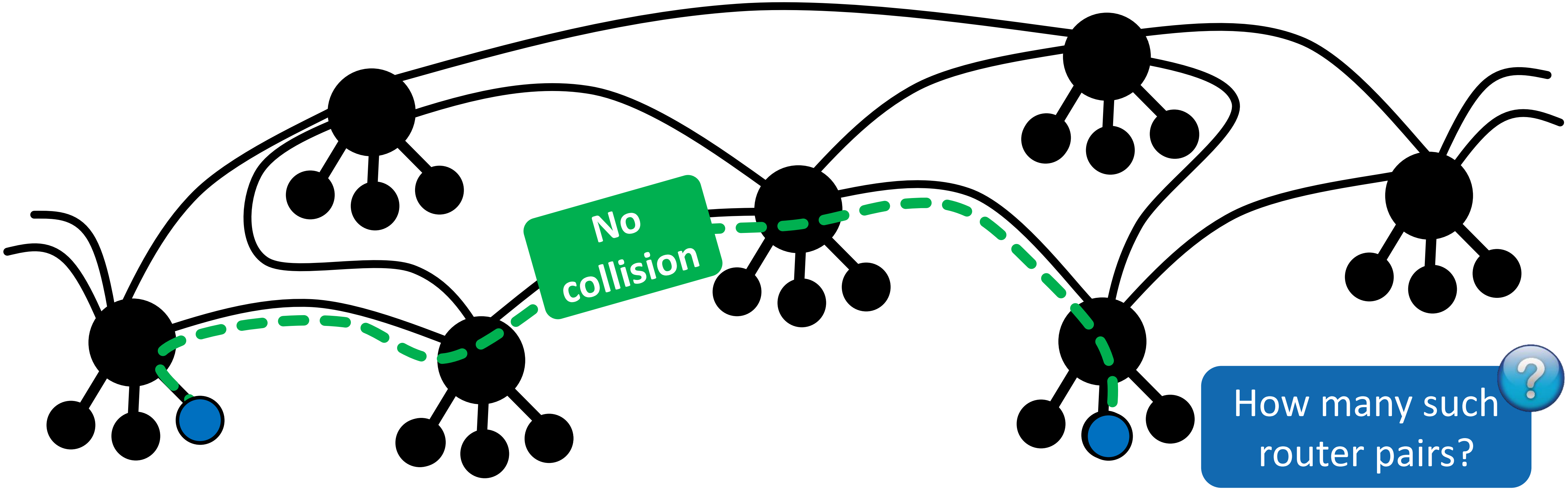
Random uniform

Off-diagonal

...

Stencil

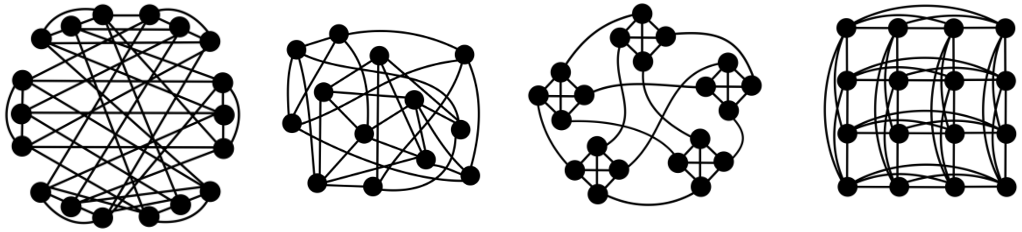
Shuffle



How many multiple paths do we need to tackle flow collisions?

Part 1

...For different topologies



...For different workloads

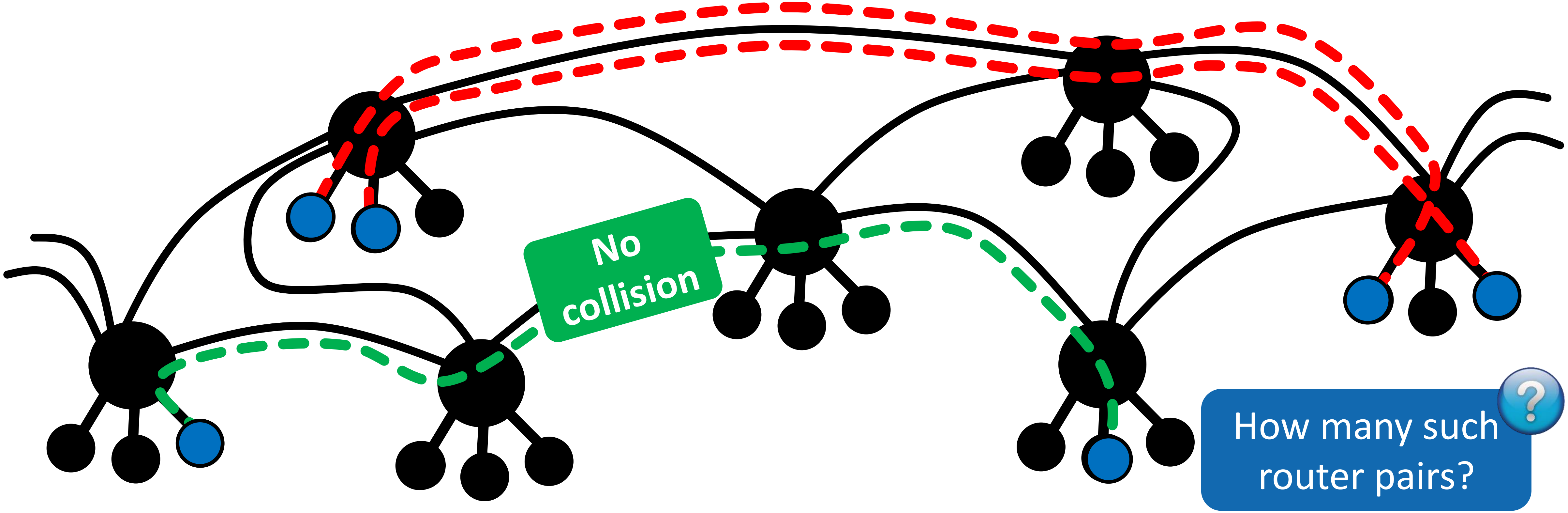
Random uniform

Off-diagonal

Stencil

Shuffle

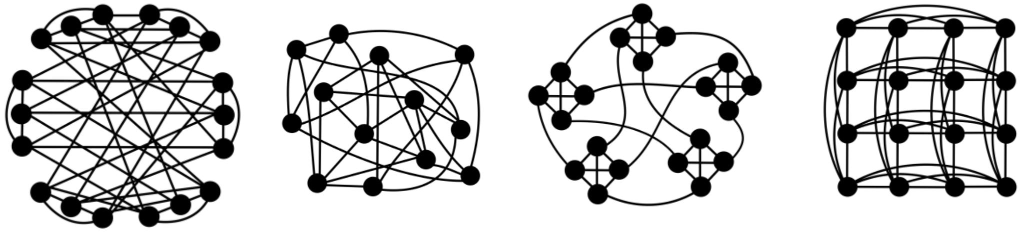
...



HOW MANY MULTIPLE PATHS DO WE NEED TO TACKLE FLOW COLLISIONS?

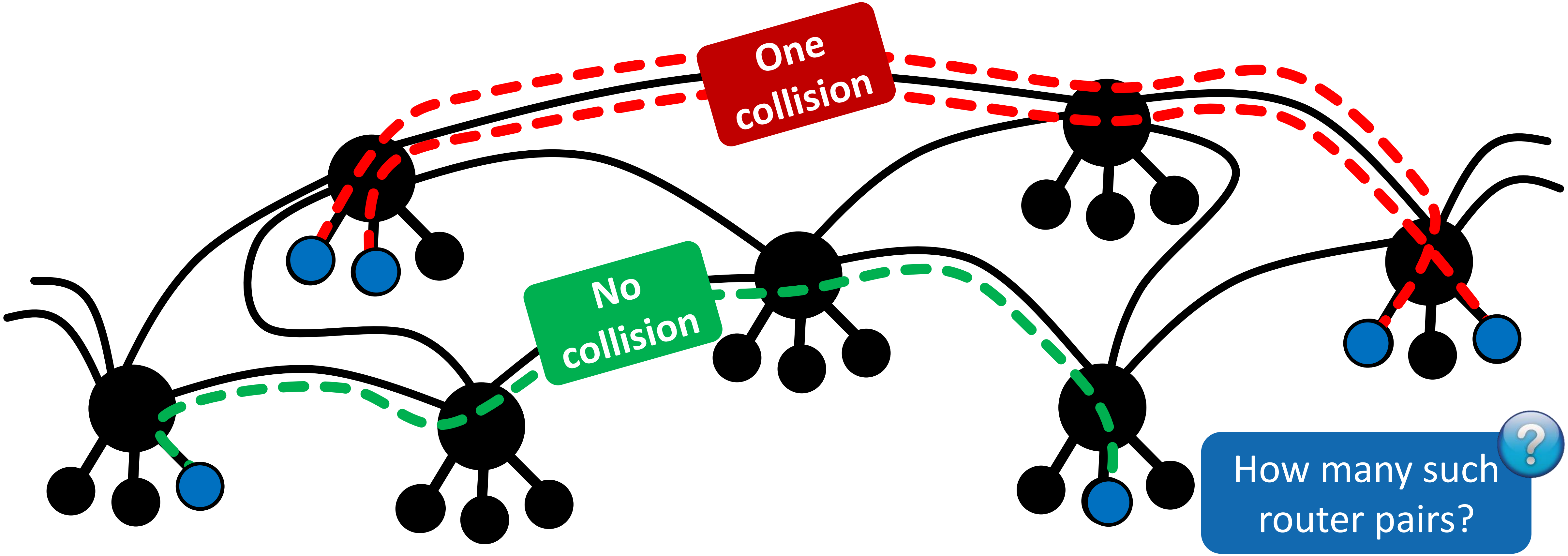
Part 1

...For different topologies



...For different workloads

Random uniform | Off-diagonal | Stencil | Shuffle | ...

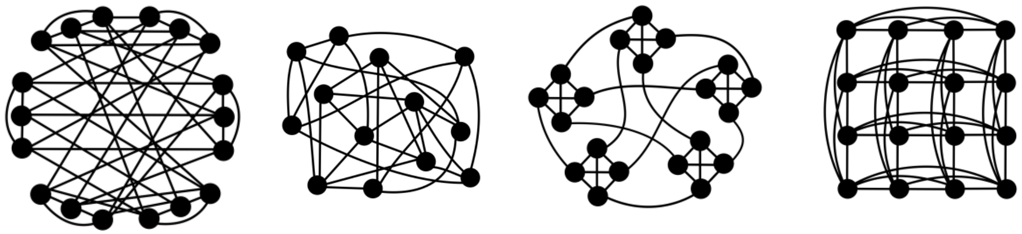


How many such router pairs?

How many multiple paths do we need to tackle flow collisions?

Part 1

...For different topologies



...For different workloads

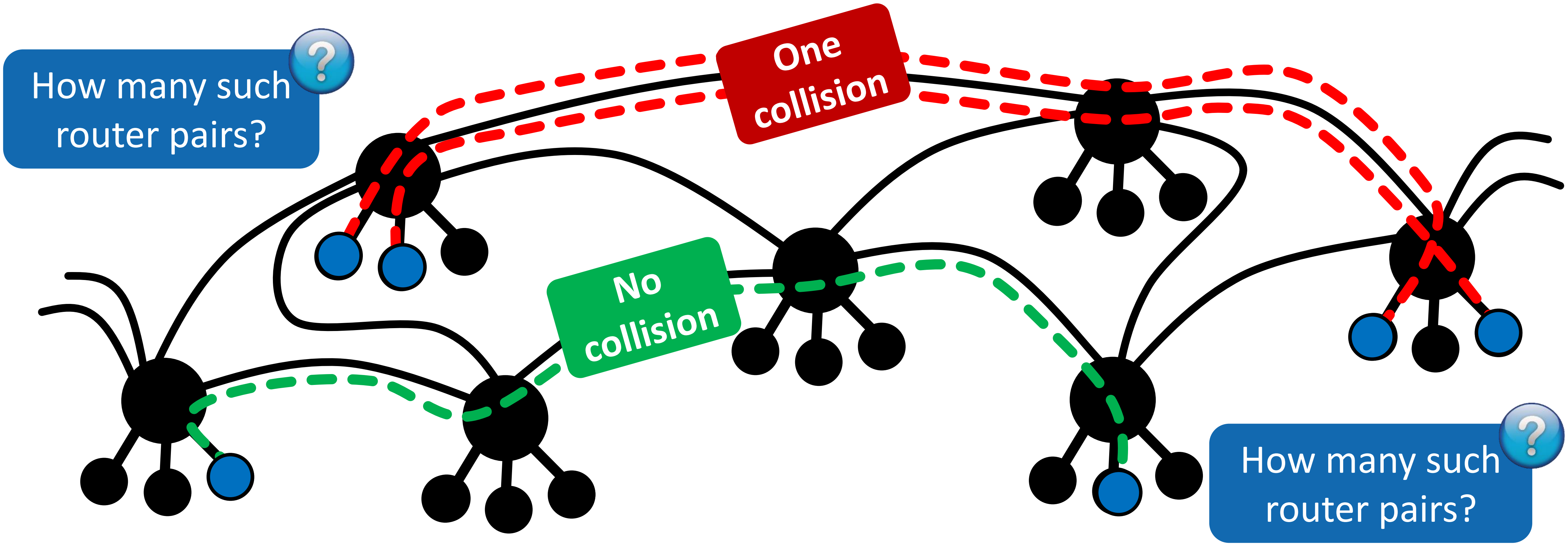
Random uniform

Off-diagonal

...

Stencil

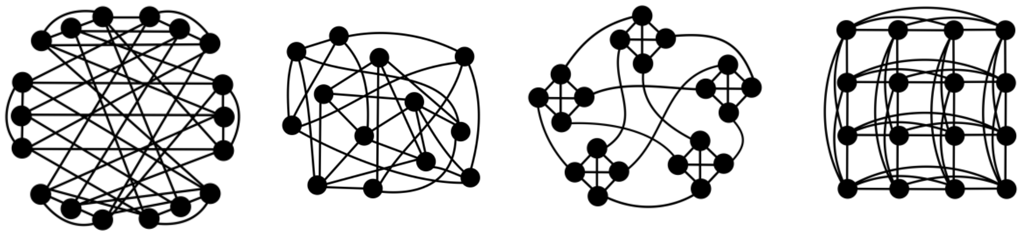
Shuffle



How many multiple paths do we need to tackle flow collisions?

Part 1

...For different topologies



...For different workloads

Random uniform

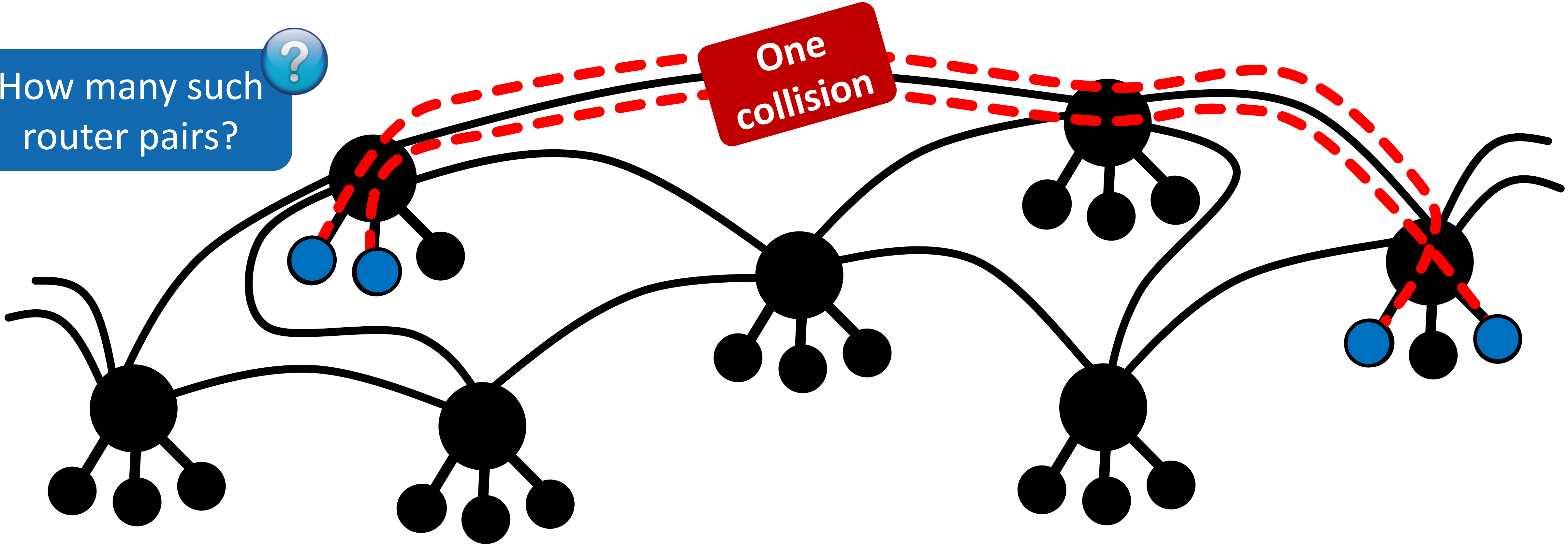
Off-diagonal

...

Stencil

Shuffle

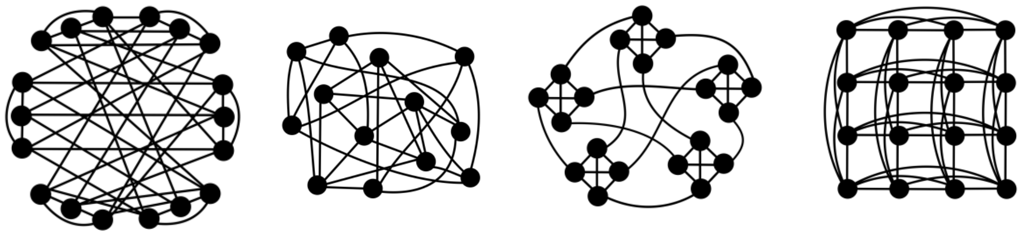
How many such router pairs?



How many multiple paths do we need to tackle flow collisions?

Part 1

...For different topologies



...For different workloads

Random uniform

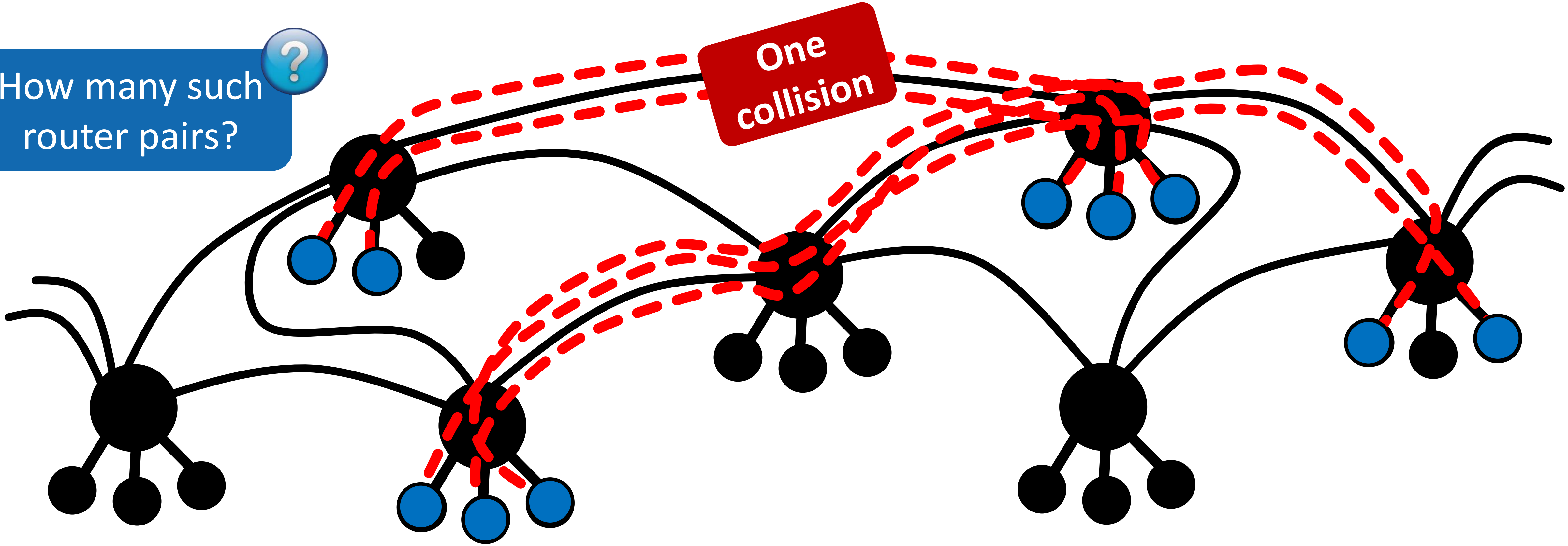
Off-diagonal

Stencil

Shuffle

...

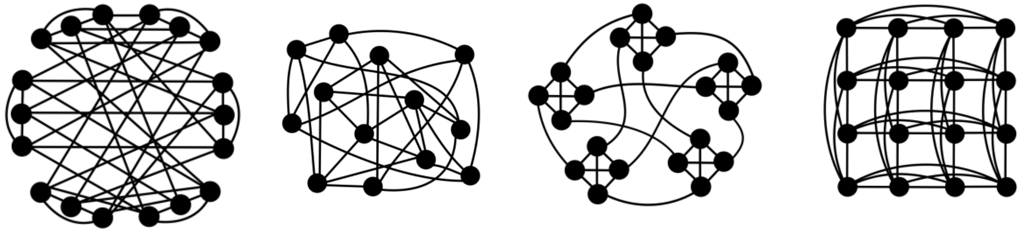
How many such router pairs?



How many multiple paths do we need to tackle flow collisions?

Part 1

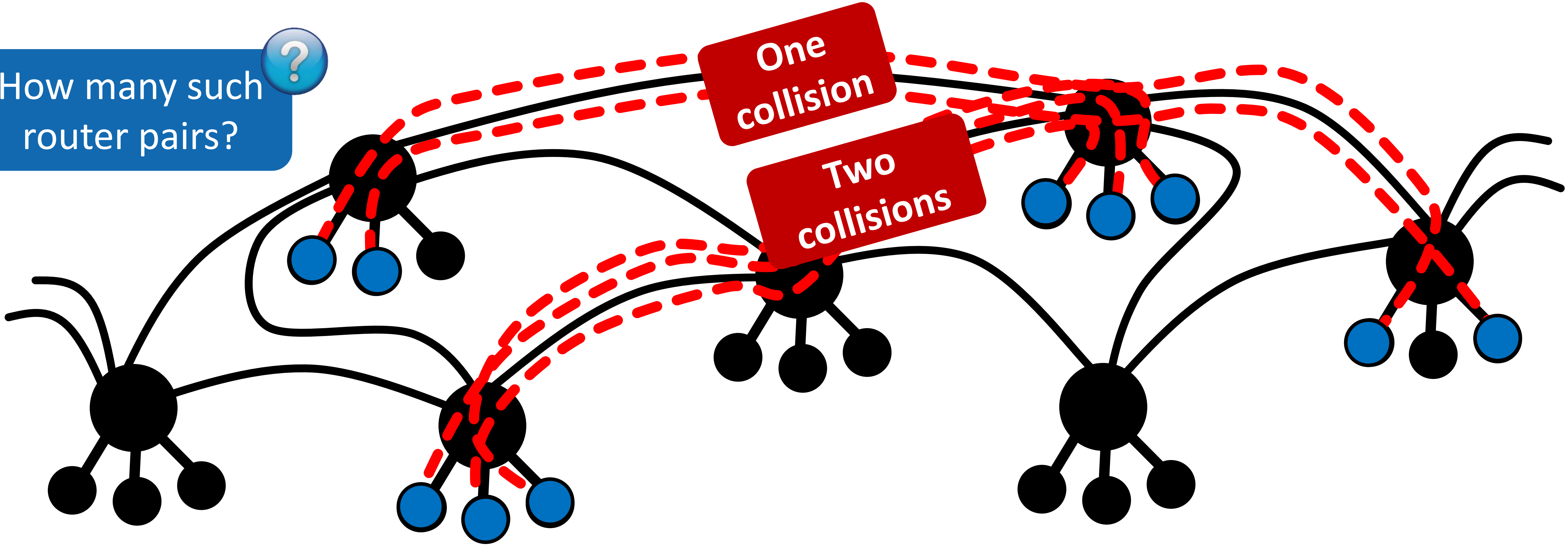
...For different topologies



...For different workloads

- Random uniform
- Off-diagonal
- Stencil
- Shuffle
- ...

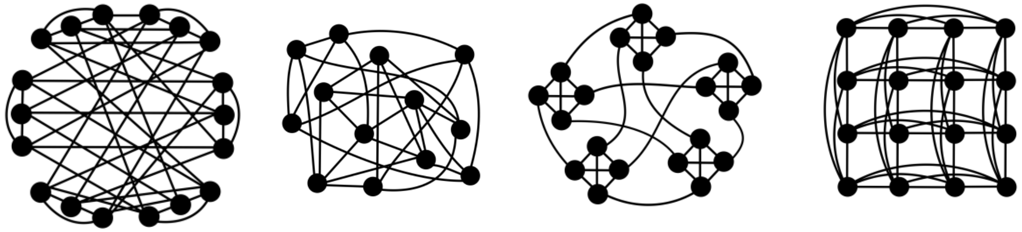
How many such router pairs?



How many multiple paths do we need to tackle flow collisions?

Part 1

...For different topologies



...For different workloads

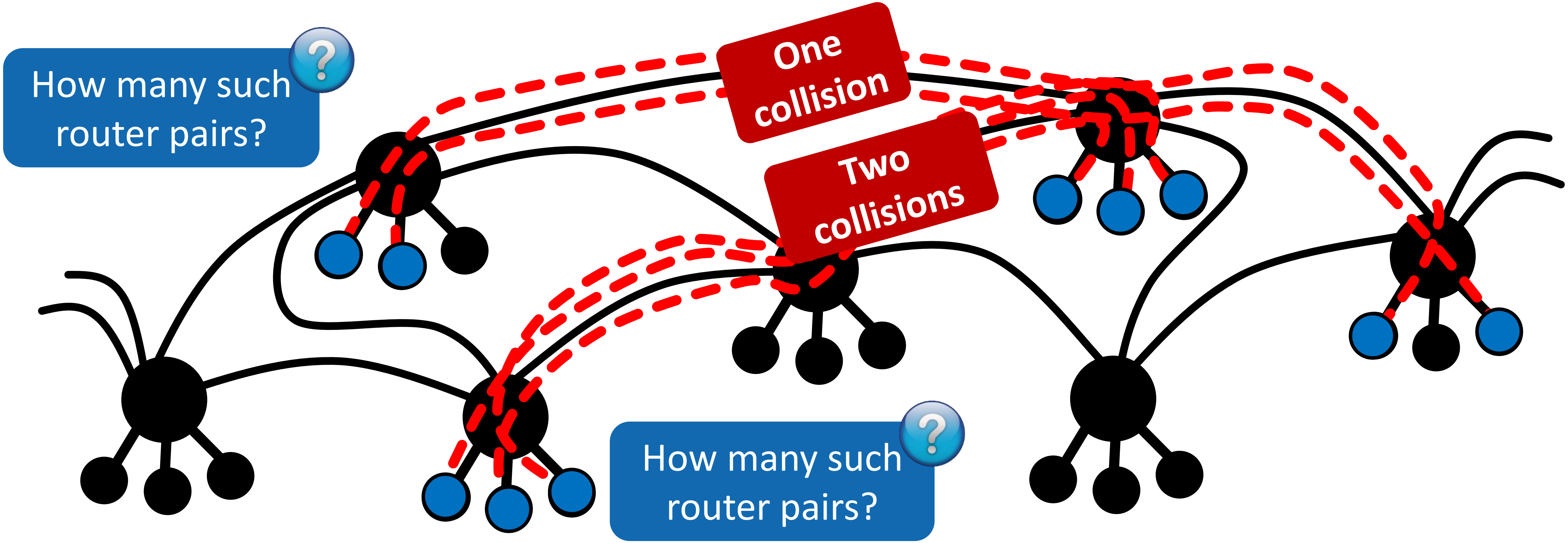
Random uniform

Off-diagonal

...

Stencil

Shuffle

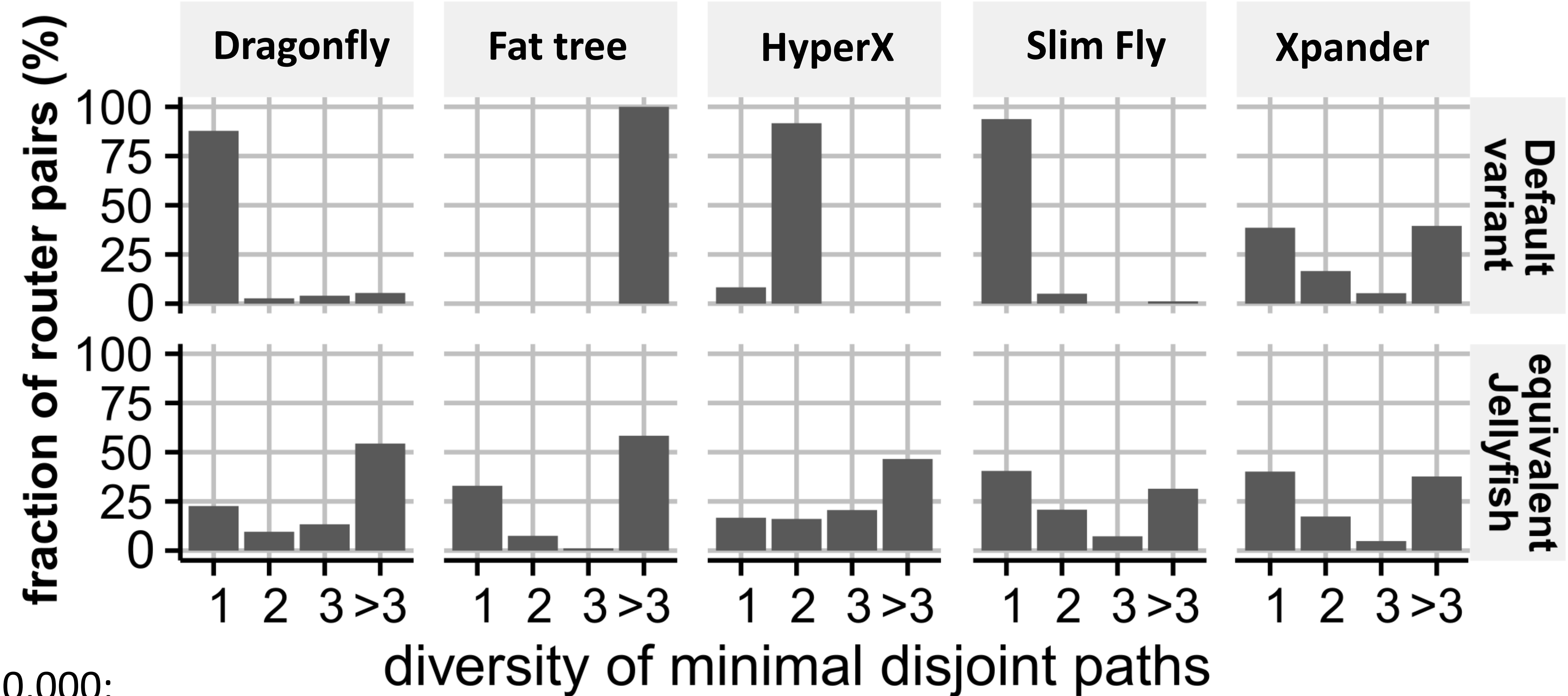


PATH DIVERSITY ANALYSIS

Do we have enough such shortest paths?

Path Diversity Analysis

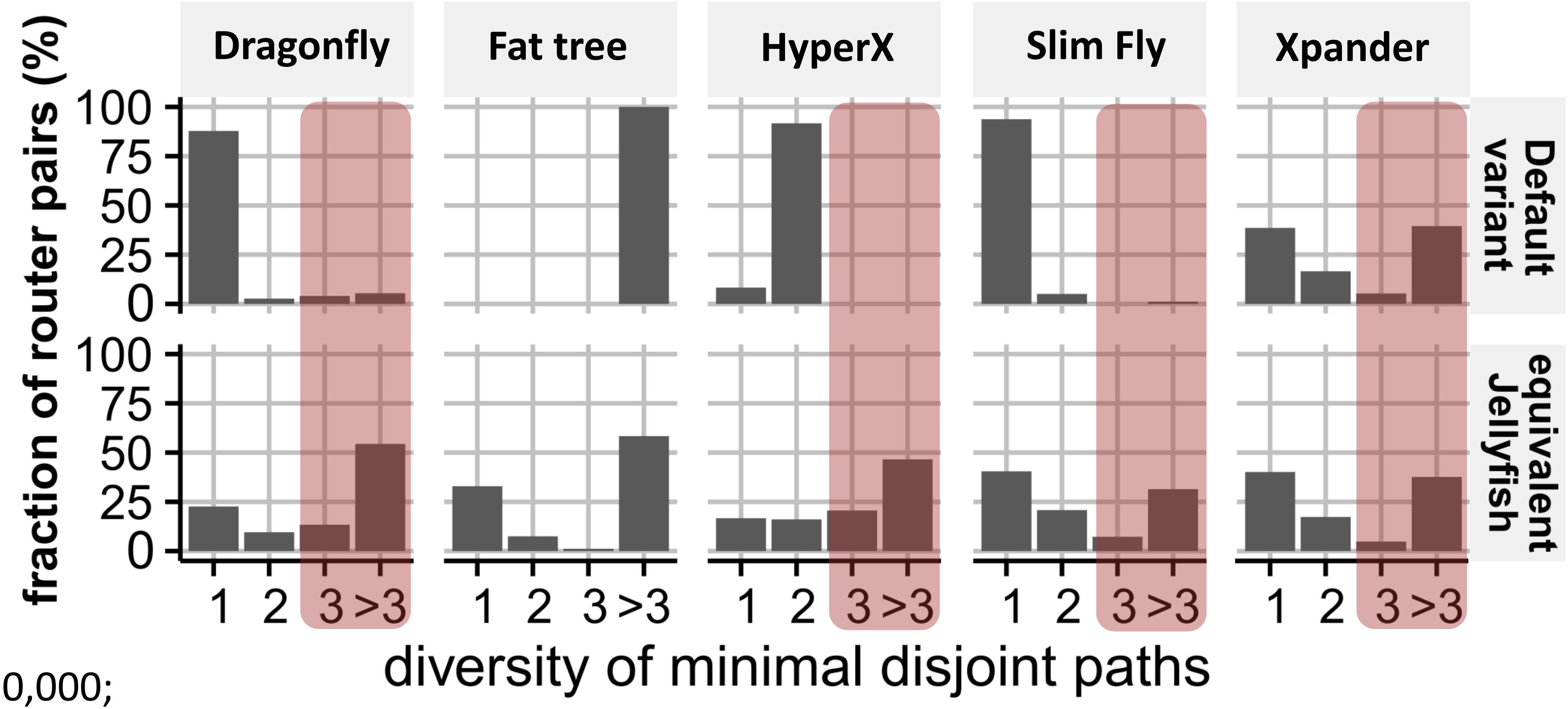
Do we have enough such shortest paths?



N ≈ 10,000;
comparable cost

Path Diversity Analysis

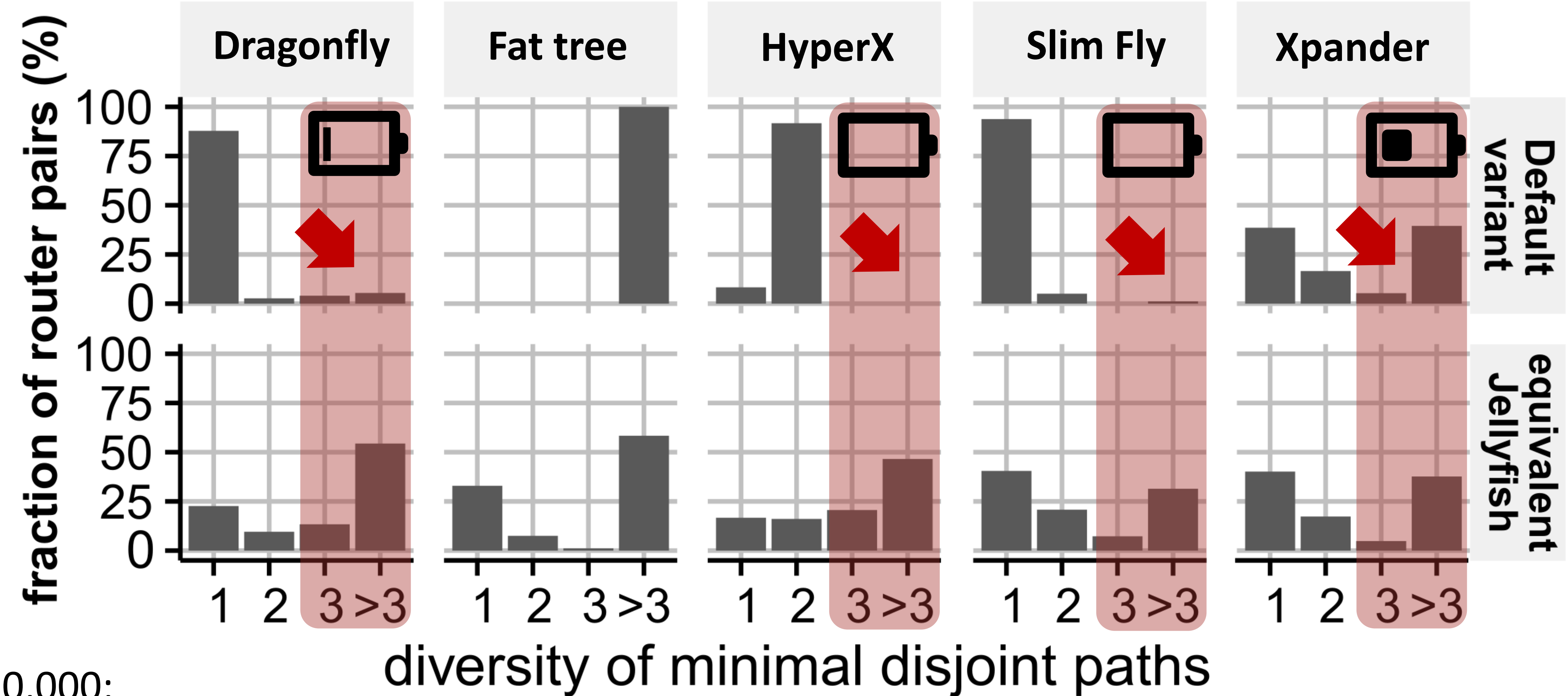
Do we have enough such shortest paths?



N ≈ 10,000;
comparable cost

Path Diversity Analysis

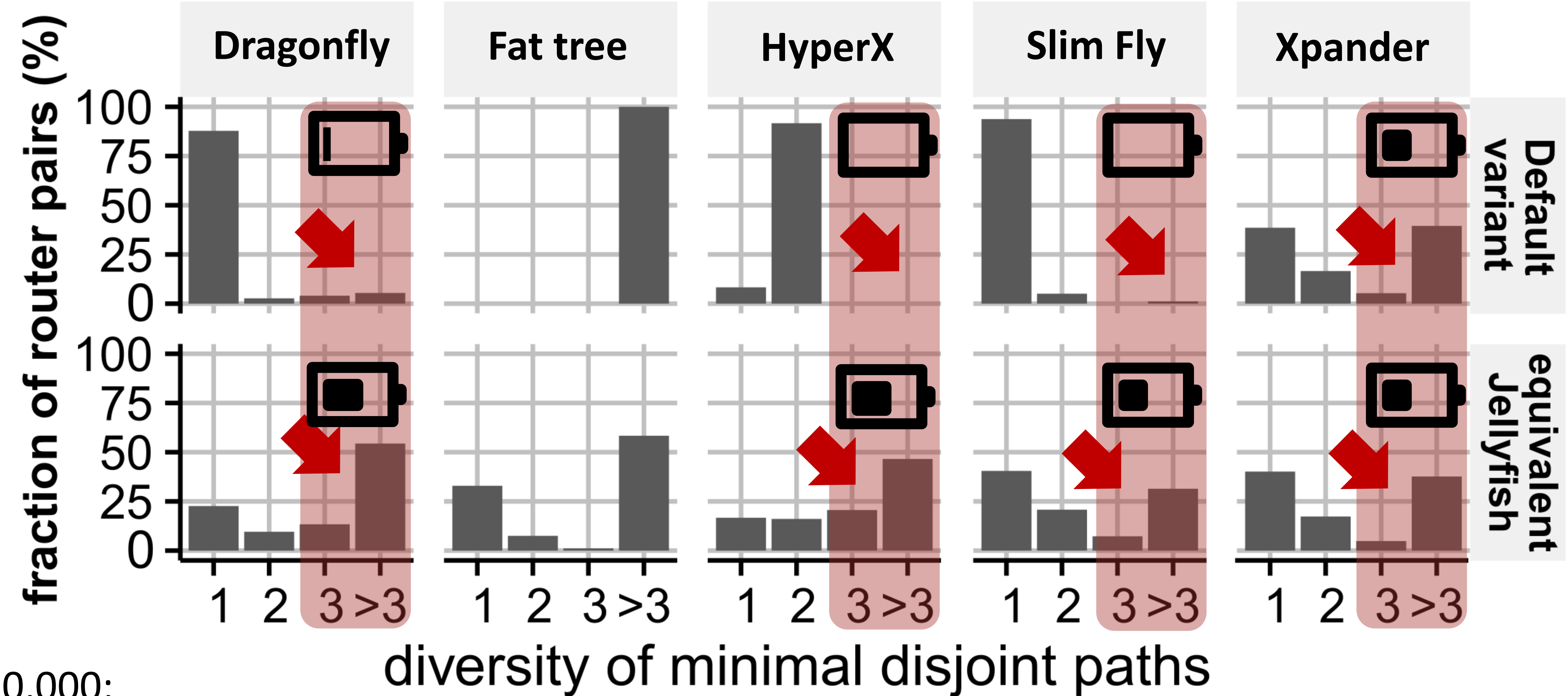
Do we have enough such shortest paths?



N ≈ 10,000;
comparable cost

Path Diversity Analysis

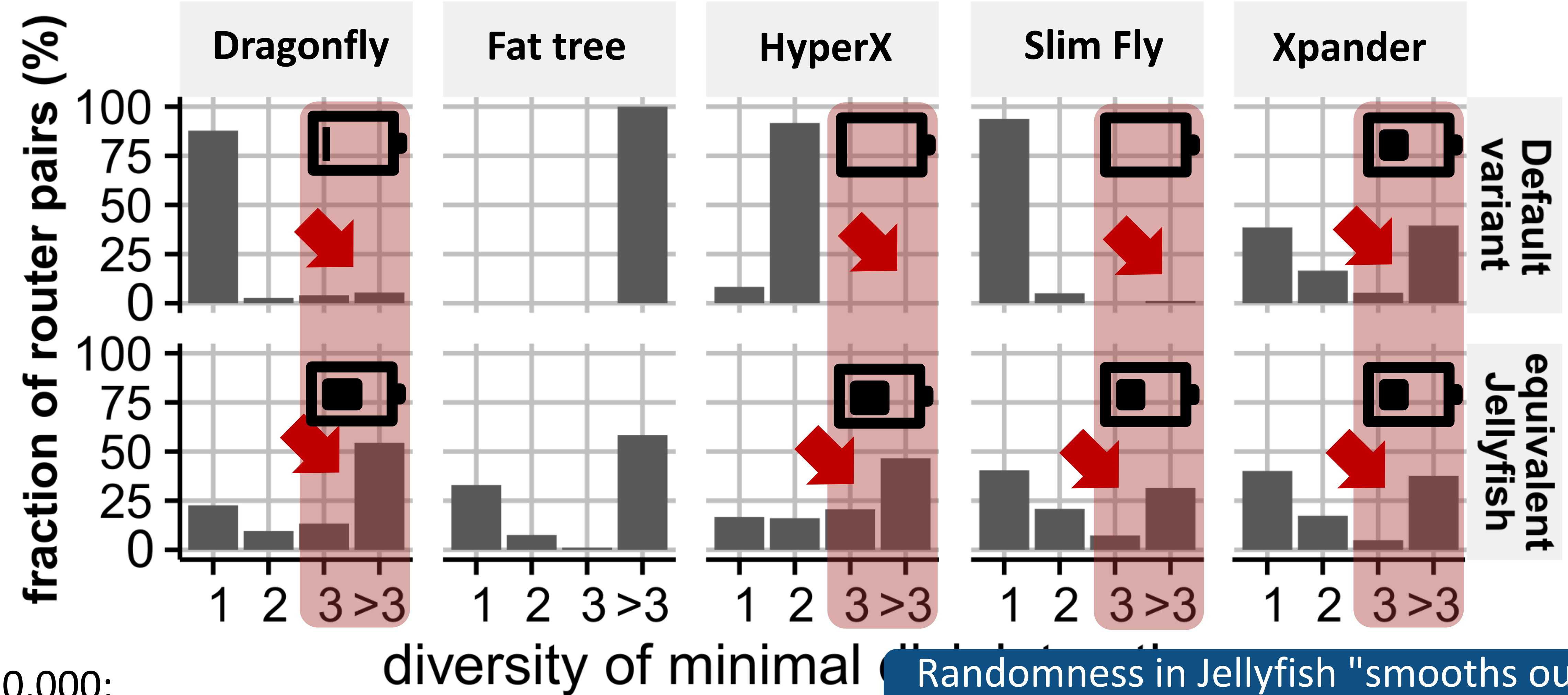
Do we have enough such shortest paths?



N ≈ 10,000;
comparable cost

Path Diversity Analysis

Do we have enough such shortest paths?

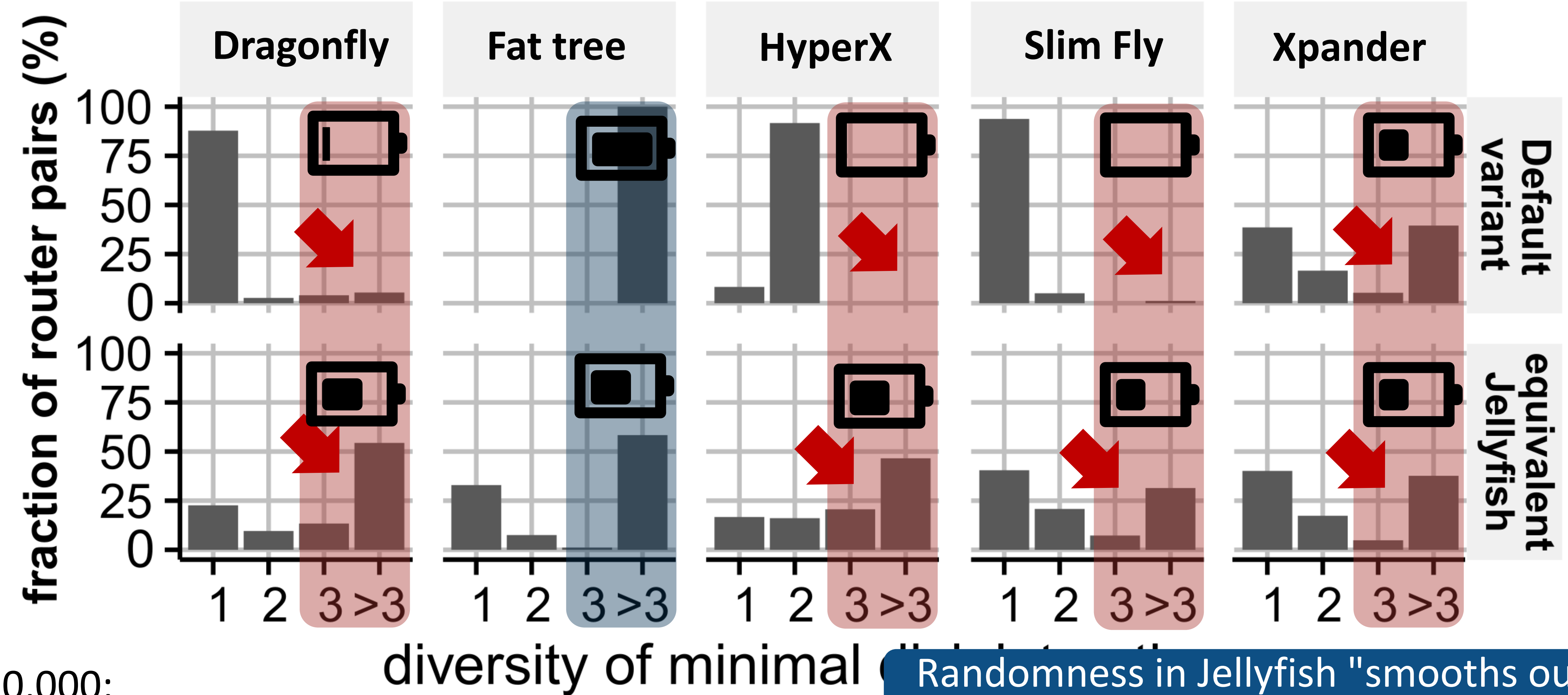


N ≈ 10,000; comparable cost

Randomness in Jellyfish "smooths out" distributions of minimal path diversities

Path Diversity Analysis

Do we have enough such shortest paths?



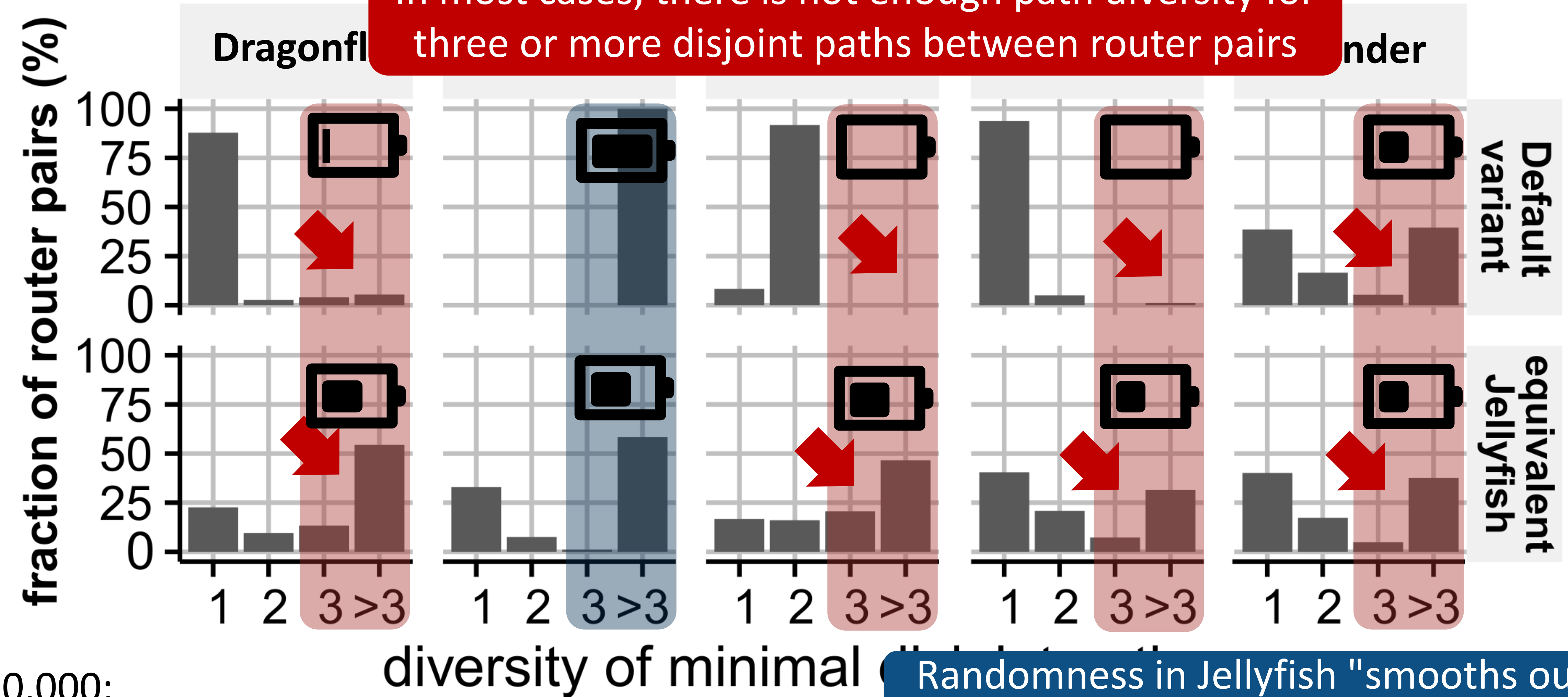
N ≈ 10,000;
comparable cost

Randomness in Jellyfish "smooths out" distributions of minimal path diversities

PATH DIVERSITY ANALYSIS

Do we have enough such shortest paths?

In most cases, there is not enough path diversity for three or more disjoint paths between router pairs

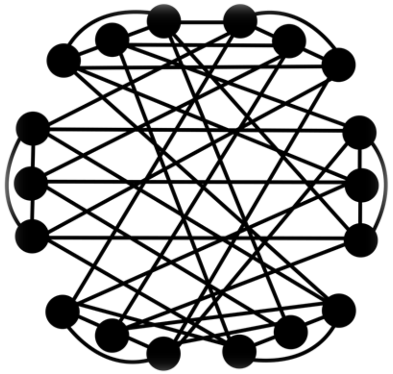


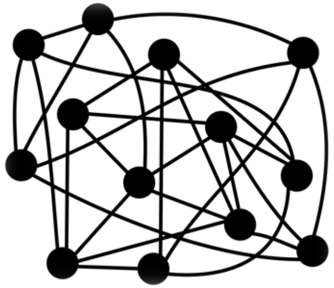
Randomness in Jellyfish "smooths out" distributions of minimal path diversities

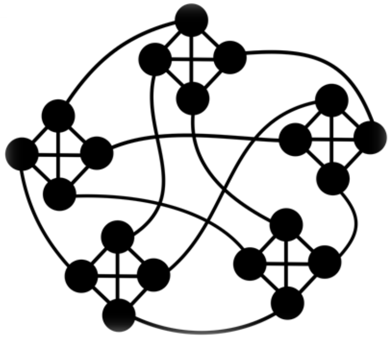
N ≈ 10,000; comparable cost

Low-Diameter Network Topologies

Can we use multipathing?
i.e., are the multiple
paths between routers?



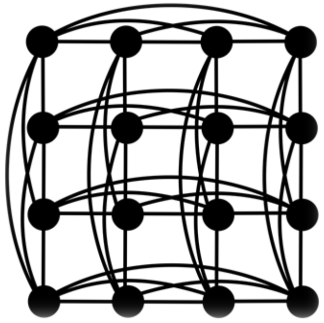




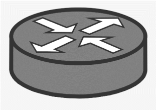
... shortest

... equal length

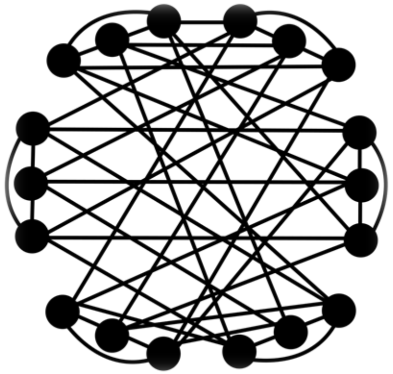
... Between all
router pairs

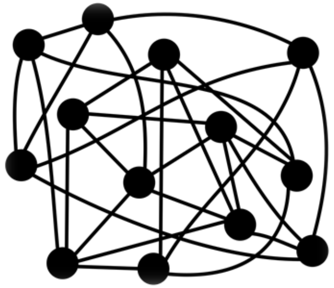


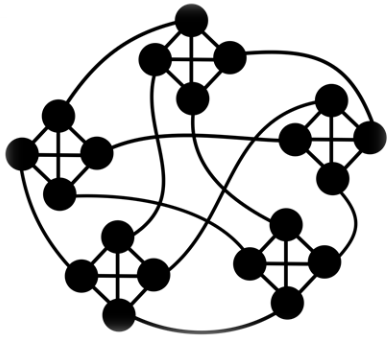
Low-Diameter Network Topologies

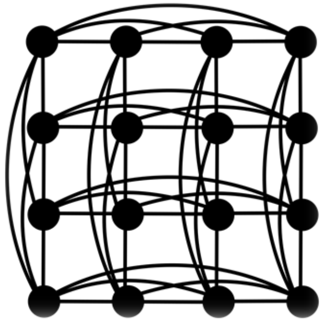


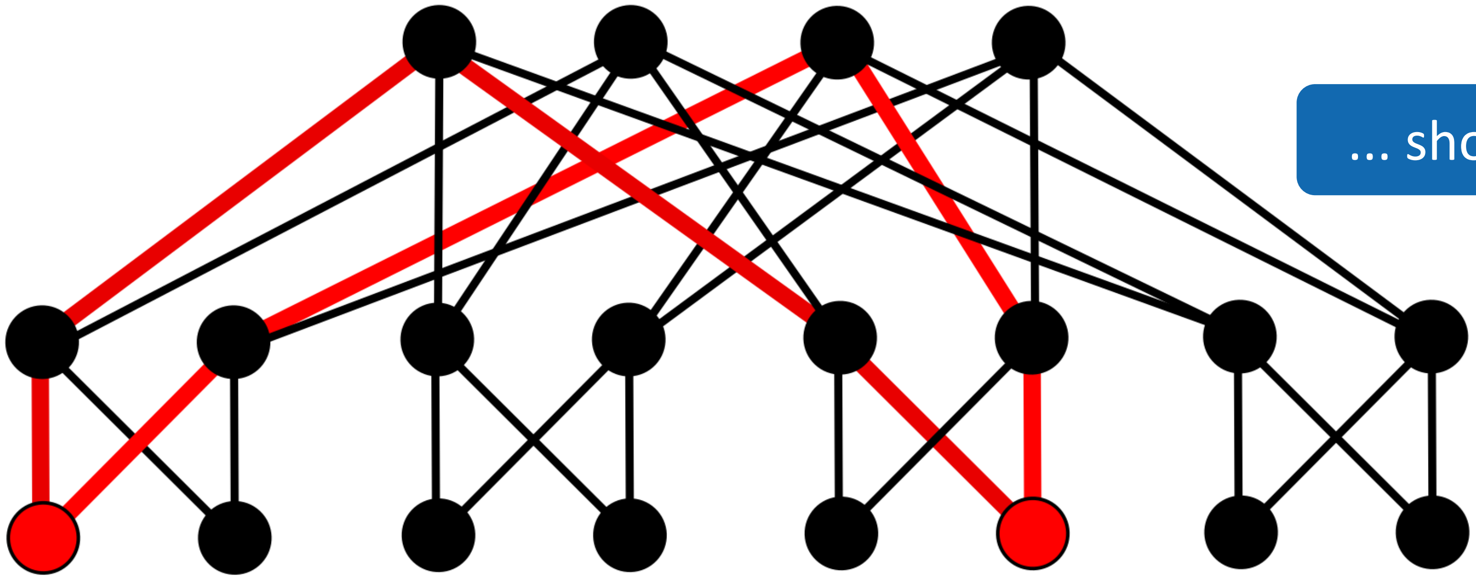
Can we use multipathing? i.e., are the multiple paths between routers?









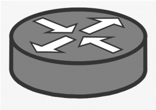


... shortest

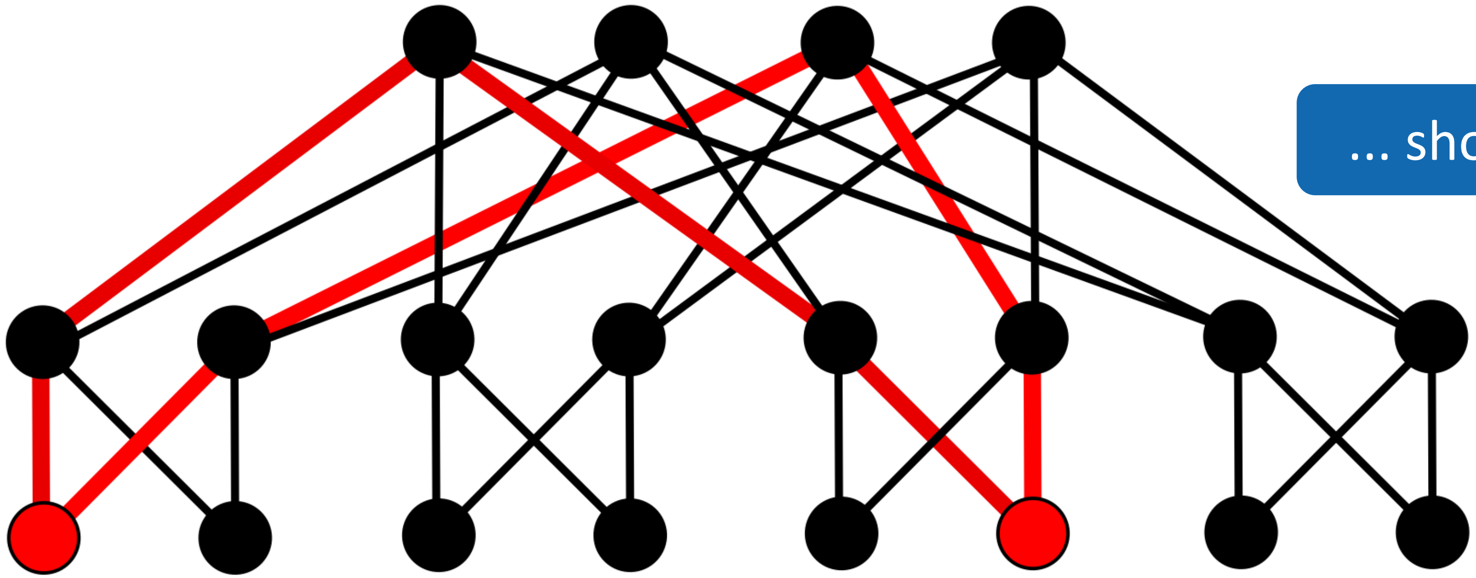
... equal length

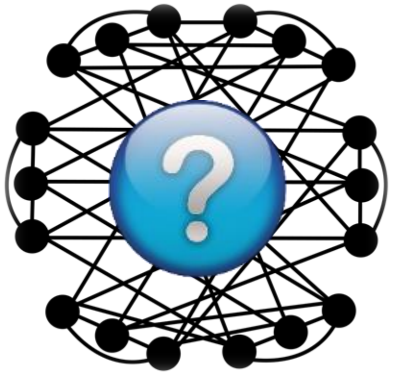
... Between all router pairs

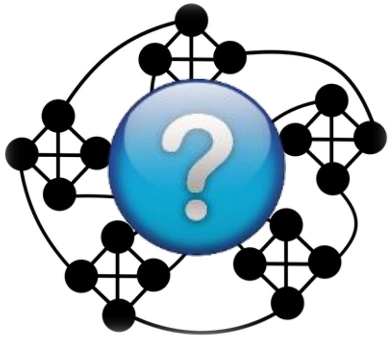
Low-Diameter Network Topologies



Can we use multipathing?
i.e., are the multiple paths between routers?





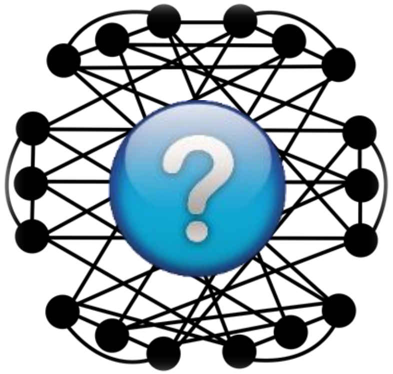


... shortest

... equal length

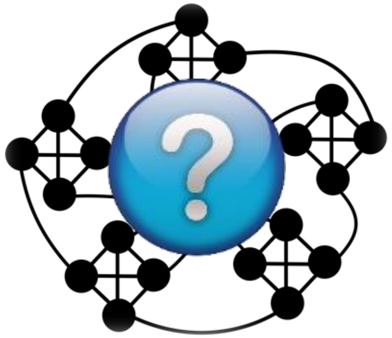
... Between all router pairs

Low-Diameter Network Topologies



Can we use multipathing? i.e., are the multiple paths between routers?

Part 2

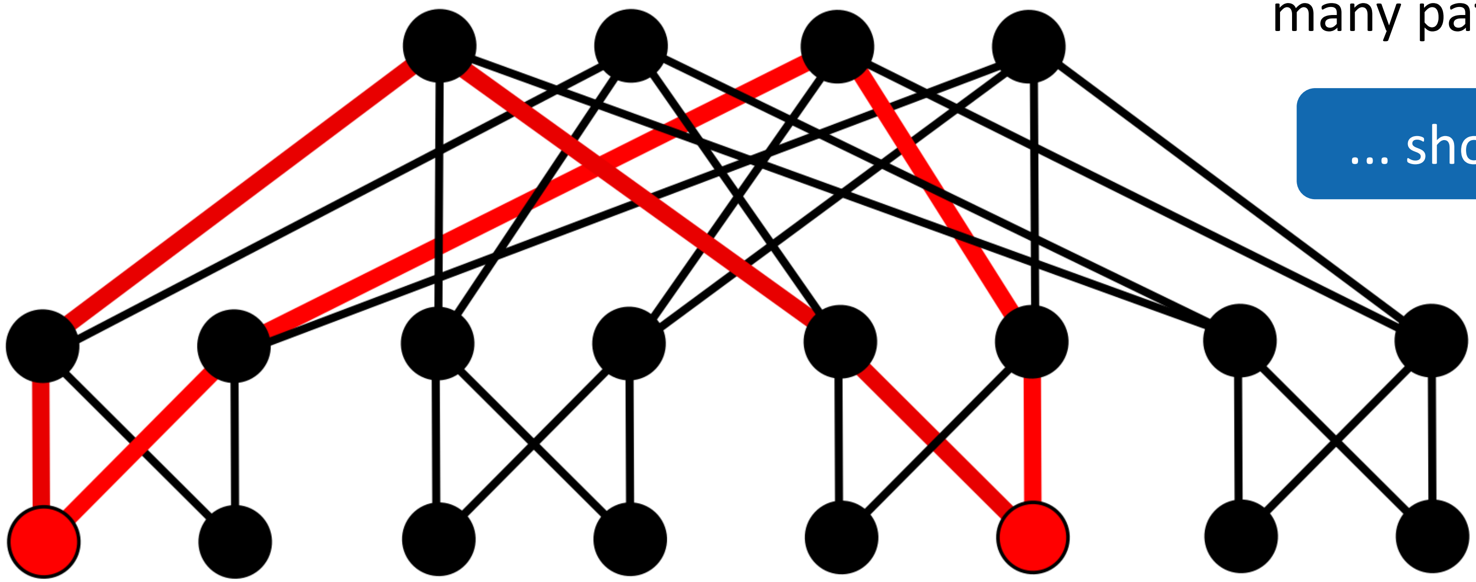


In Fat trees, easily, as we have many paths...

... shortest

... equal length

... Between all router pairs



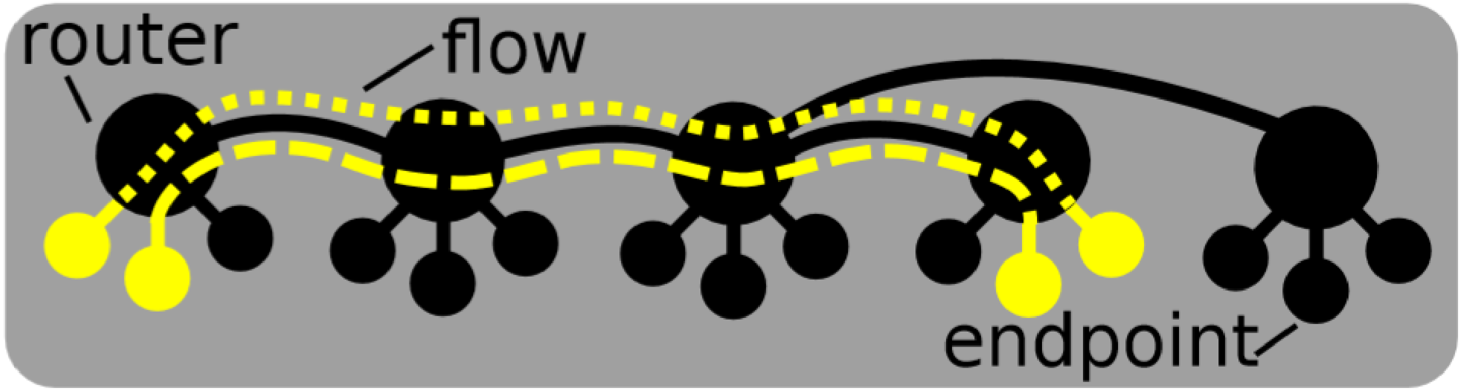
Path Diversity Analysis

Path Diversity Analysis

What are the problems to be tackled with multipathing?

PATH DIVERSITY ANALYSIS

What are the problems to be tackled with multipathing?

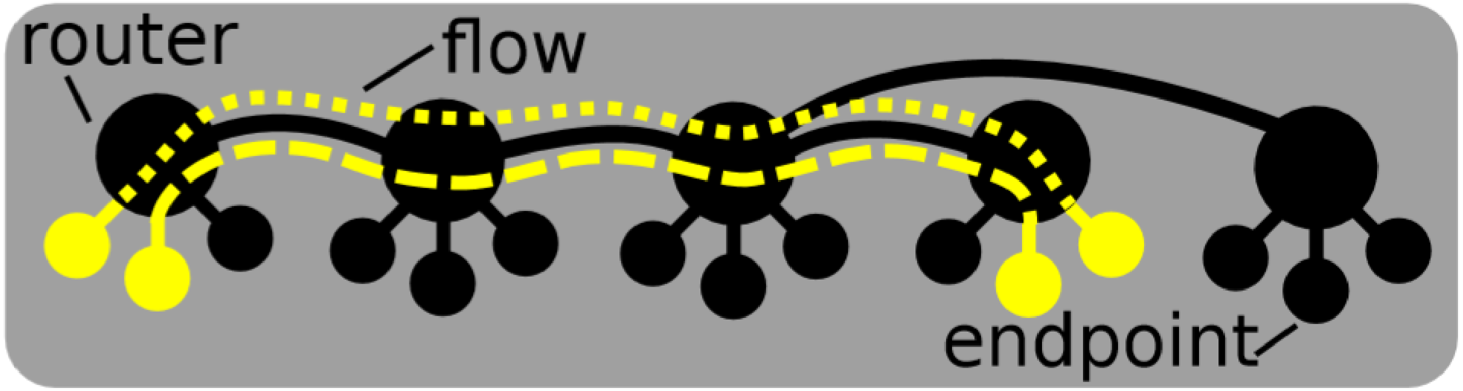


Communicating endpoint pairs mapped to the same routers

Flow collisions

PATH DIVERSITY ANALYSIS

What are the problems to be tackled with multipathing?



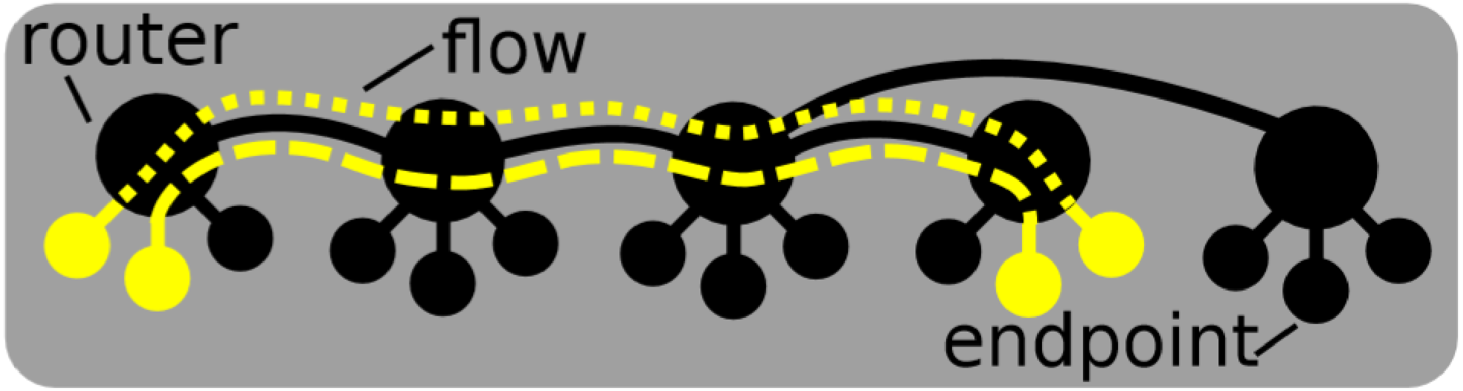
Communicating endpoint pairs mapped to the same routers

Flow collisions

Depend on workload mapping (assignment of communicating endpoints to routers)

PATH DIVERSITY ANALYSIS

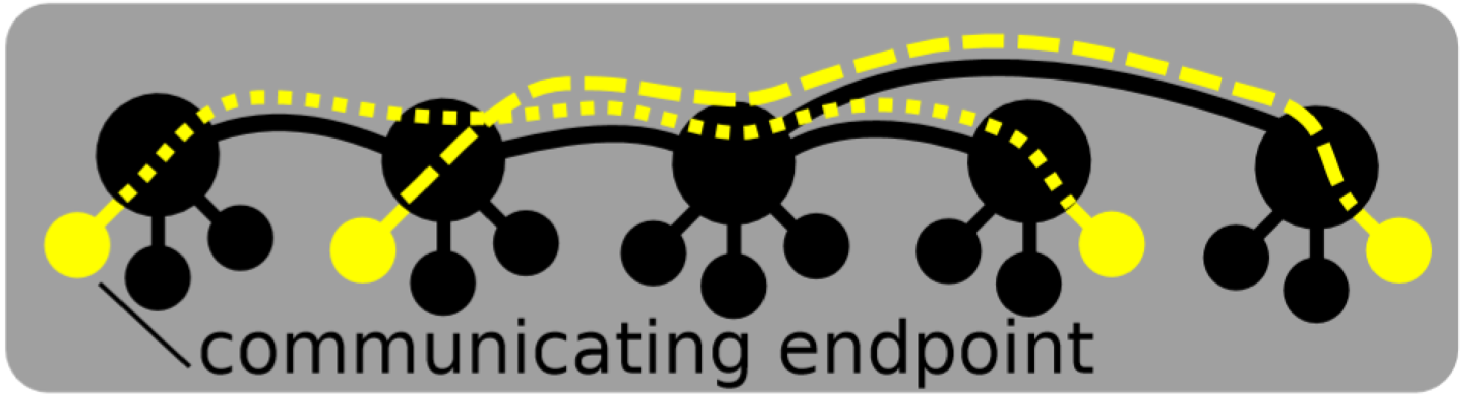
What are the problems to be tackled with multipathing?



Communicating endpoint pairs mapped to the same routers

Flow collisions

Depend on workload mapping (assignment of communicating endpoints to routers)

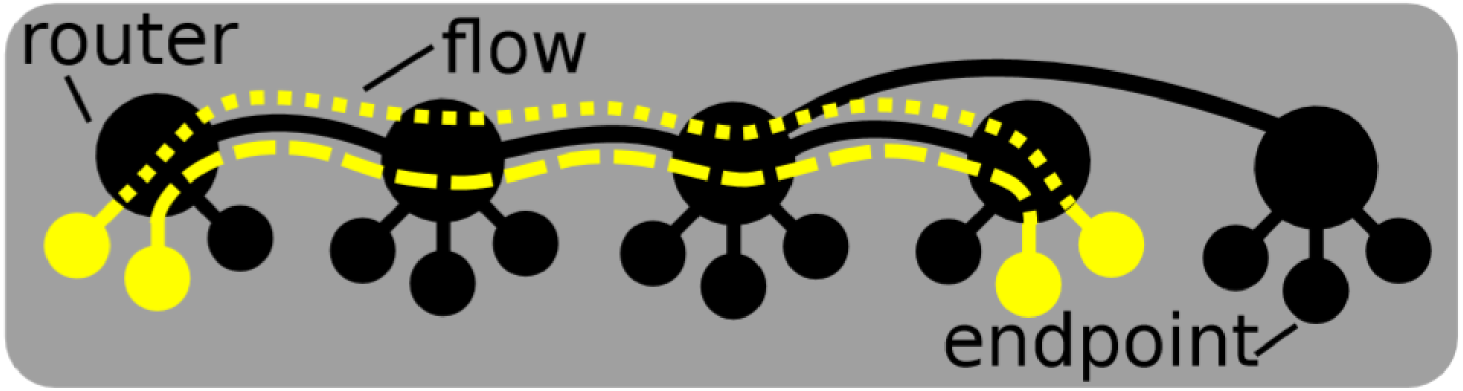


Communicating endpoint pairs mapped to different routers

Flow overlaps

PATH DIVERSITY ANALYSIS

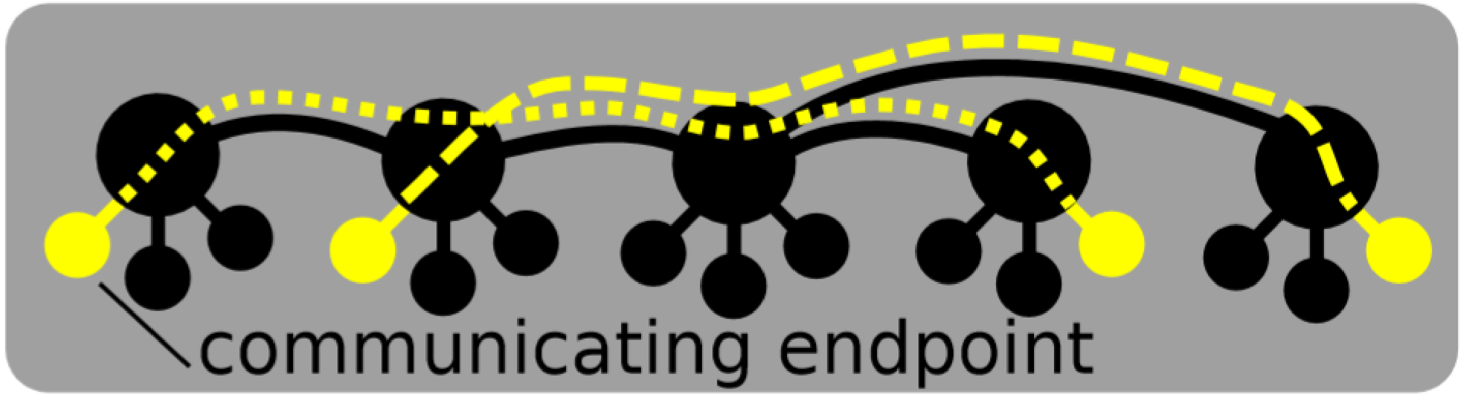
What are the problems to be tackled with multipathing?



Communicating endpoint pairs mapped to the same routers

Flow collisions

Depend on workload mapping (assignment of communicating endpoints to routers)



Communicating endpoint pairs mapped to different routers

Flow overlaps

Depend on topology details (router-router connections)

Path Diversity Analysis

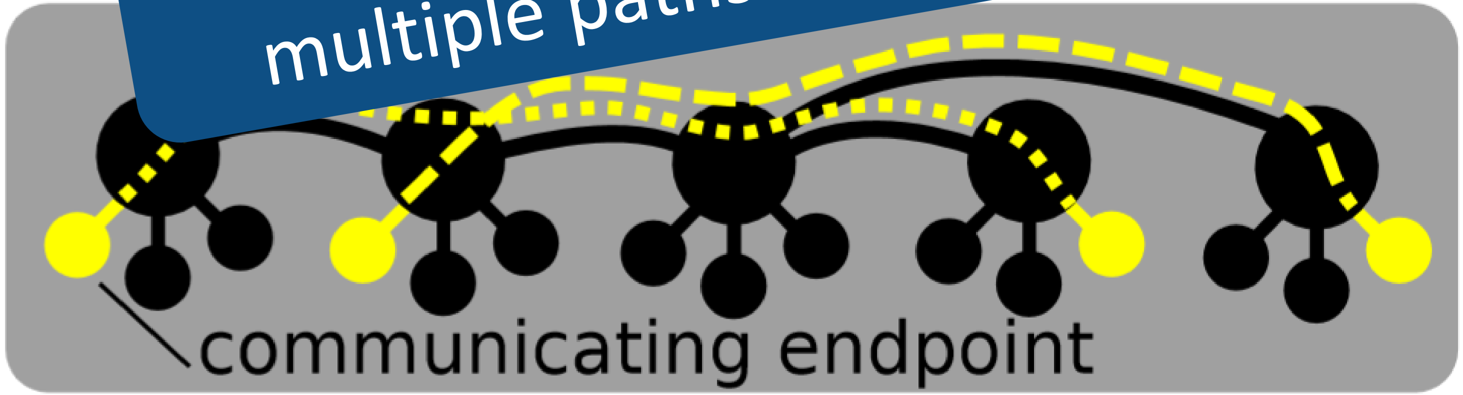
What are the problems to be tackled with multipathing?

Com... ...s

Flow collisions

Depend on workload mapping (assignment of communicating endpoints to routers)

We want to tackle these issues with multipathing. But how many multiple paths do we need…?



Communicating endpoint pairs mapped to different routers

Flow overlaps

Depend on topology details (router-router connections)

Workload Mapping

Workload Mapping

How to further enhance performance?

Workload Mapping

How to further enhance performance?

Random workload remapping

WORKLOAD MAPPING

How to further enhance performance?

Random workload remapping

Observation: In low-diameter topologies, locality is less important

Workload Mapping

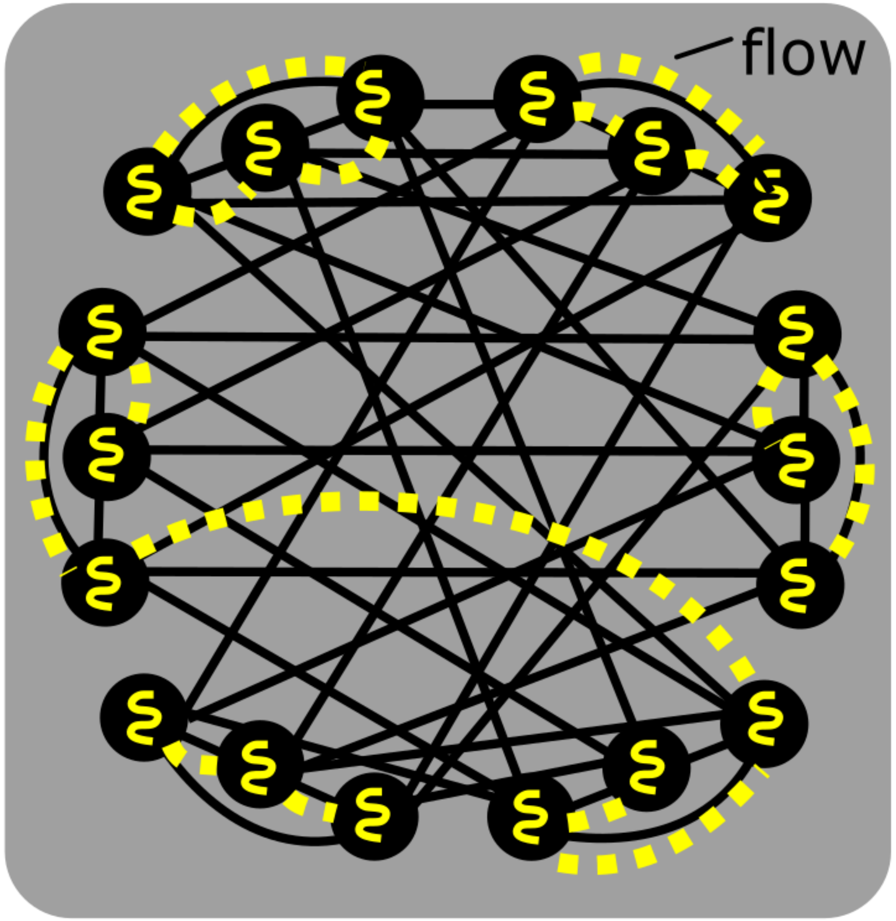
How to further enhance performance?

Random workload remapping

Observation: In low-diameter topologies, locality is less important

A simple stencil running on a Slim Fly

Before remapping



Workload Mapping

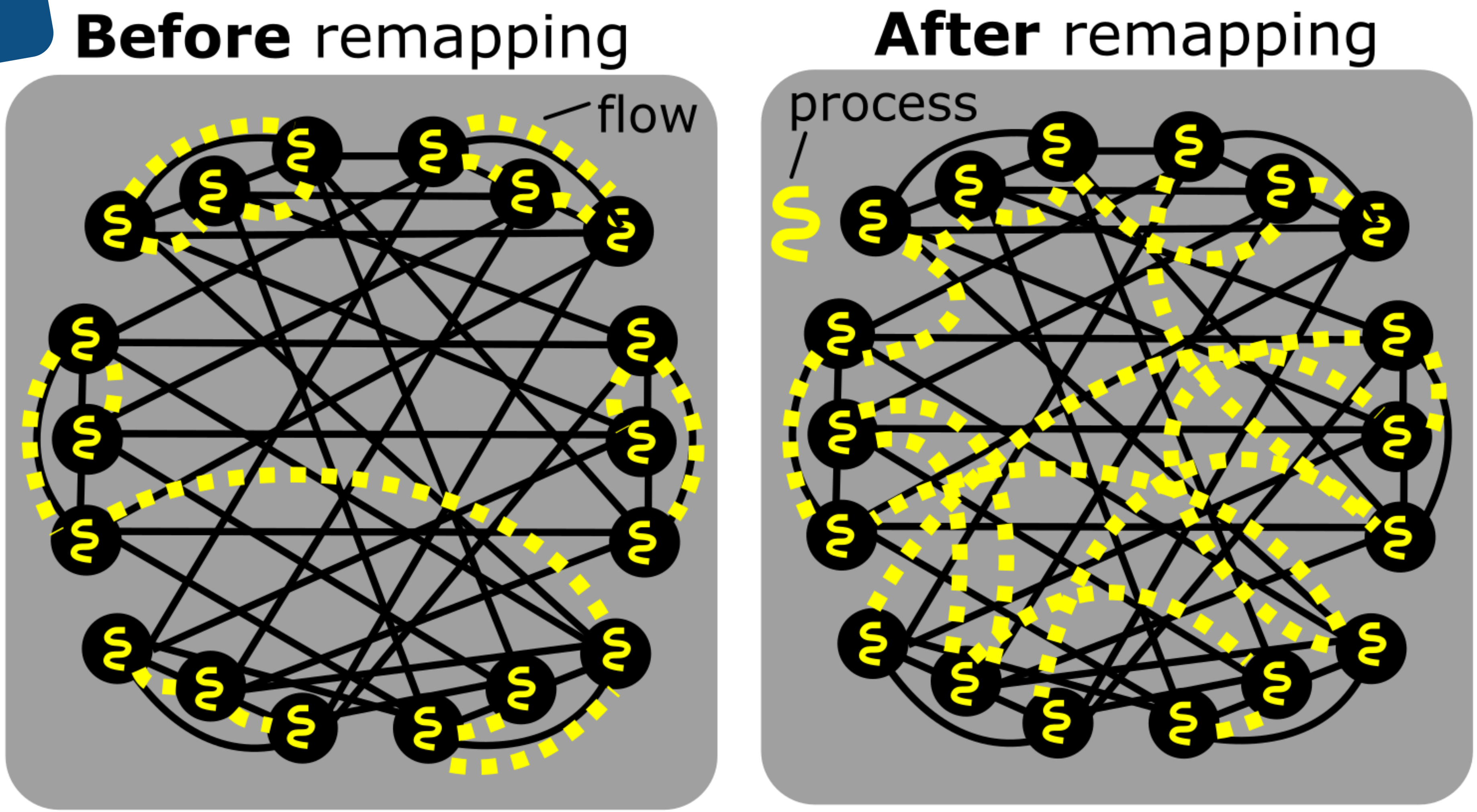
How to further enhance performance?

Random workload remapping

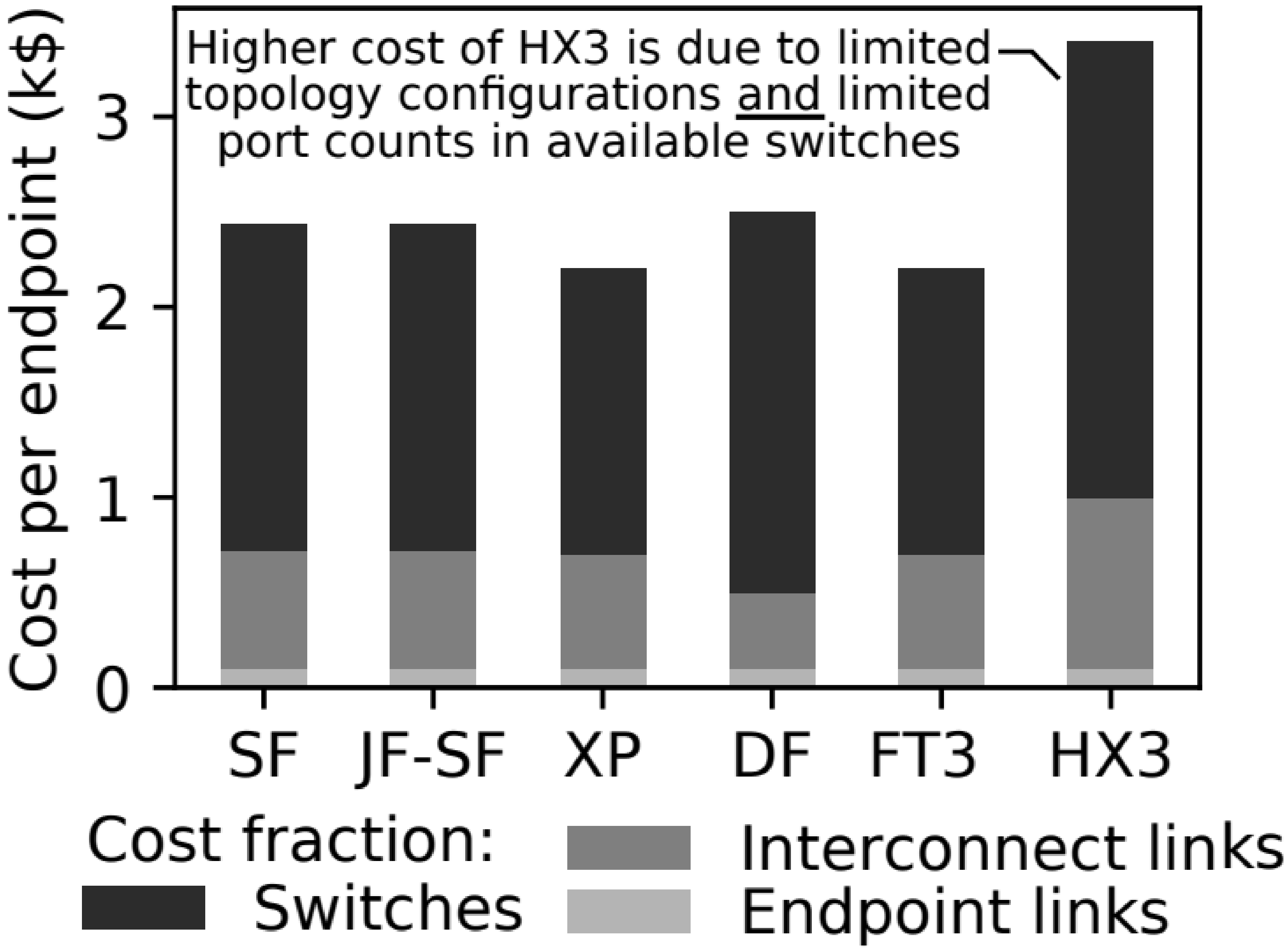
Observation: In low-diameter topologies, locality is less important

Random mapping uses rich diversity of inter-group paths

A simple stencil running on a Slim Fly



COST MODEL


Higher cost of HX3 is due to limited topology configurations and limited port counts in available switches
Cost per endpoint (k$)
0
1
2
3
SF
JF-SF
XP
DF
FT3
HX3
Cost fraction:
Switches
Interconnect links
Endpoint links

EVALUATION

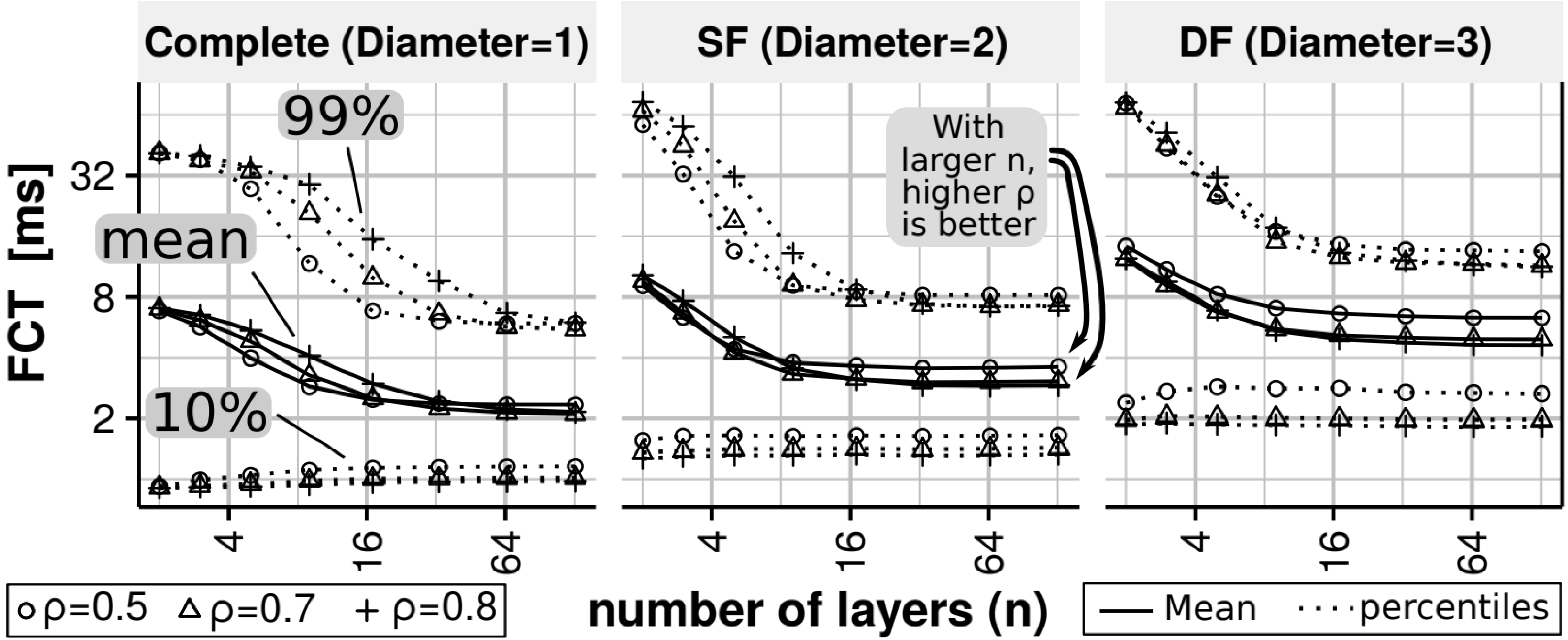
EVALUATION

What layer setup fares best?

EVALUATION

What layer setup fares best?

N ≈ 10,000; comparable cost; random uniform traffic; „bare Ethernet"

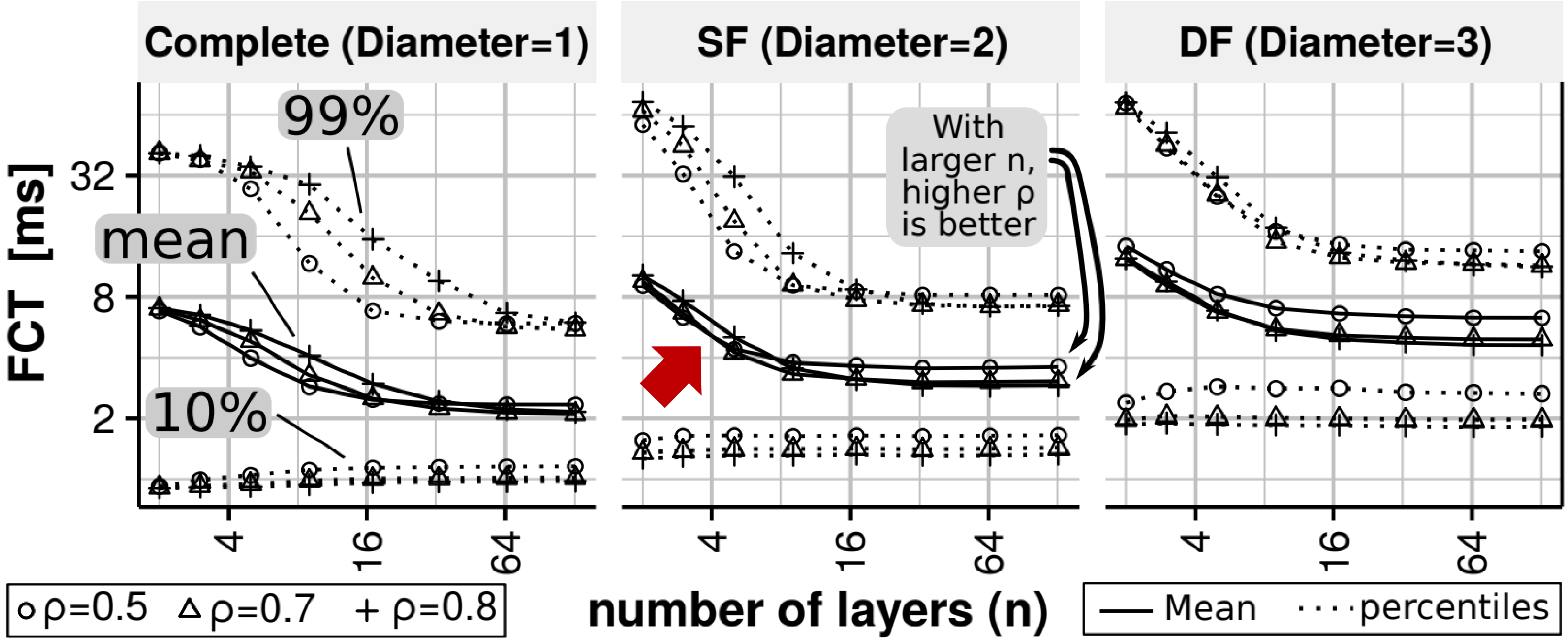


rho: fraction of links kept in a layer

EVALUATION

What layer setup fares best?

N ≈ 10,000; comparable cost; random uniform traffic; „bare Ethernet"



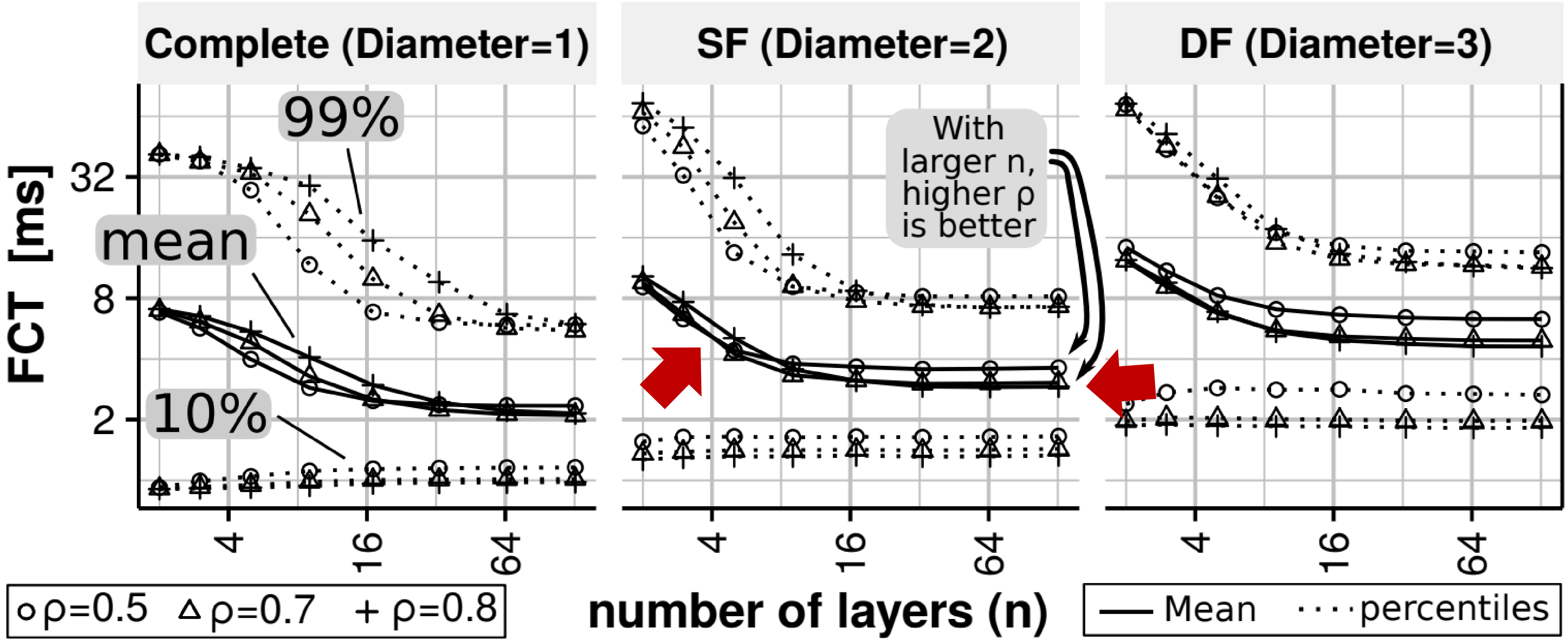
With fewer layers, lower rho is better (sparser layers)

rho: fraction of links kept in a layer

EVALUATION

What layer setup fares best?

N ≈ 10,000; comparable cost; random uniform traffic; „bare Ethernet"



rho: fraction of links kept in a layer

With fewer layers, lower rho is better (sparser layers)

With more layers, higher rho is better (denser layers)

ARE THERE ENOUGH MULTIPLE PATHS IN LOW-DIAMETER TOPOLOGIES?

Part 2

Non- shortest paths

ARE THERE ENOUGH MULTIPLE PATHS IN LOW-DIAMETER TOPOLOGIES?

Part 2

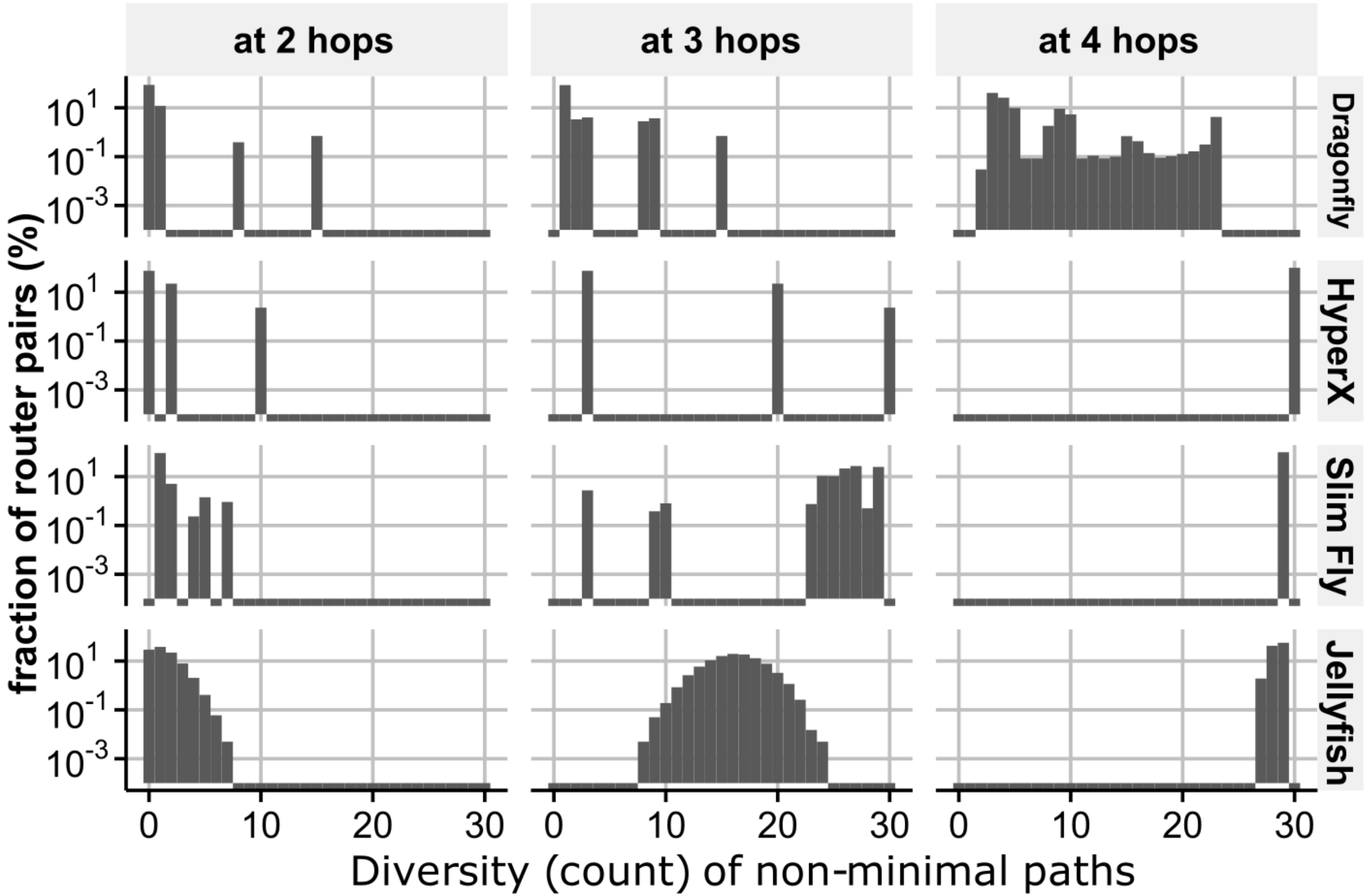
Non- shortest paths

How about non-shortest paths?

ARE THERE ENOUGH MULTIPLE PATHS IN LOW-DIAMETER TOPOLOGIES?

Part 2

Non- shortest paths

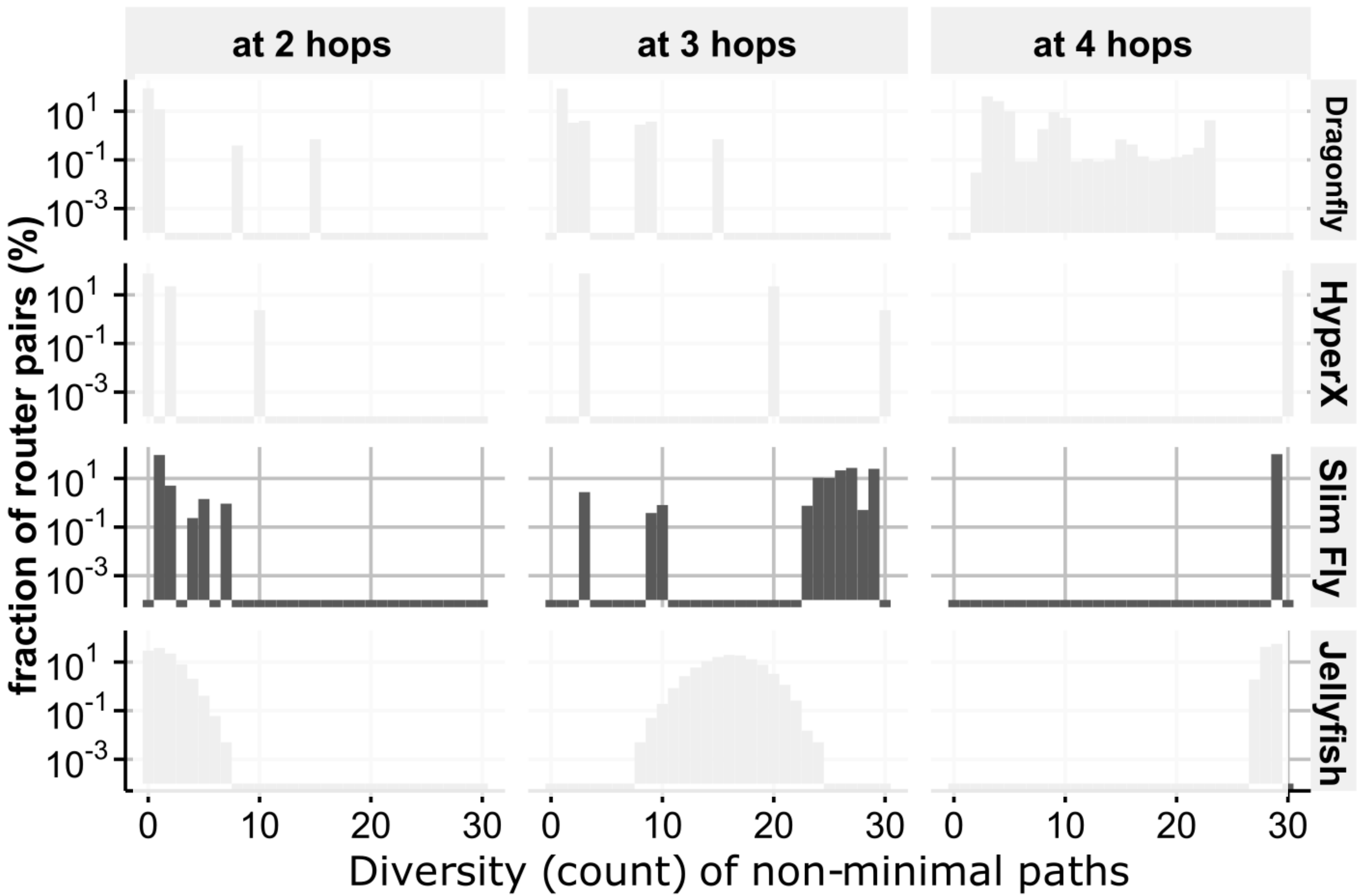


How about non-shortest paths?

ARE THERE ENOUGH MULTIPLE PATHS IN LOW-DIAMETER TOPOLOGIES?

Part 2

Non- shortest paths



How about non-shortest paths?

Are there enough multiple paths in low-diameter topologies?

Part 2

Non- shortest paths

How about non-shortest paths?

ARE THERE ENOUGH MULTIPLE PATHS IN LOW-DIAMETER TOPOLOGIES?

Part 2

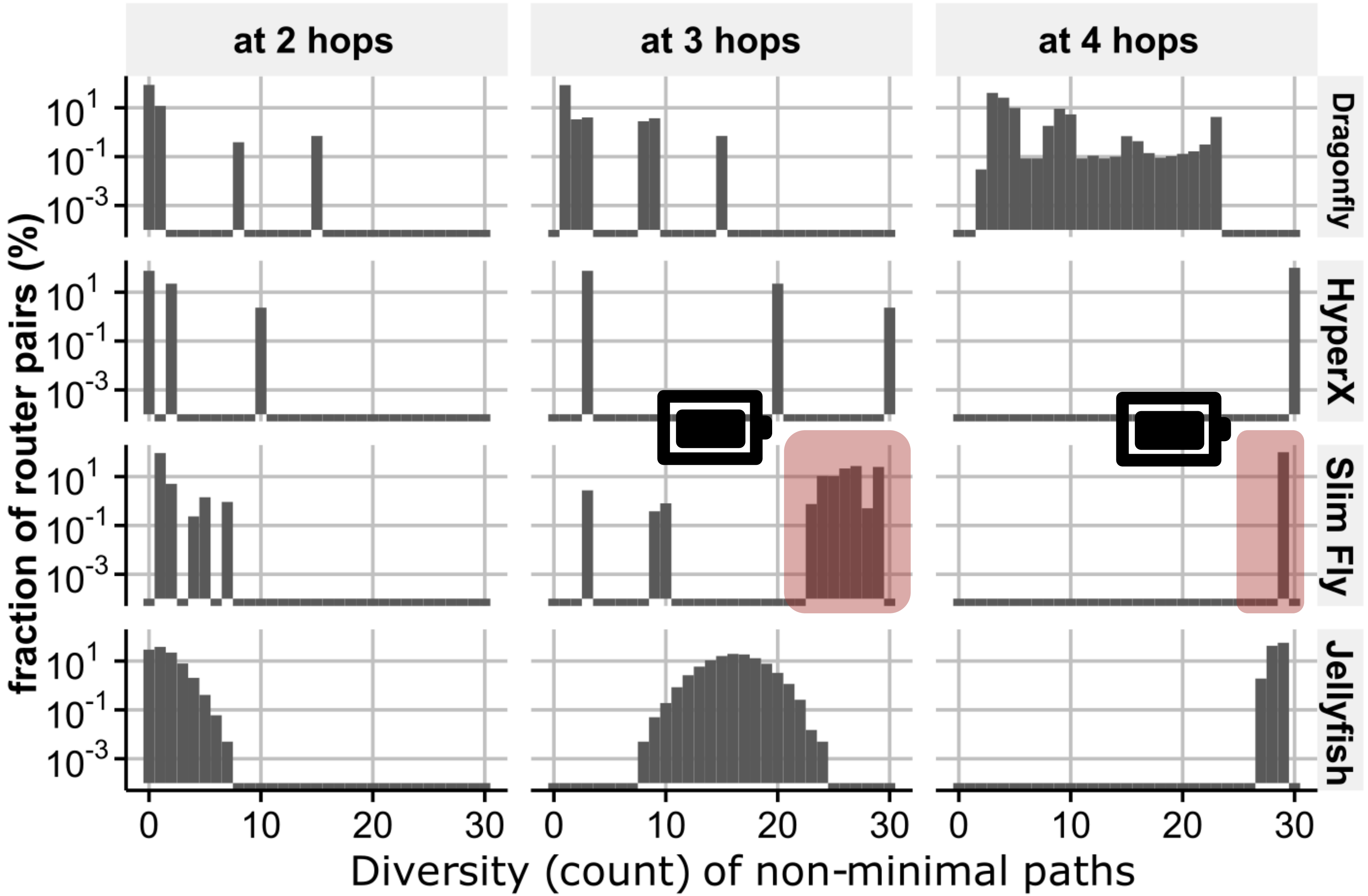
Non- shortest paths

How about non-shortest paths?

ARE THERE ENOUGH MULTIPLE PATHS IN LOW-DIAMETER TOPOLOGIES?

Part 2

Non- shortest paths

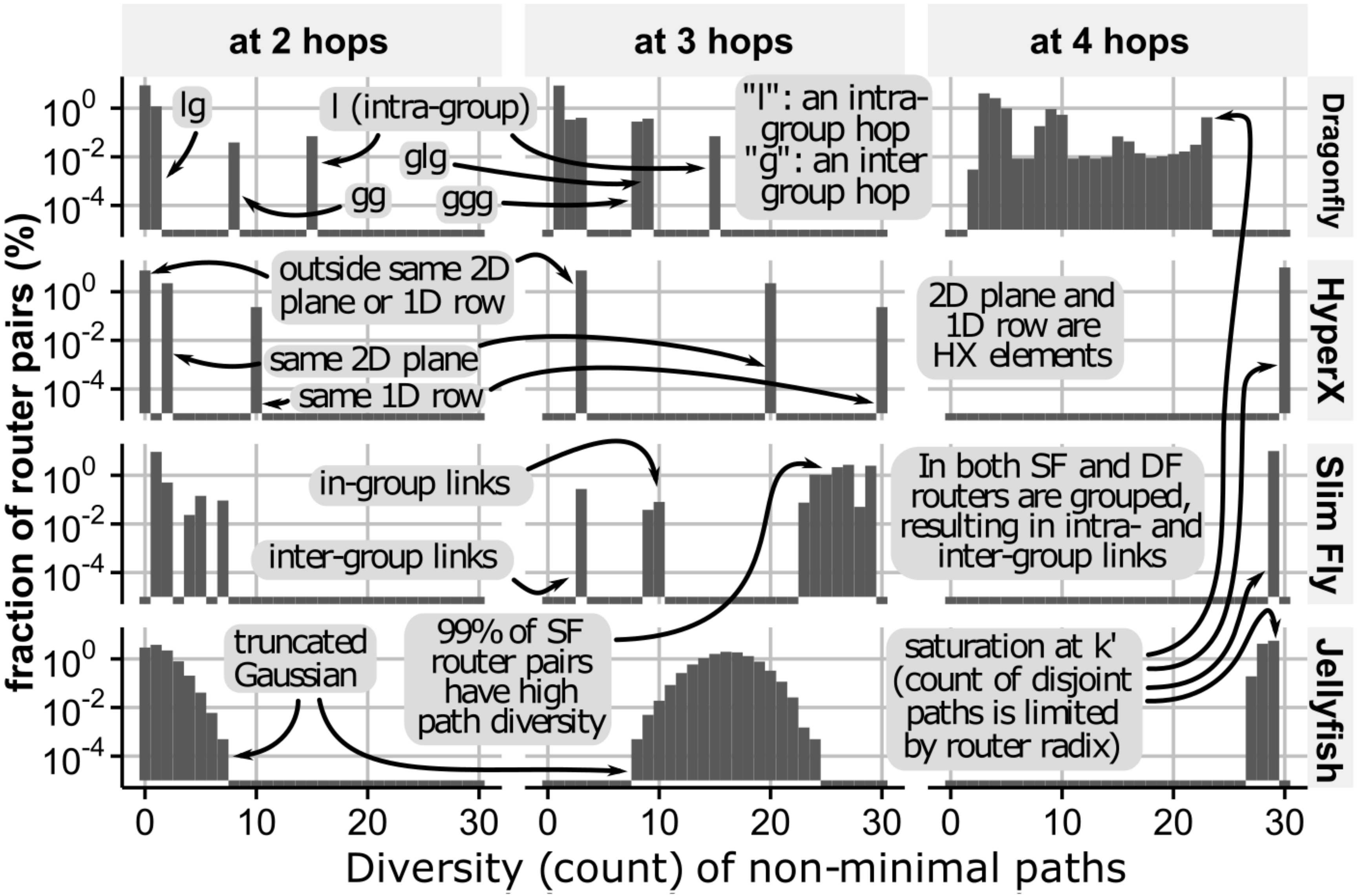


How about non-shortest paths?

ARE THERE ENOUGH MULTIPLE PATHS IN LOW-DIAMETER TOPOLOGIES?

Part 2

Non- shortest paths

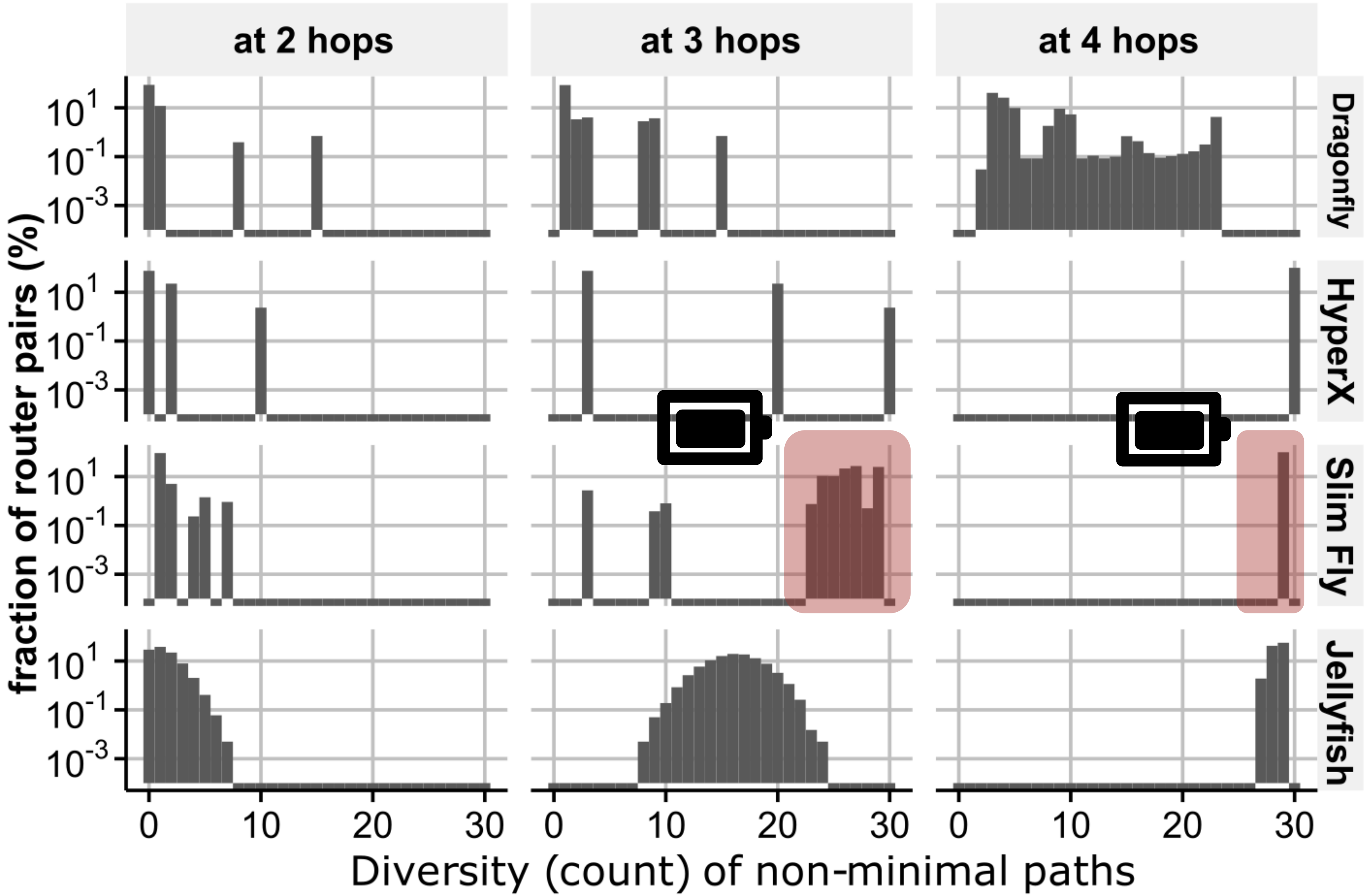


How about non-shortest paths?

Are there enough multiple paths in low-diameter topologies?

Part 2

Non- shortest paths

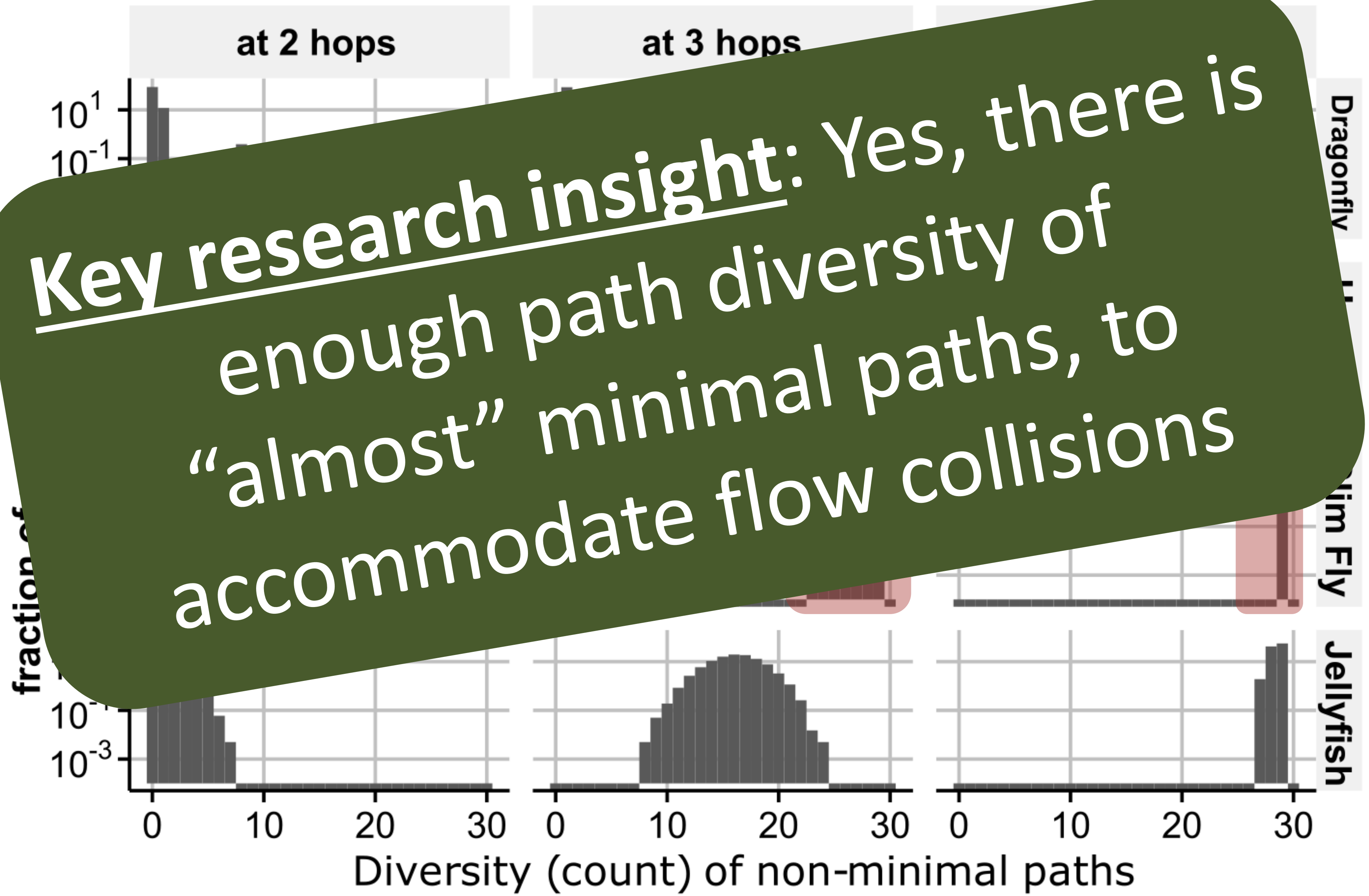


How about non-shortest paths?

ARE THERE ENOUGH MULTIPLE PATHS IN LOW-DIAMETER TOPOLOGIES?

Part 2

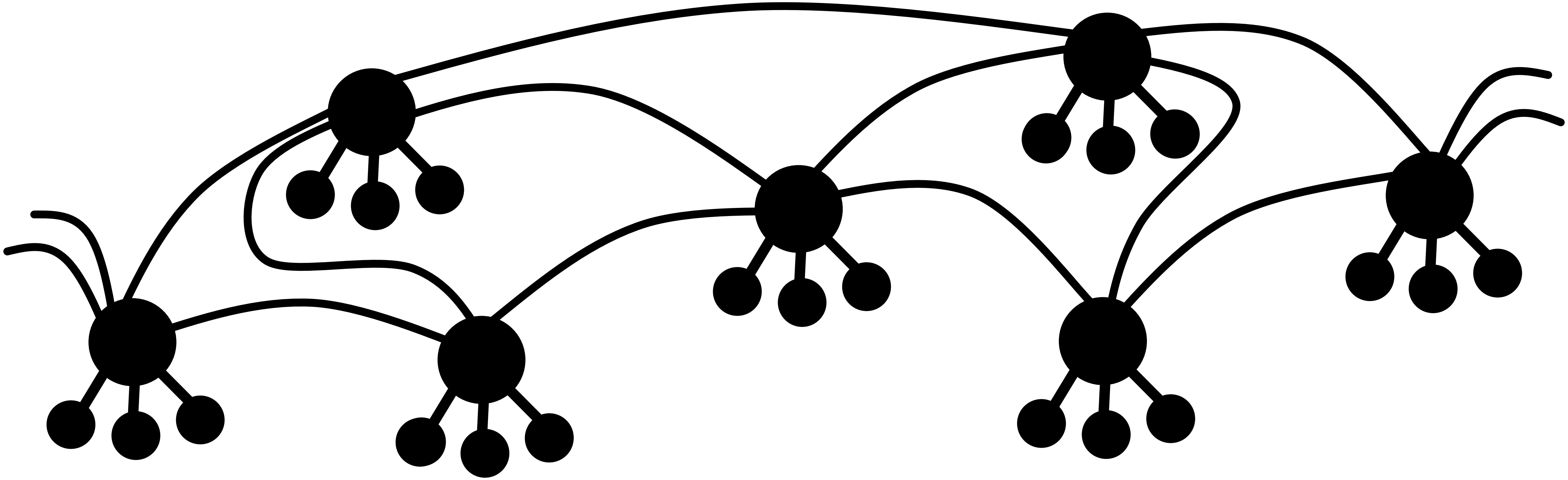
Non- shortest paths



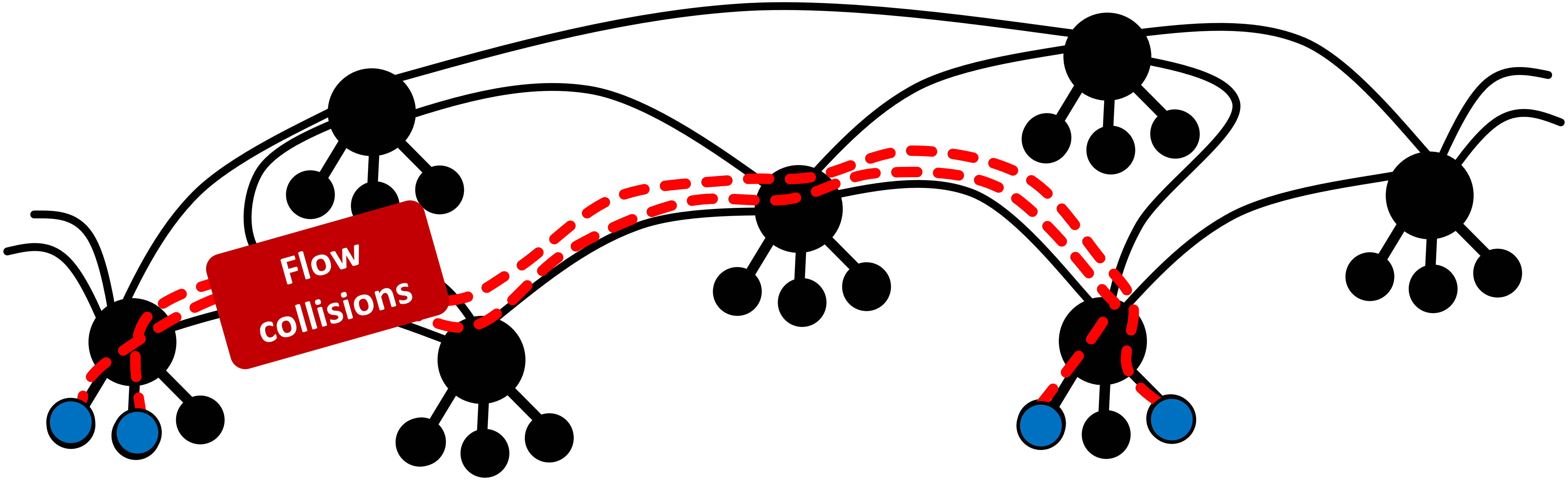
Key research insight: Yes, there is enough path diversity of "almost" minimal paths, to accommodate flow collisions

How about non-shortest paths?

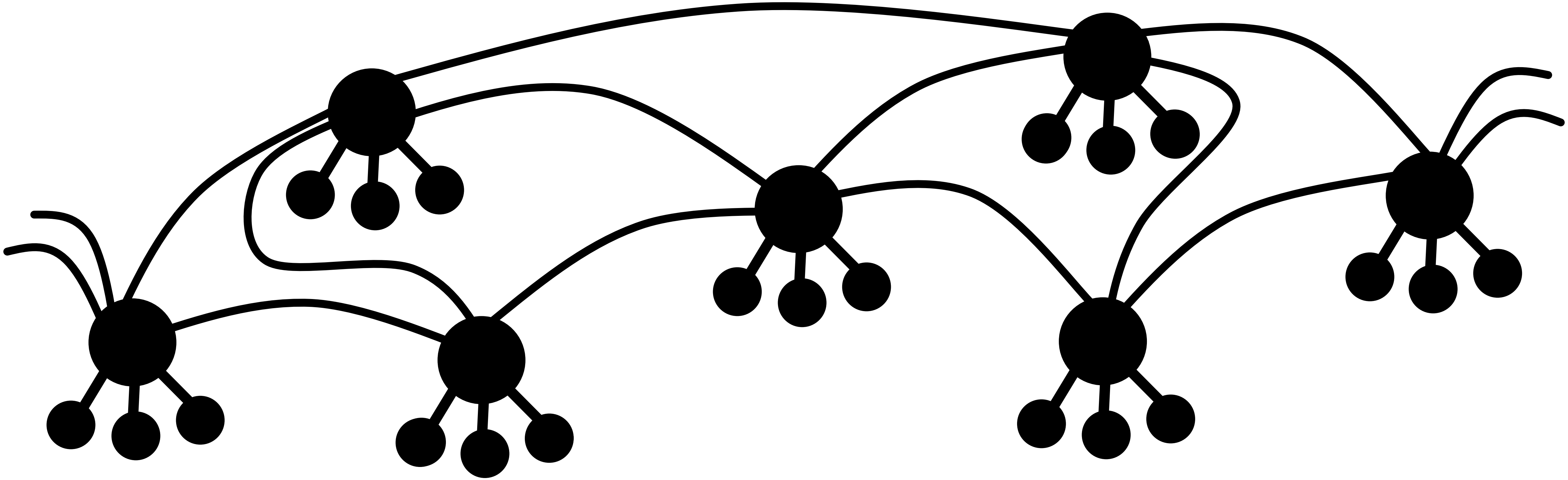
There is much more...



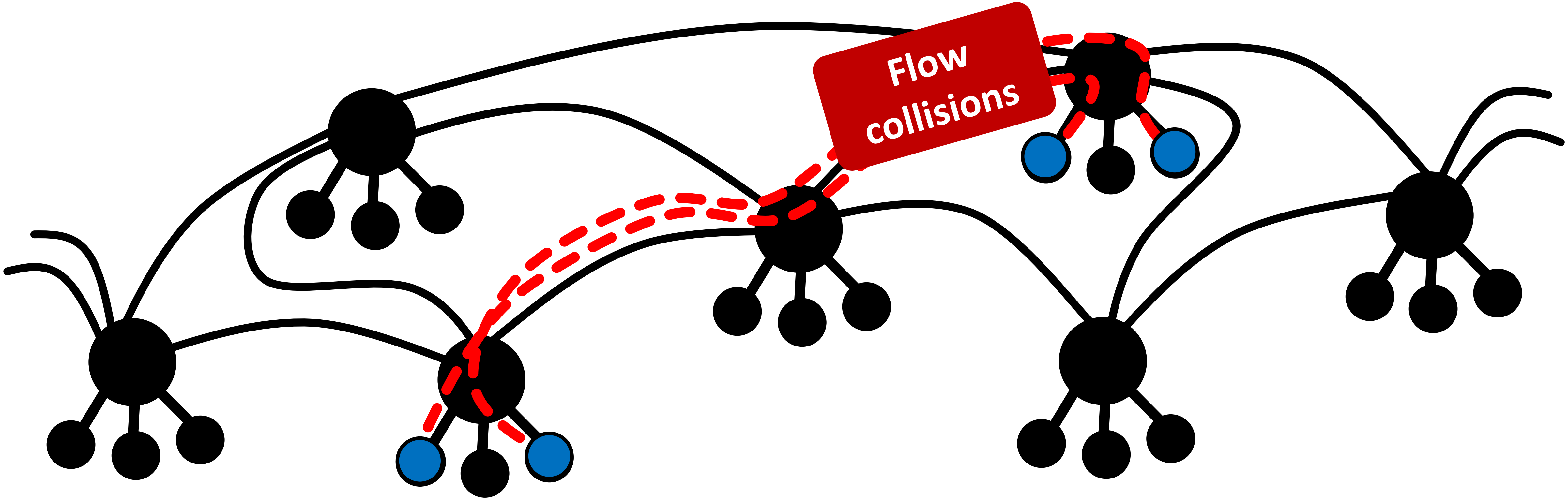
THERE IS MUCH MORE...



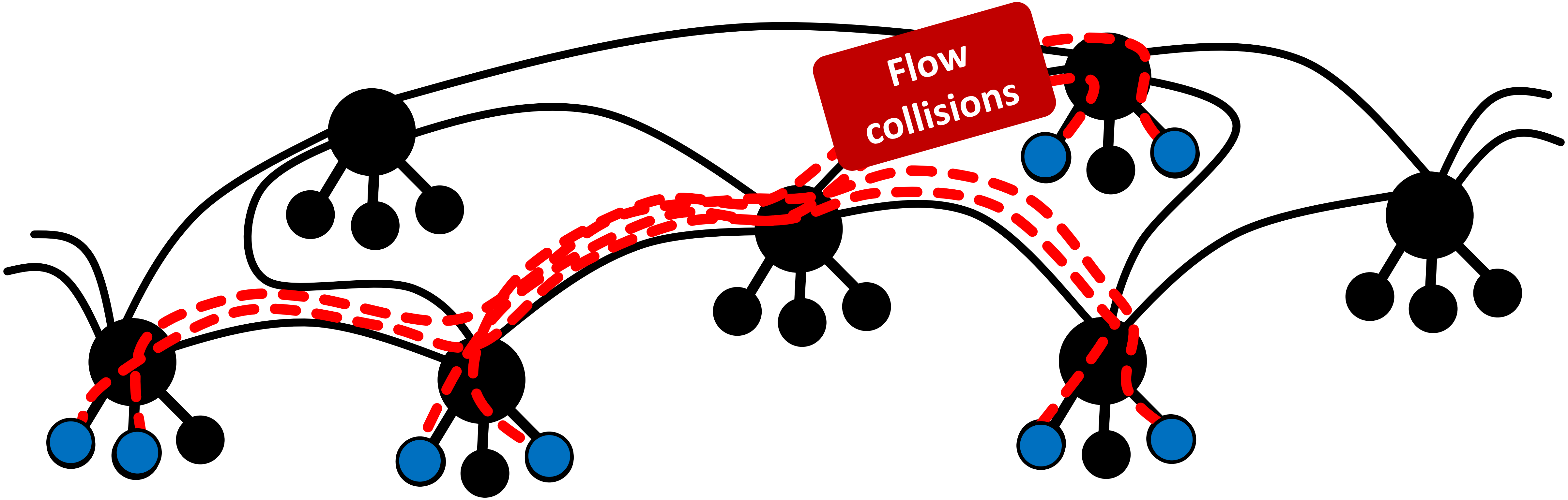
There is much more...



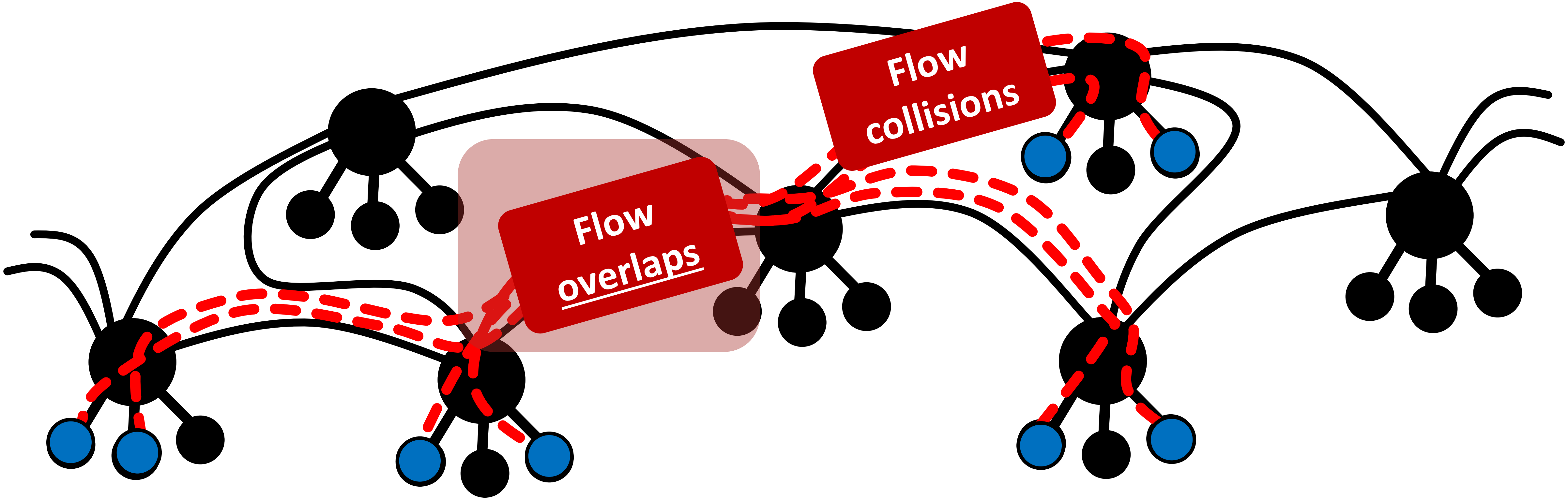
THERE IS MUCH MORE...



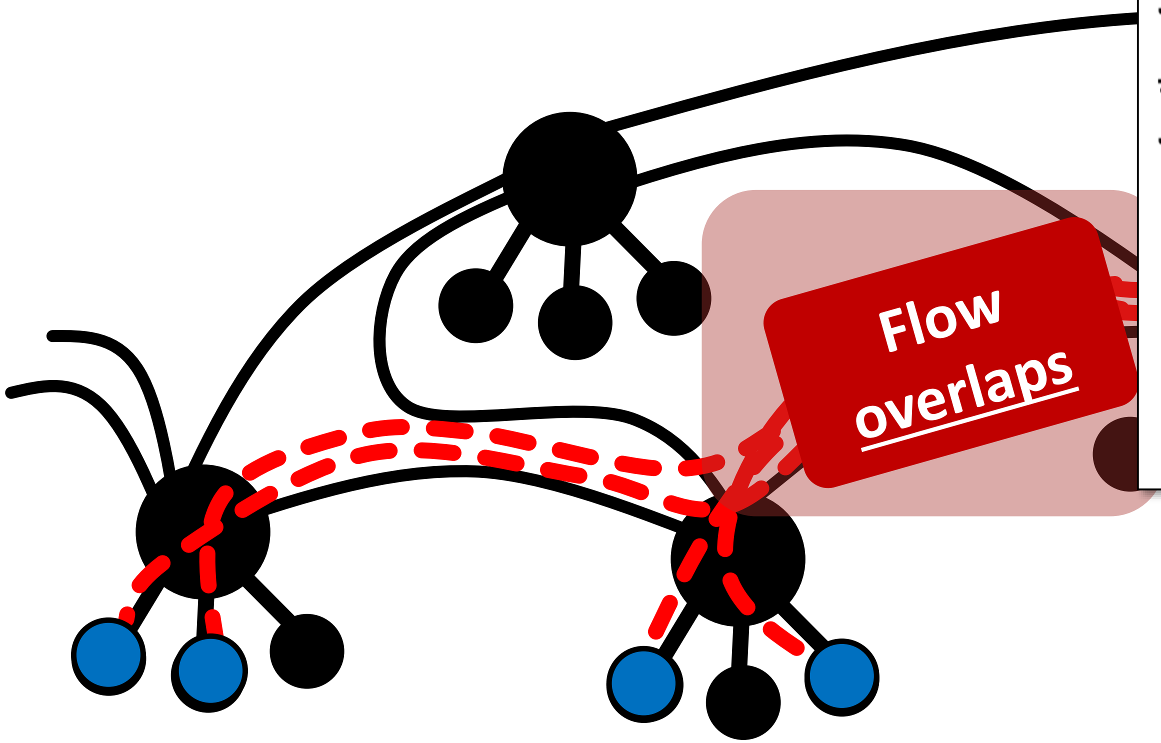
THERE IS MUCH MORE...

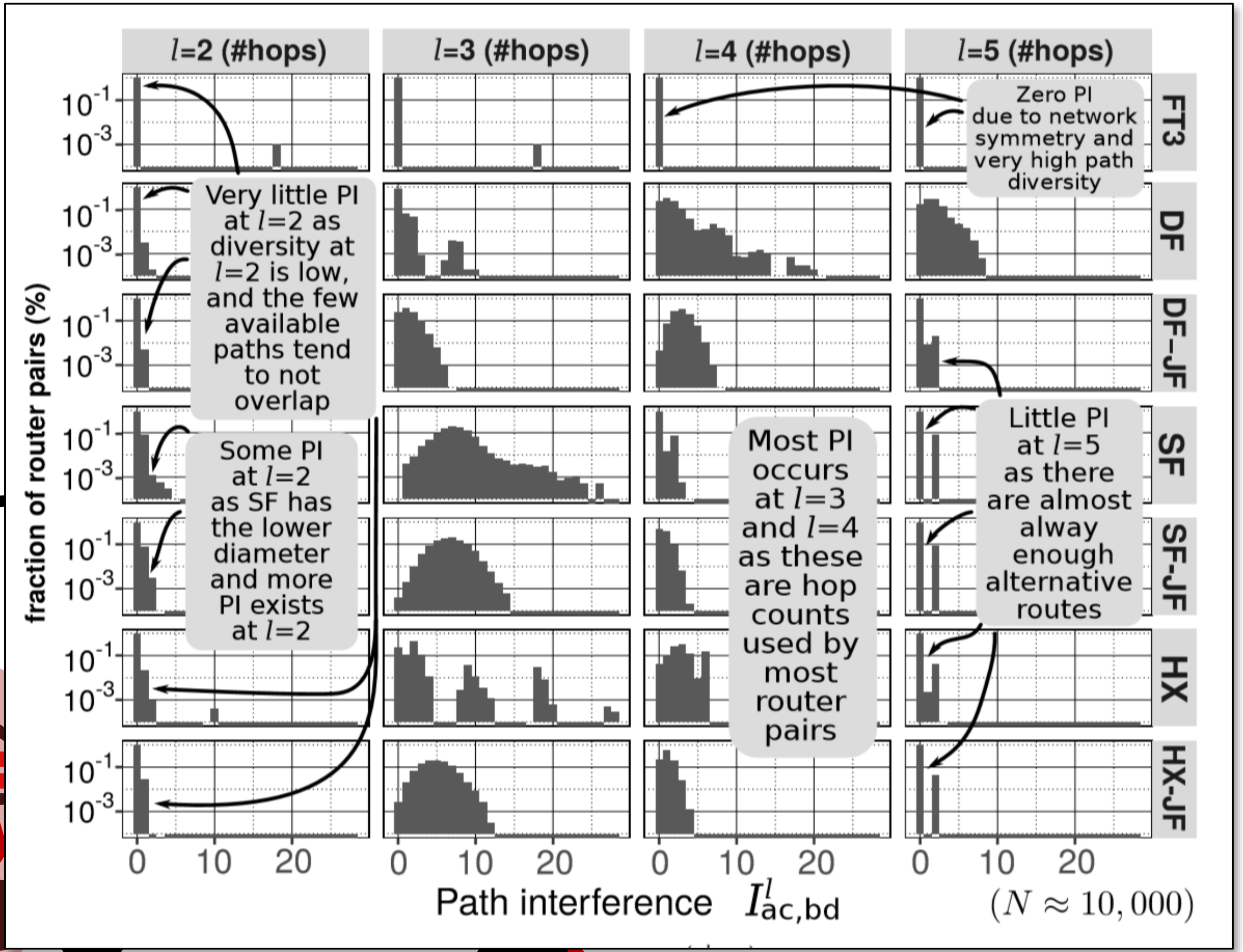


THERE IS MUCH MORE...

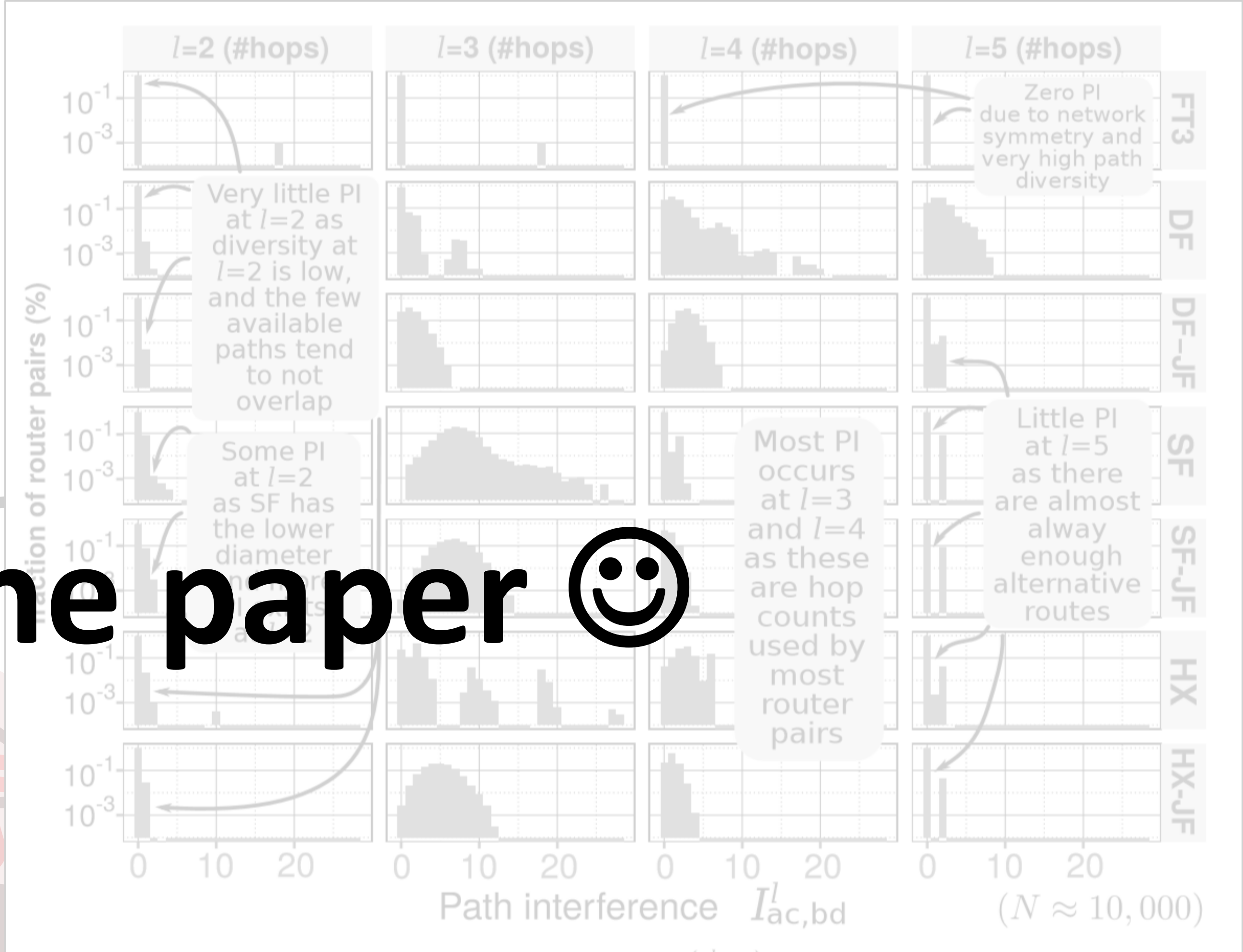


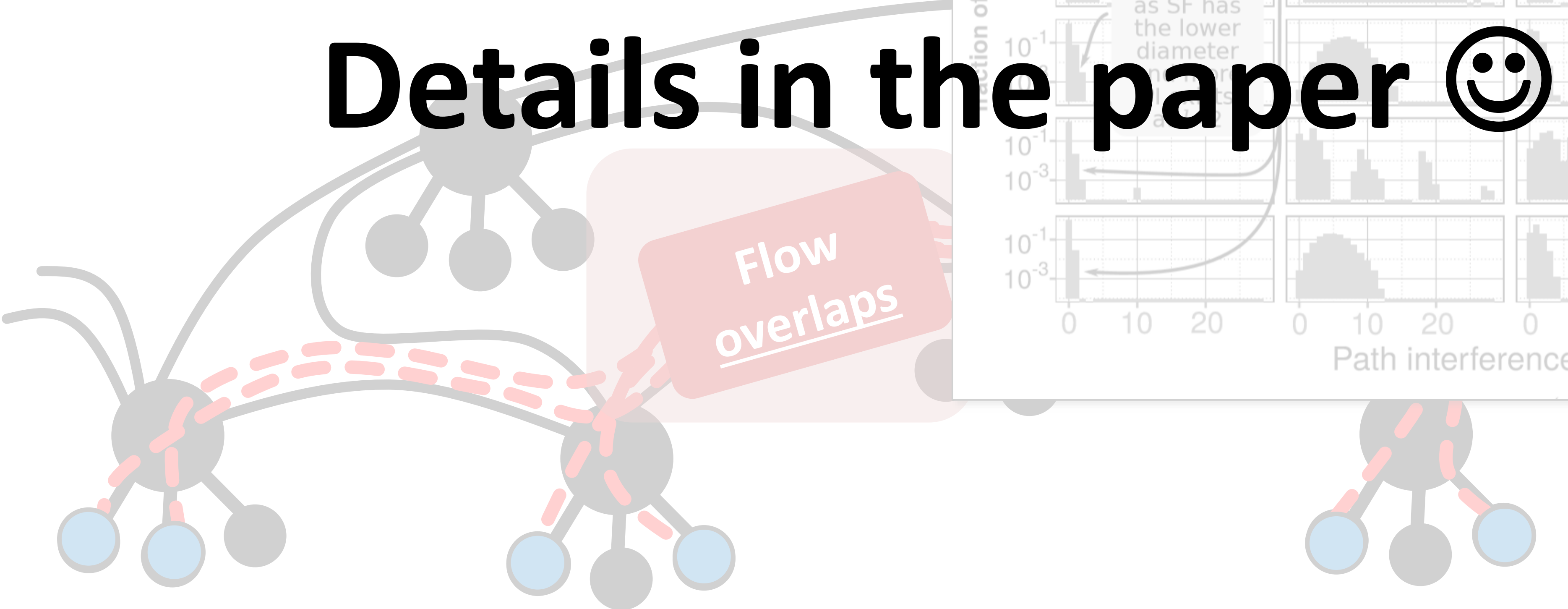
THERE IS MUCH MORE…





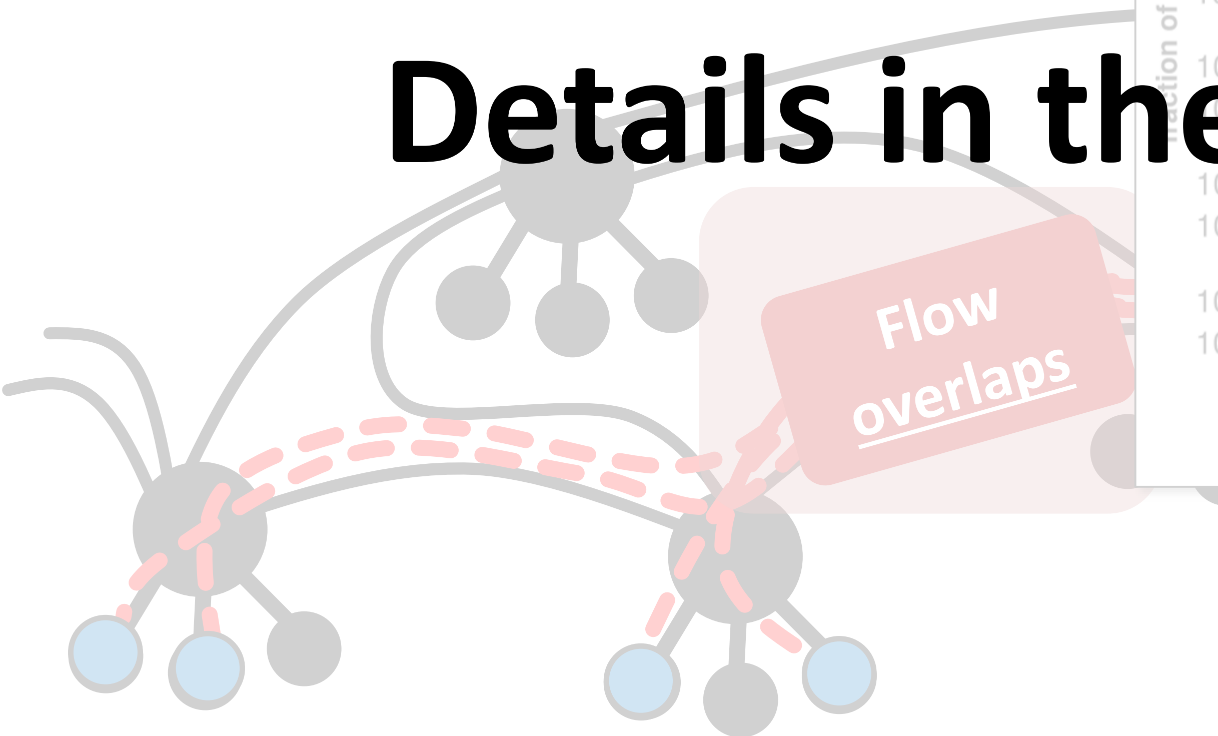
THERE IS MUCH MORE…





Details in the paper ☺

THERE IS MUCH MORE…



Details in the paper ☺

Let's sum up…

FatPaths architecture: which technology?

FatPaths architecture: which technology?

InfiniBand

FatPaths architecture: which technology?

InfiniBand

Myrinet

FatPaths architecture: which technology?

InfiniBand

Myrinet

Ethernet

FatPaths architecture: which technology?

InfiniBand

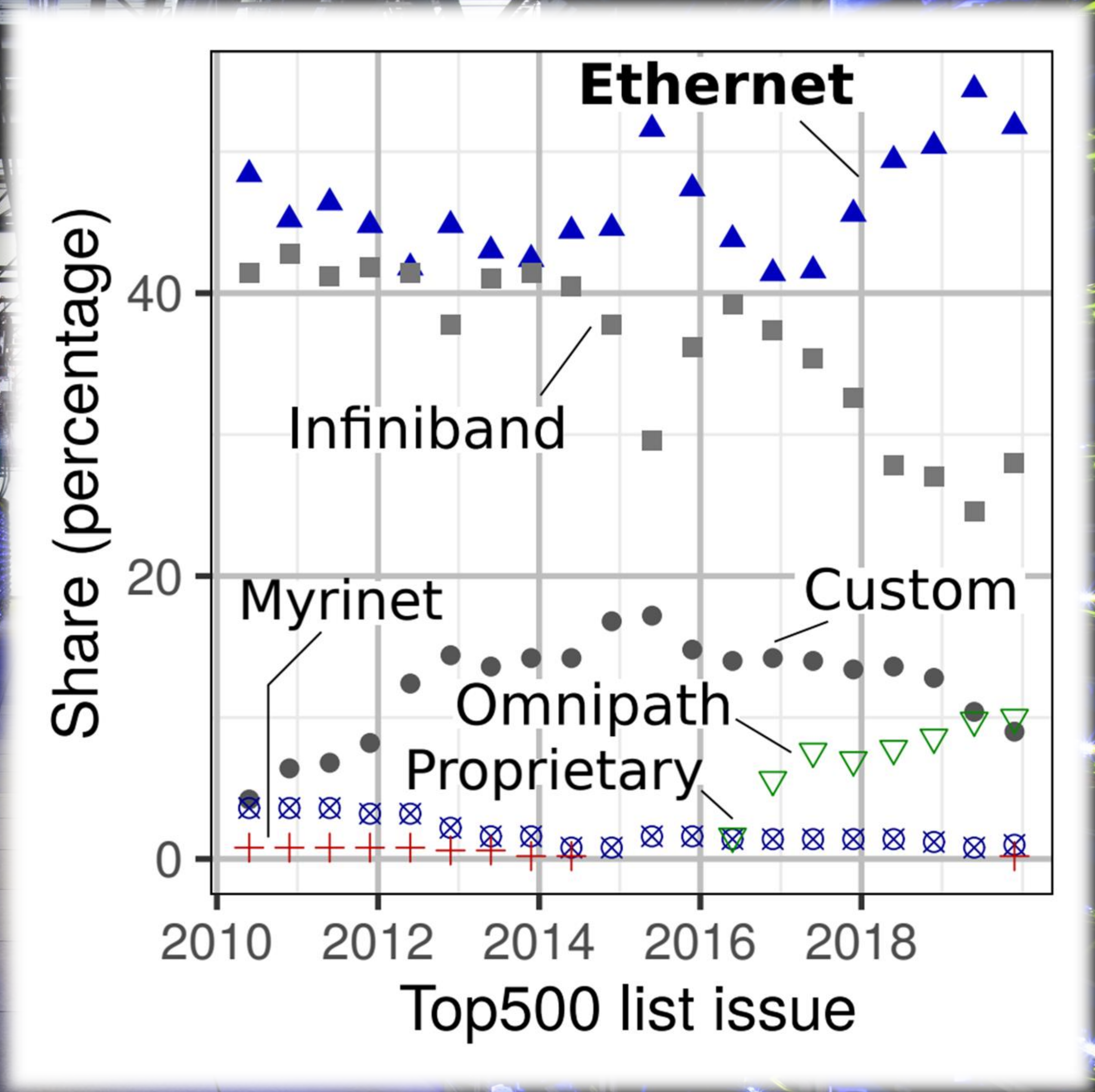
Others

Ethernet

Myrinet

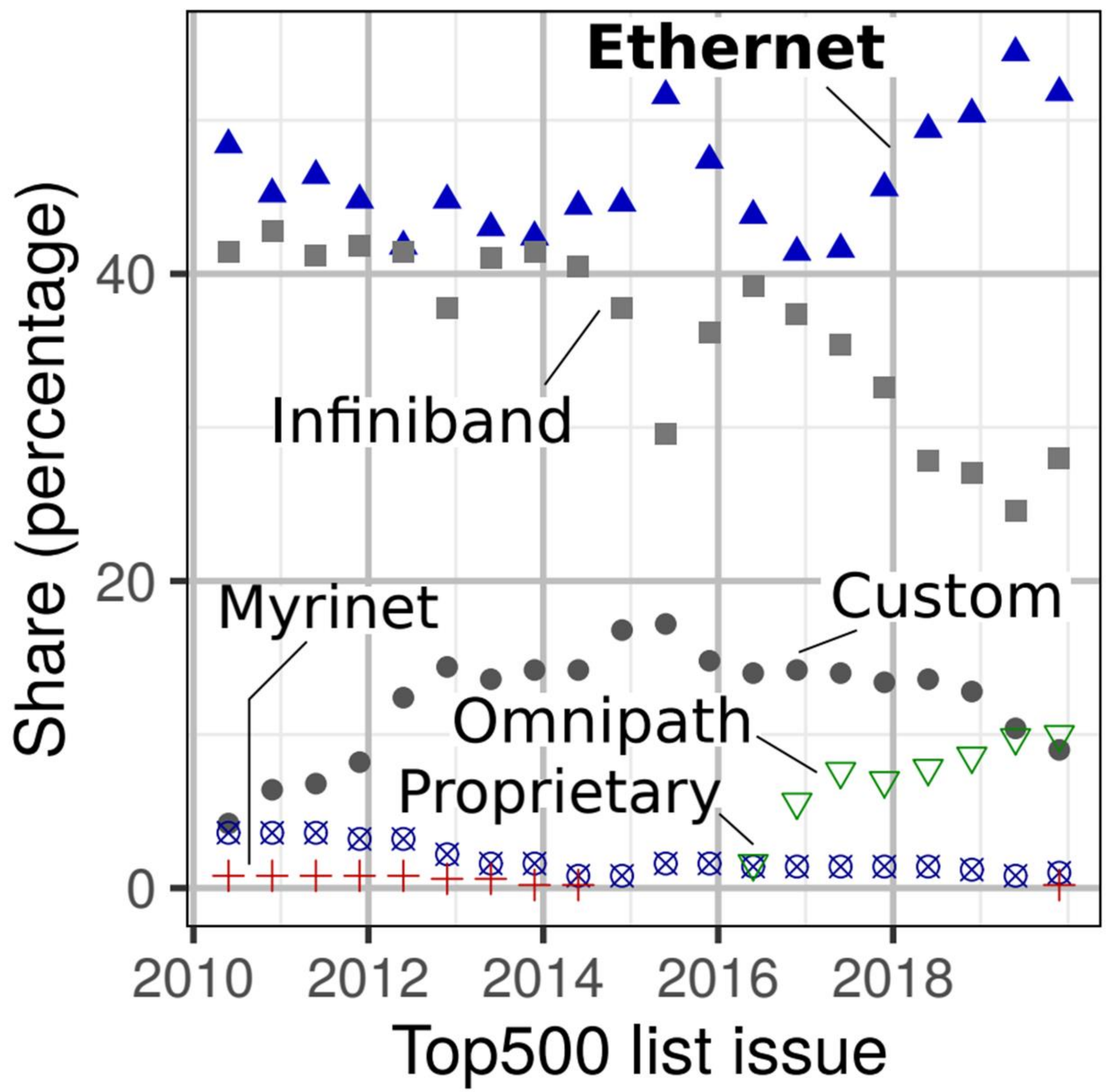
Ethernet & HPC

ETHERNET & HPC



Ethernet & HPC

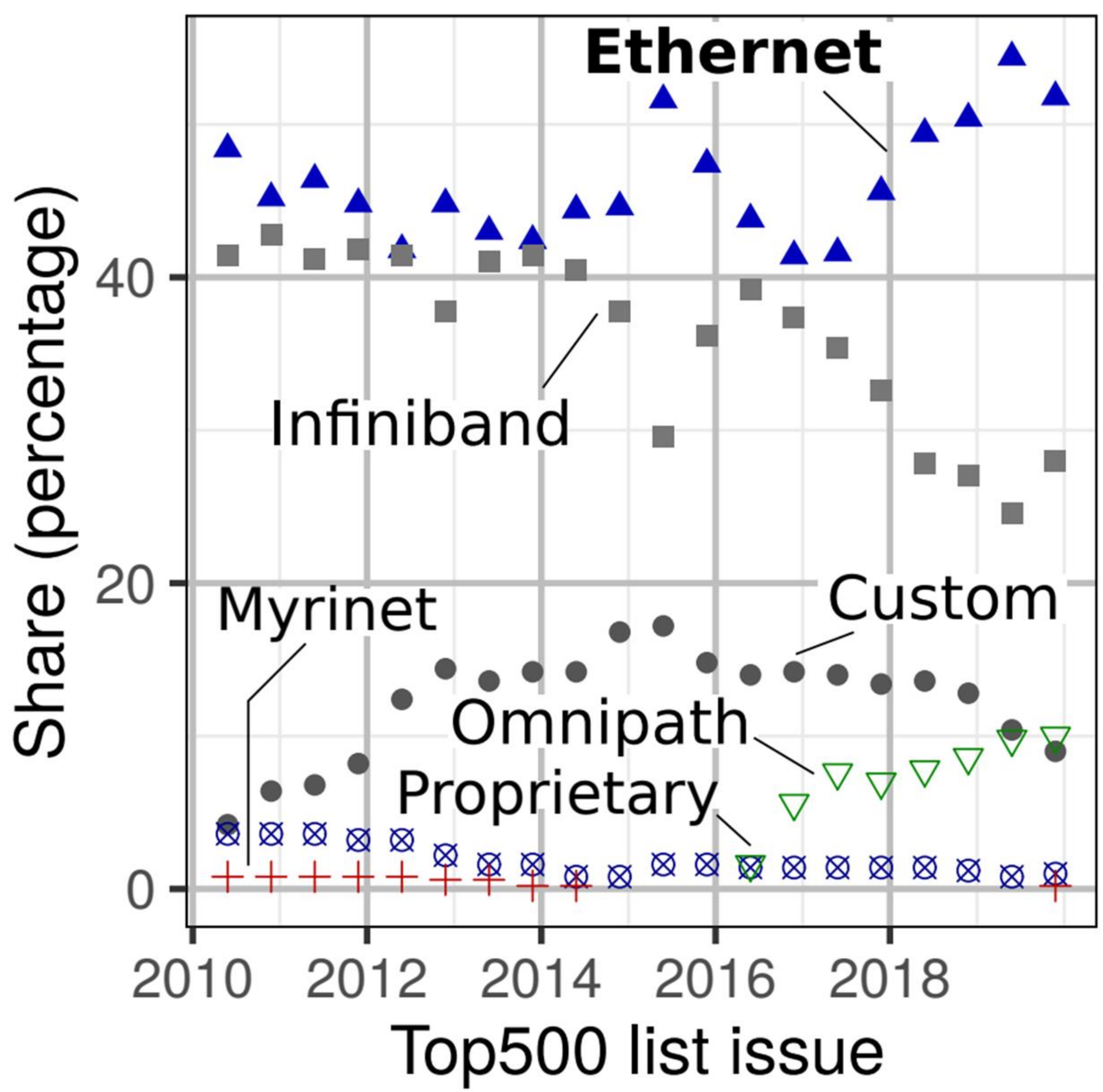
More than 50% of systems in Top500 use Ethernet



Ethernet & HPC

More than 50% of systems in Top500 use Ethernet

However, systems based on Ethernet are not as efficient as others: ≈ (details in the paper)

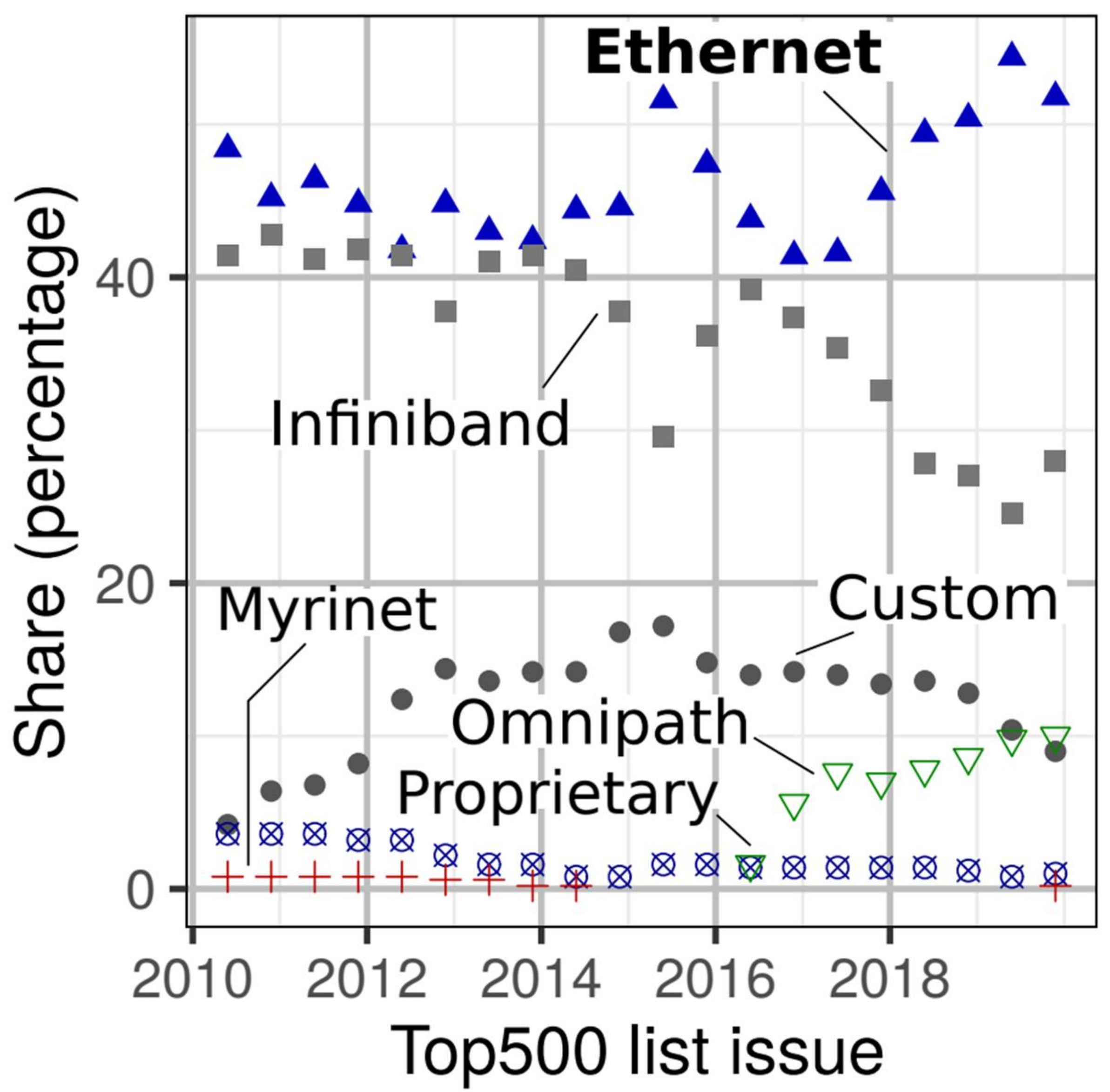


Ethernet & HPC

More than 50% of systems in Top500 use Ethernet

However, systems based on Ethernet are not as efficient as others: ≈ (details in the paper)

LINPACK Efficiency:
≈51% for 100G Ethernet
≈65% for 100G IB HDR
(details in the paper)

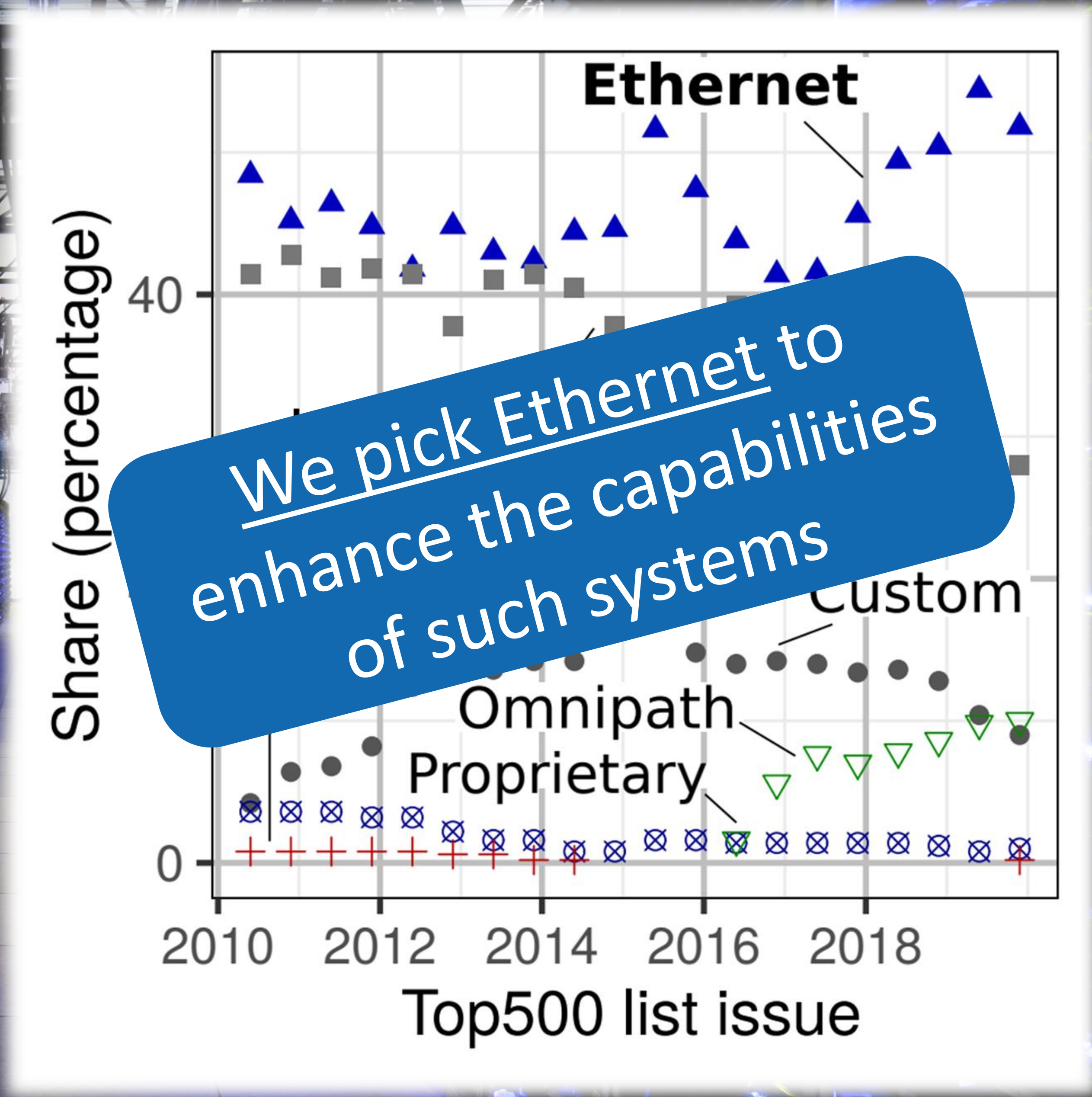


Ethernet & HPC

More than 50% of systems in Top500 use Ethernet

However, systems based on Ethernet are not as efficient as others: ≈ (details in the paper)

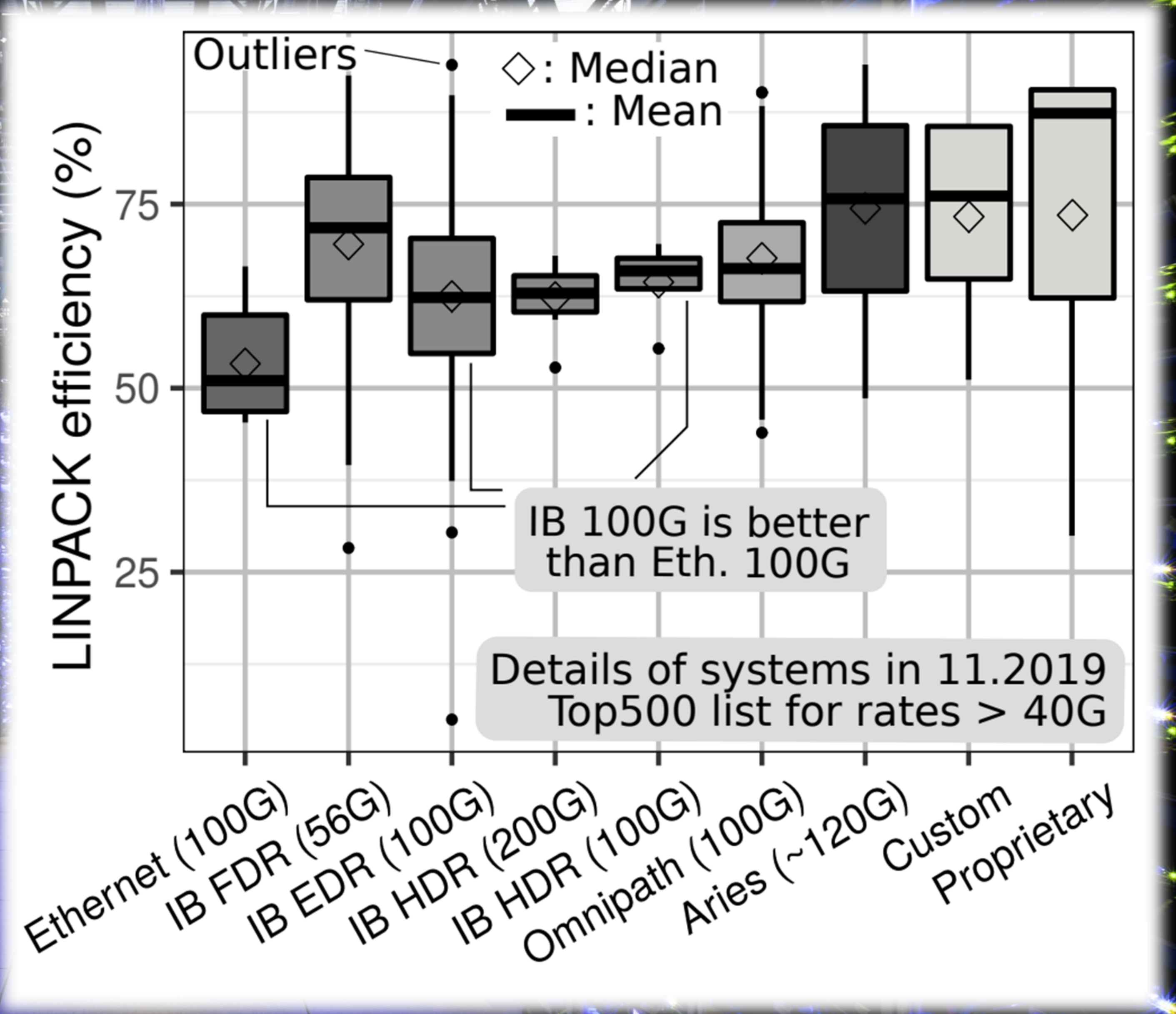
LINPACK Efficiency:
≈51% for 100G Ethernet
≈65% for 100G IB HDR
(details in the paper)



We pick Ethernet to enhance the capabilities of such systems

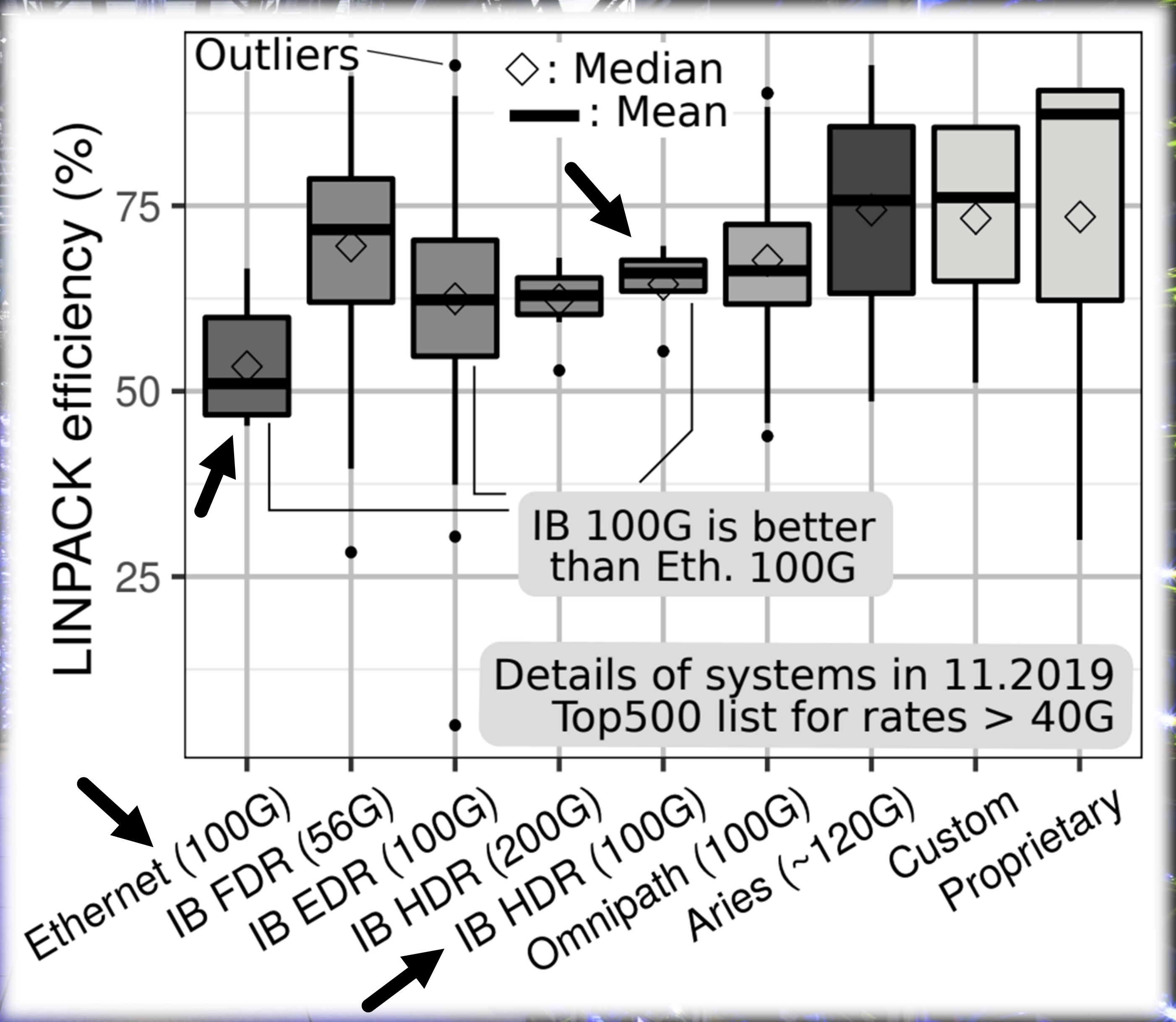
Ethernet & HPC

Ethernet & HPC



Ethernet & HPC

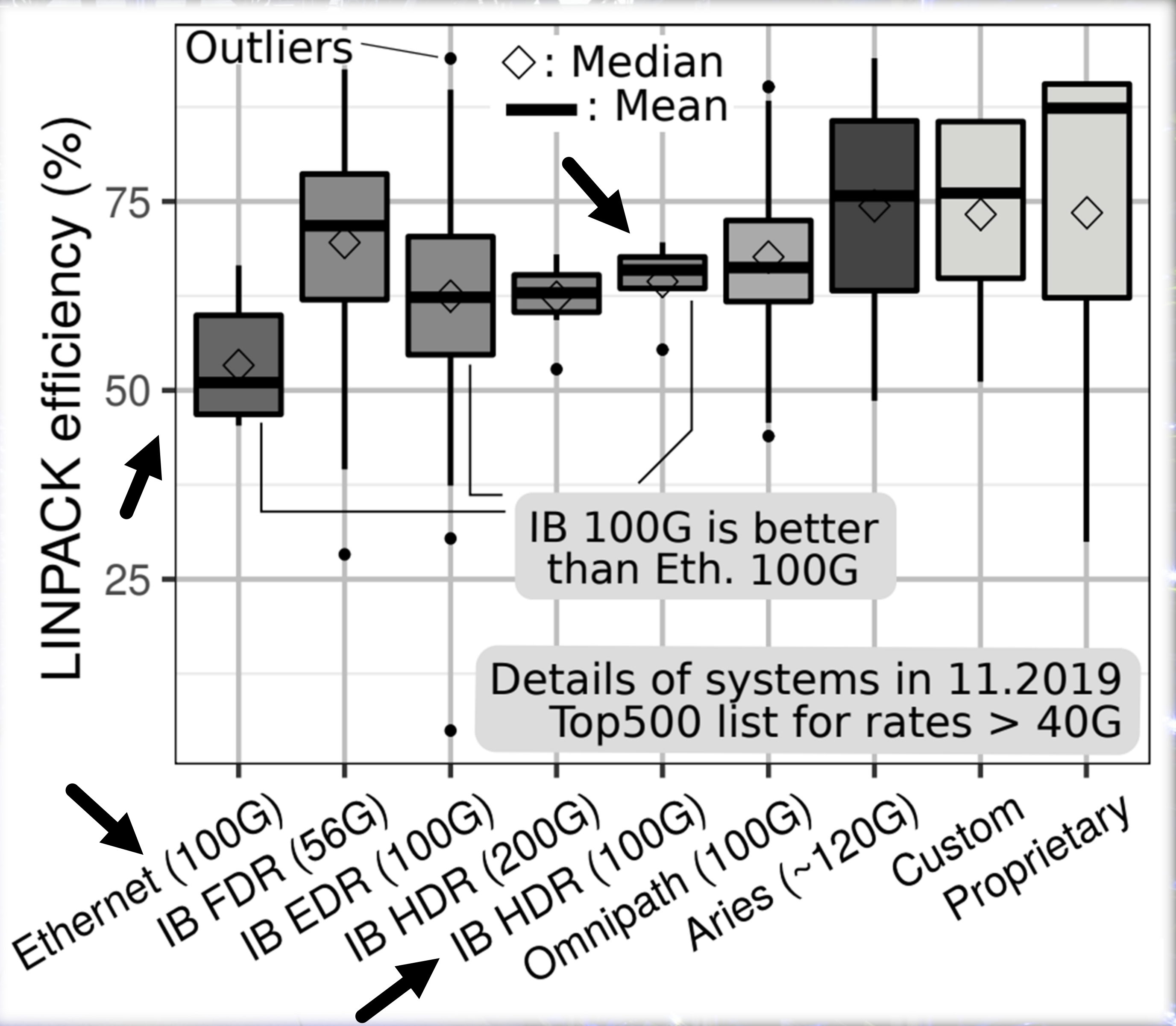
However, systems based on Ethernet are not as efficient as others



Ethernet & HPC

However, systems based on Ethernet are not as efficient as others

<u>We pick Ethernet</u> to enhance the capabilities of such systems



Transport Design

Transport Design

How to maximize performance of the transport layer?

Transport Design

How to maximize performance of the transport layer?

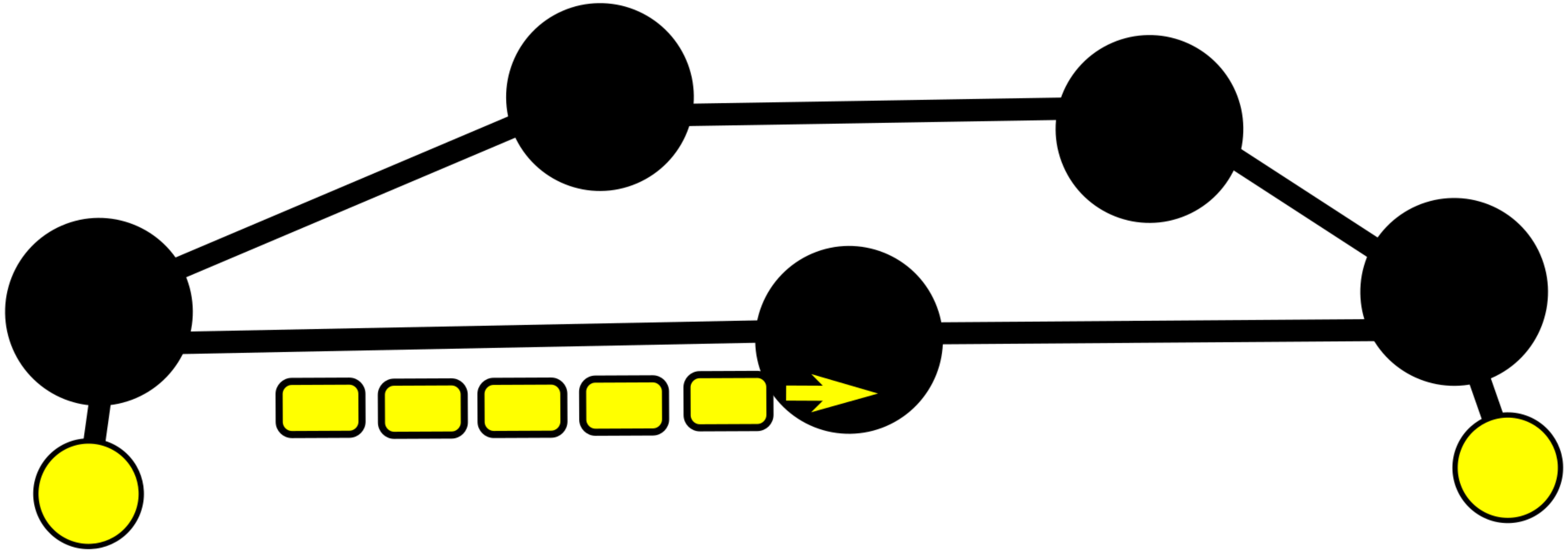
Adapt recent flow control for fat trees [1]

[1] M. Handley et al. Re-architecting datacenter networks and stacks for low latency and high performance. SIGCOMM'17.

Transport Design

How to maximize performance of the transport layer?

Adapt recent flow control for fat trees [1]



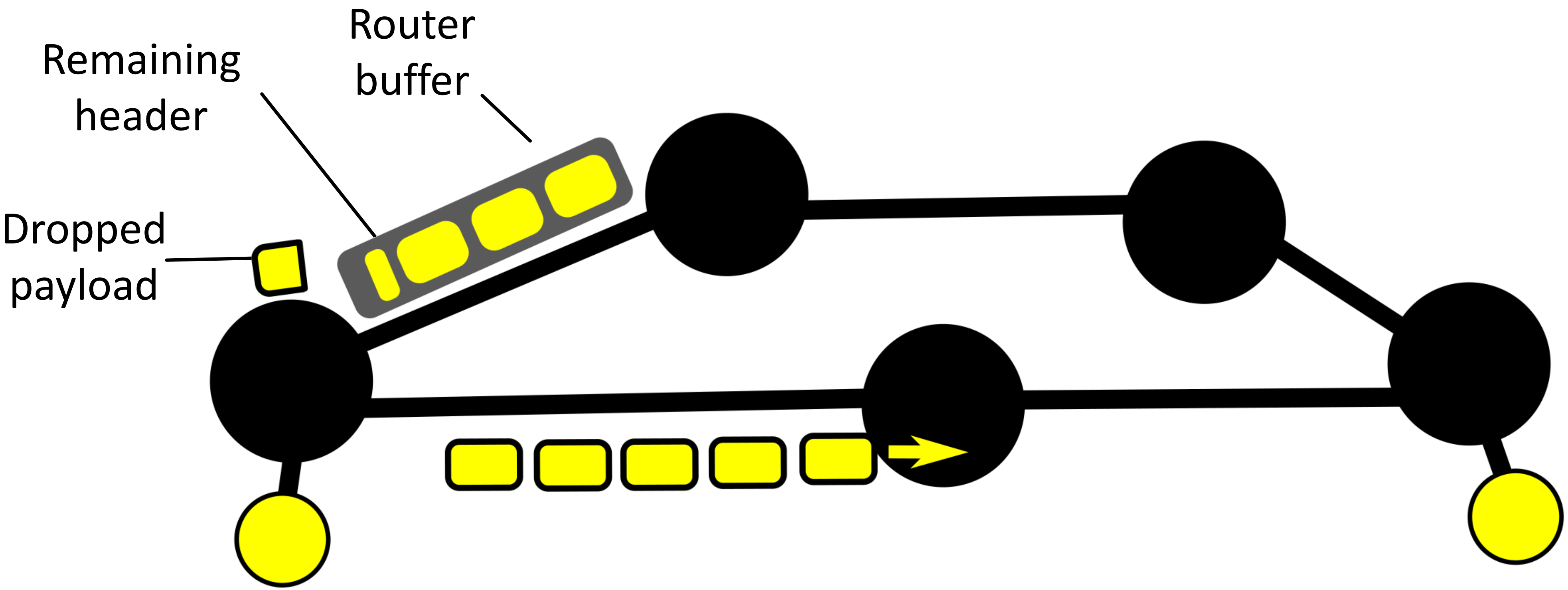
[1] M. Handley et al. Re-architecting datacenter networks and stacks for low latency and high performance. SIGCOMM'17.

Transport Design

How to maximize performance of the transport layer?

Adapt recent flow control for fat trees [1]

Key design choice: Drop only payload if router buffers fill up



[1] M. Handley et al. Re-architecting datacenter networks and stacks for low latency and high performance. SIGCOMM'17.

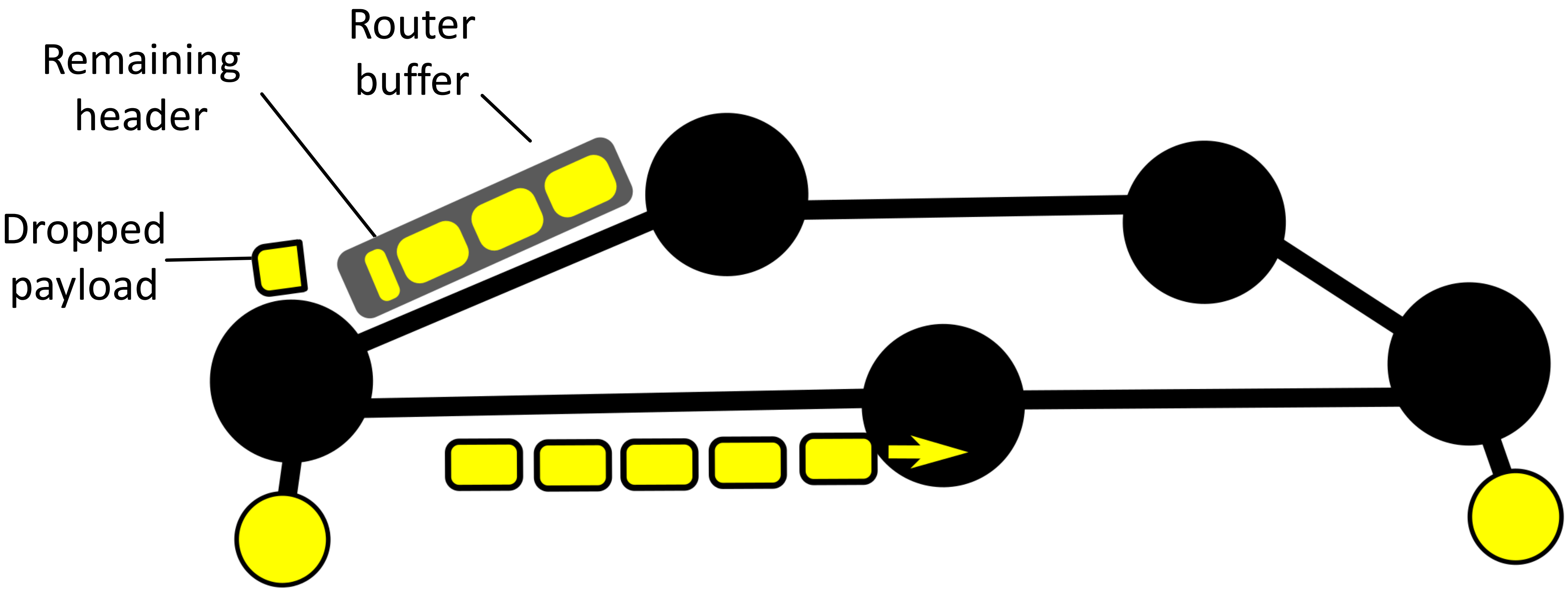
TRANSPORT DESIGN

How to maximize performance of the transport layer?

Adapt recent flow control for fat trees [1]

Prioritize packets with dropped payload and retransmitted packets

Key design choice: Drop only payload if router buffers fill up



[1] M. Handley et al. Re-architecting datacenter networks and stacks for low latency and high performance. SIGCOMM'17.

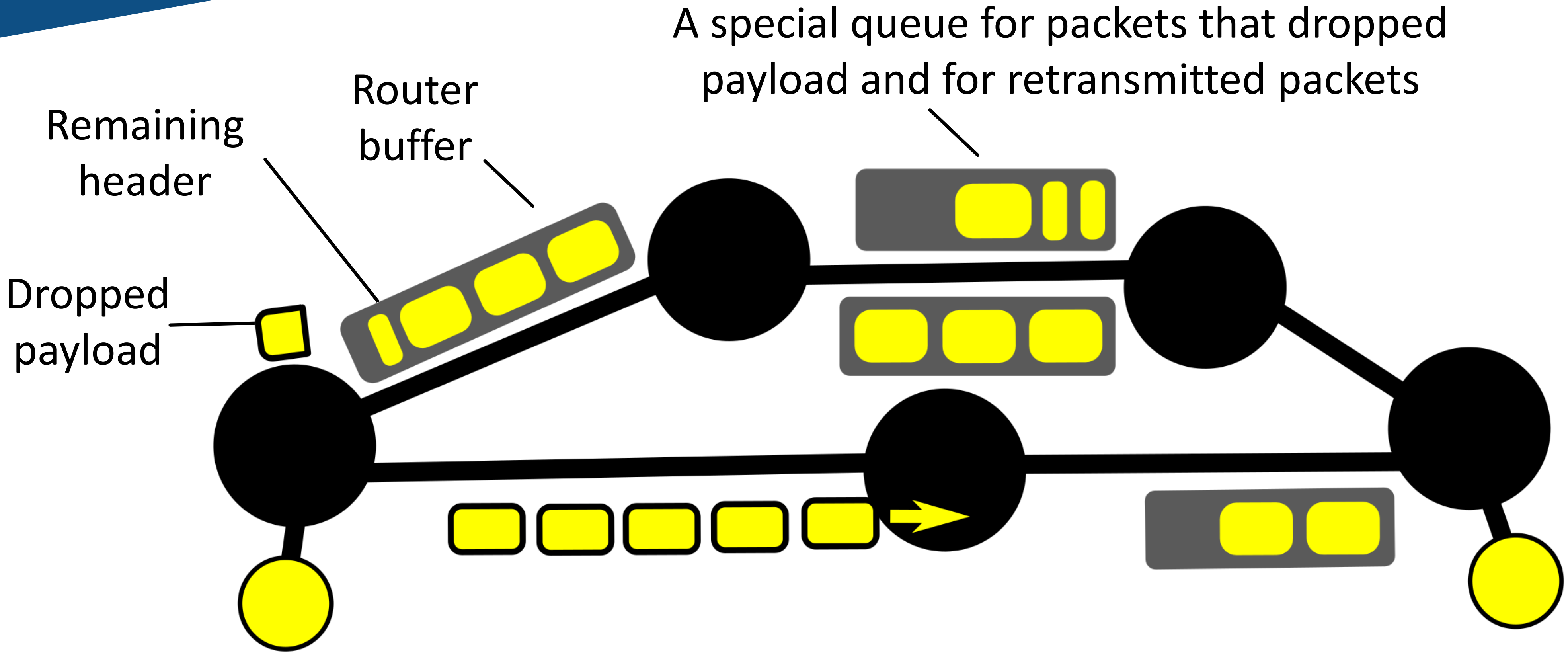
Transport Design

How to maximize performance of the transport layer?

Adapt recent flow control for fat trees [1]

Prioritize packets with dropped payload and retransmitted packets

Key design choice: Drop only payload if router buffers fill up



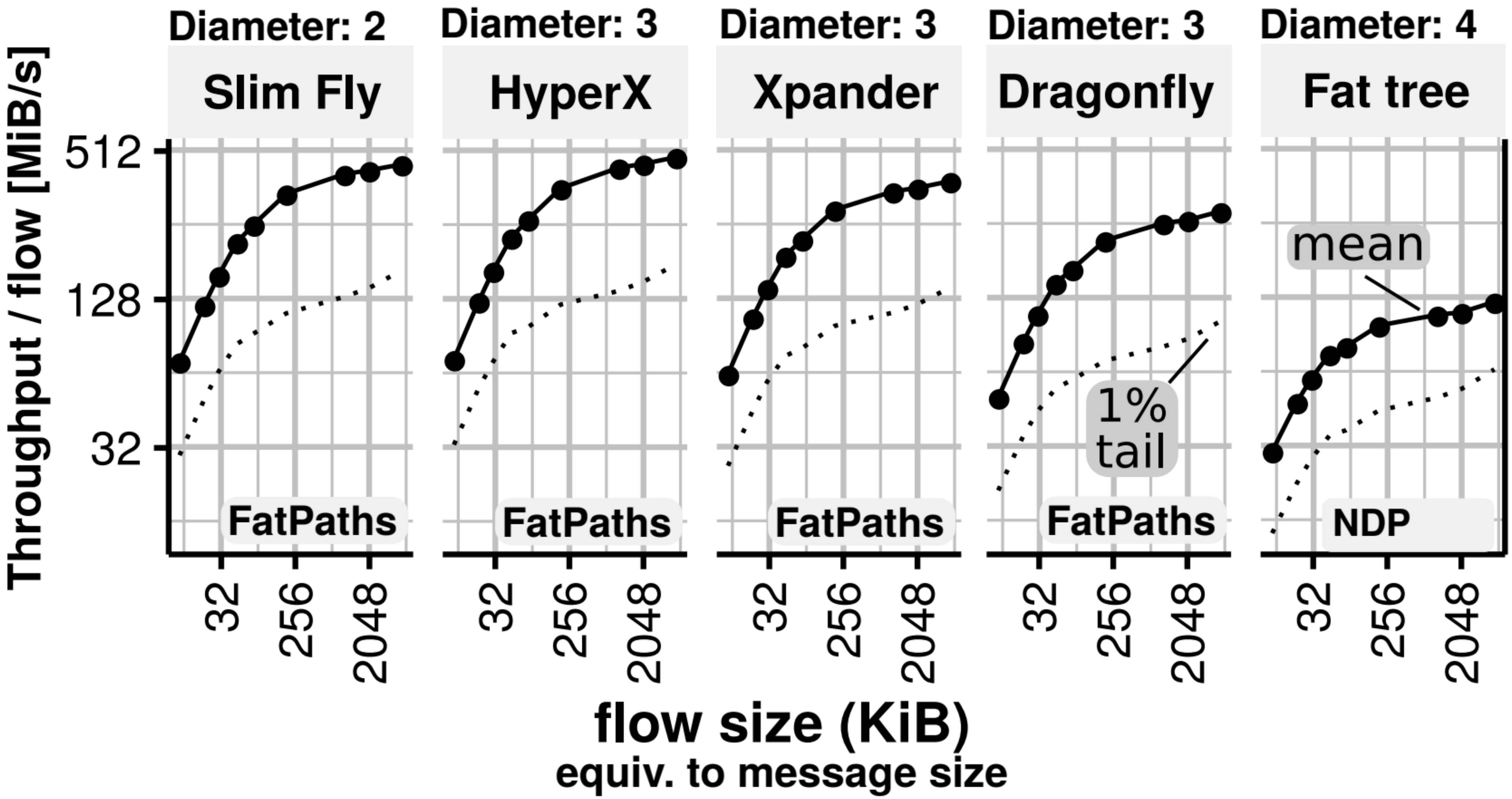
[1] M. Handley et al. Re-architecting datacenter networks and stacks for low latency and high performance. SIGCOMM'17.

EVALUATION

N ≈ 10,000; comparable cost; random uniform traffic; “bare Ethernet” setting

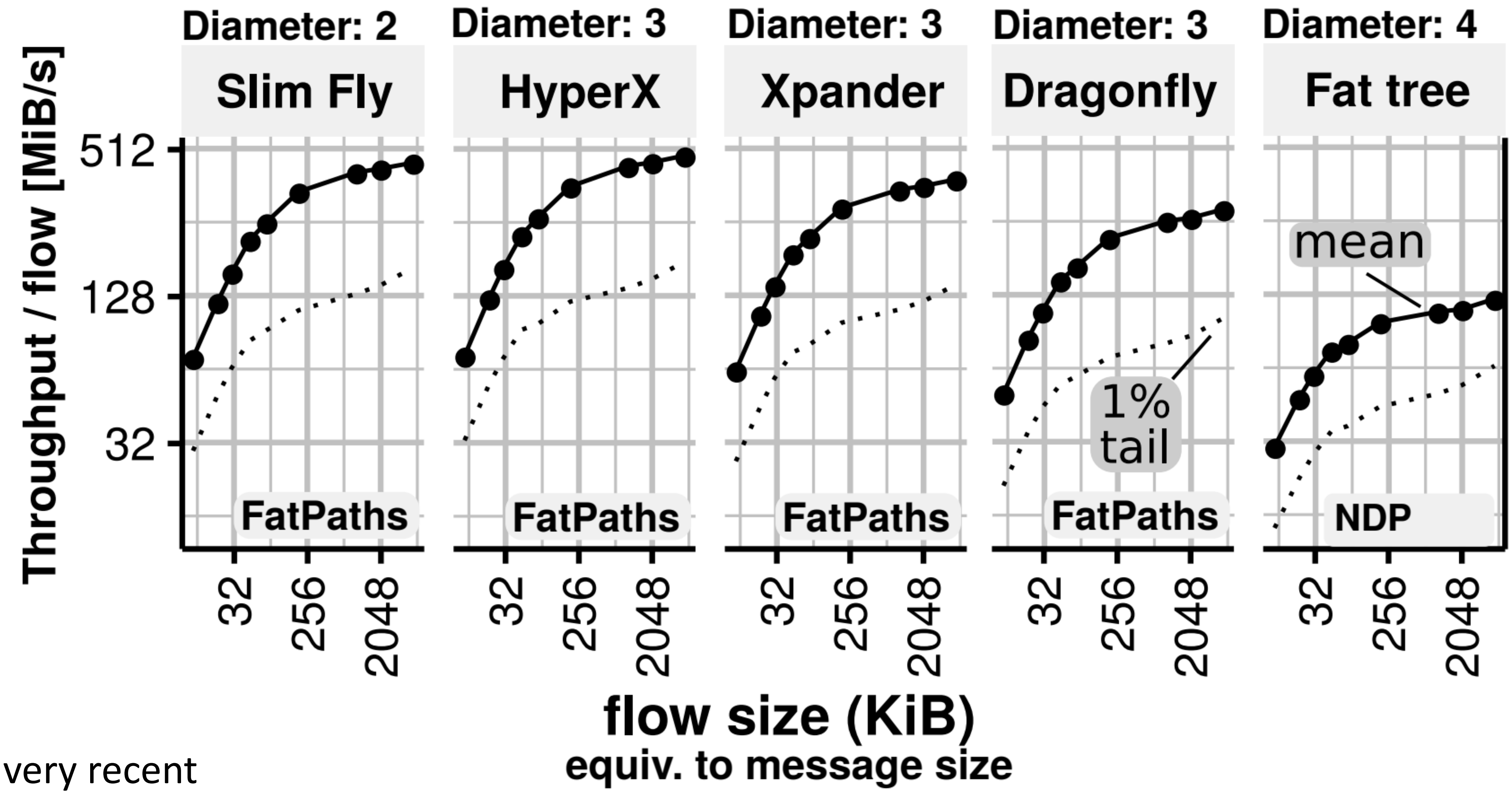
EVALUATION

N ≈ 10,000; comparable cost; random uniform traffic; “bare Ethernet” setting



Evaluation

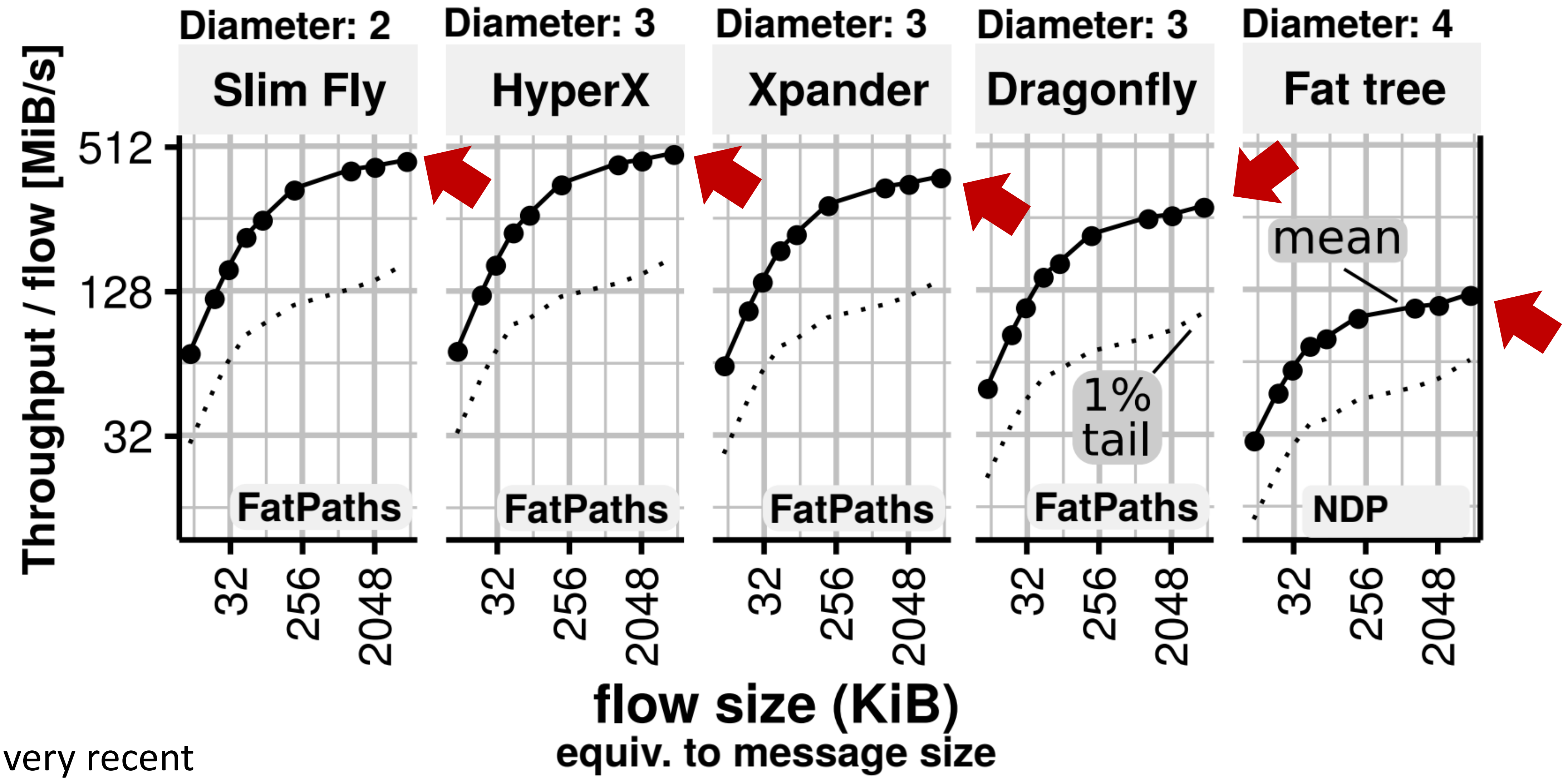
N ≈ 10,000; comparable cost; random uniform traffic; “bare Ethernet” setting



NDP: a very recent baseline for fat trees

EVALUATION

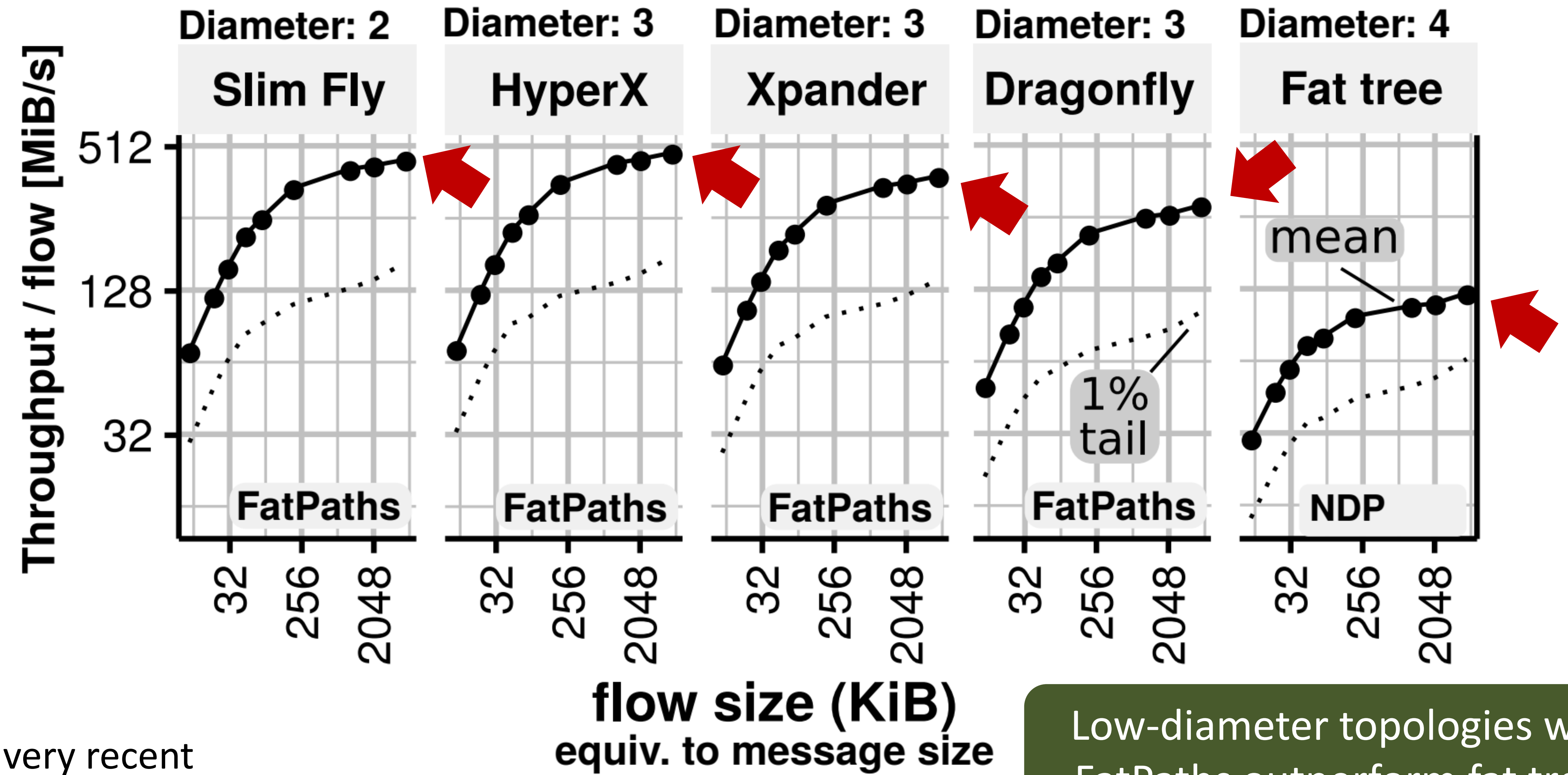
N ≈ 10,000; comparable cost; random uniform traffic; “bare Ethernet” setting



NDP: a very recent baseline for fat trees

EVALUATION

N ≈ 10,000; comparable cost; random uniform traffic; “bare Ethernet” setting



NDP: a very recent baseline for fat trees

Low-diameter topologies with FatPaths outperform fat trees

EVALUATION

N ≈ 10,000; comparable cost; random uniform traffic; "full TCP" setting

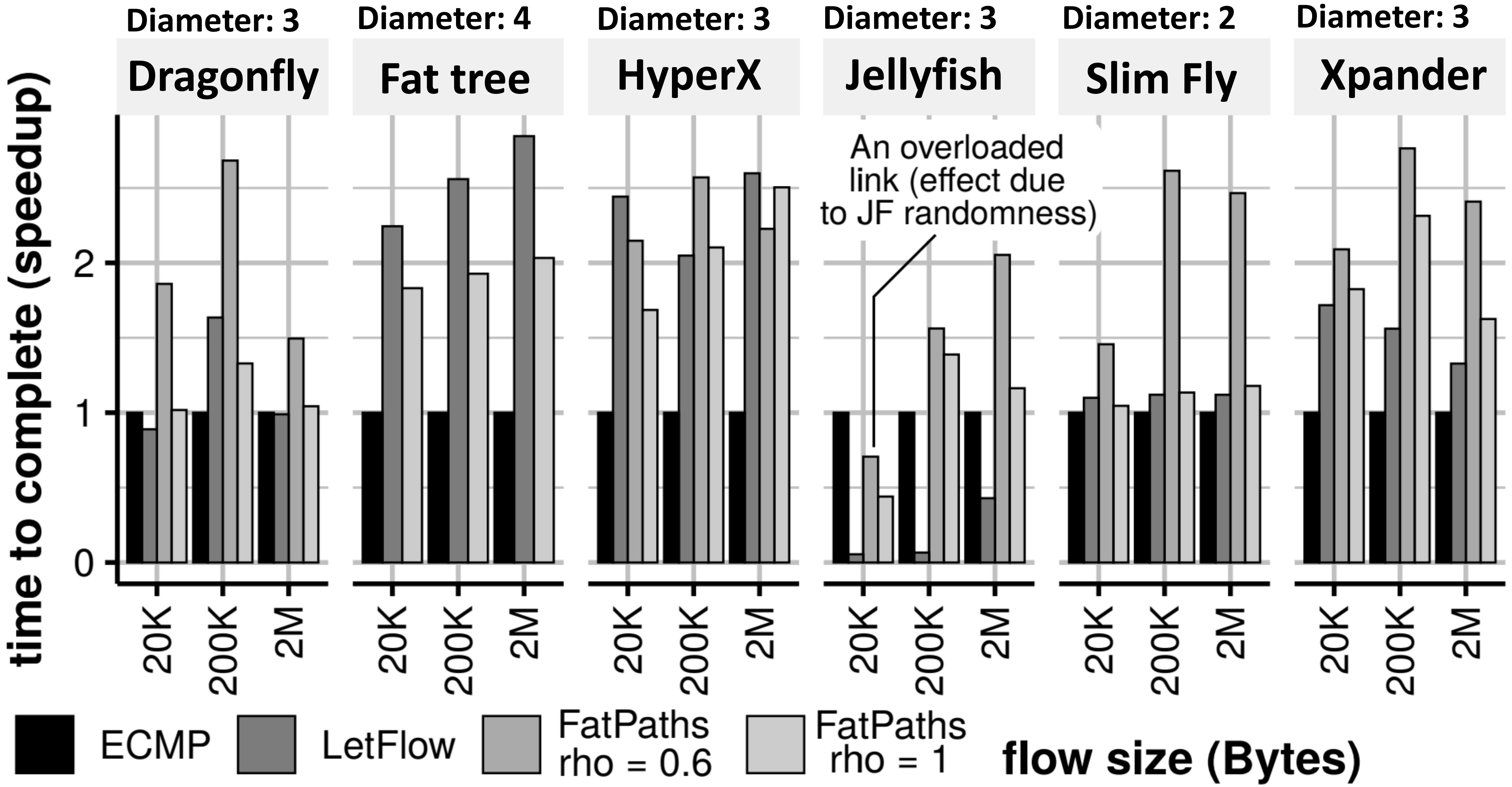
Diameter: 3	Diameter: 4	Diameter: 3	Diameter: 3	Diameter: 2	Diameter: 3
Dragonfly	**Fat tree**	**HyperX**	**Jellyfish**	**Slim Fly**	**Xpander**

ECMP: traditional static load balancing
LetFlow: recent adaptive load balancing

Speedups are measured over ECMP
rho: fraction of links kept in a layer

EVALUATION

N ≈ 10,000; comparable cost; random uniform traffic; "full TCP" setting

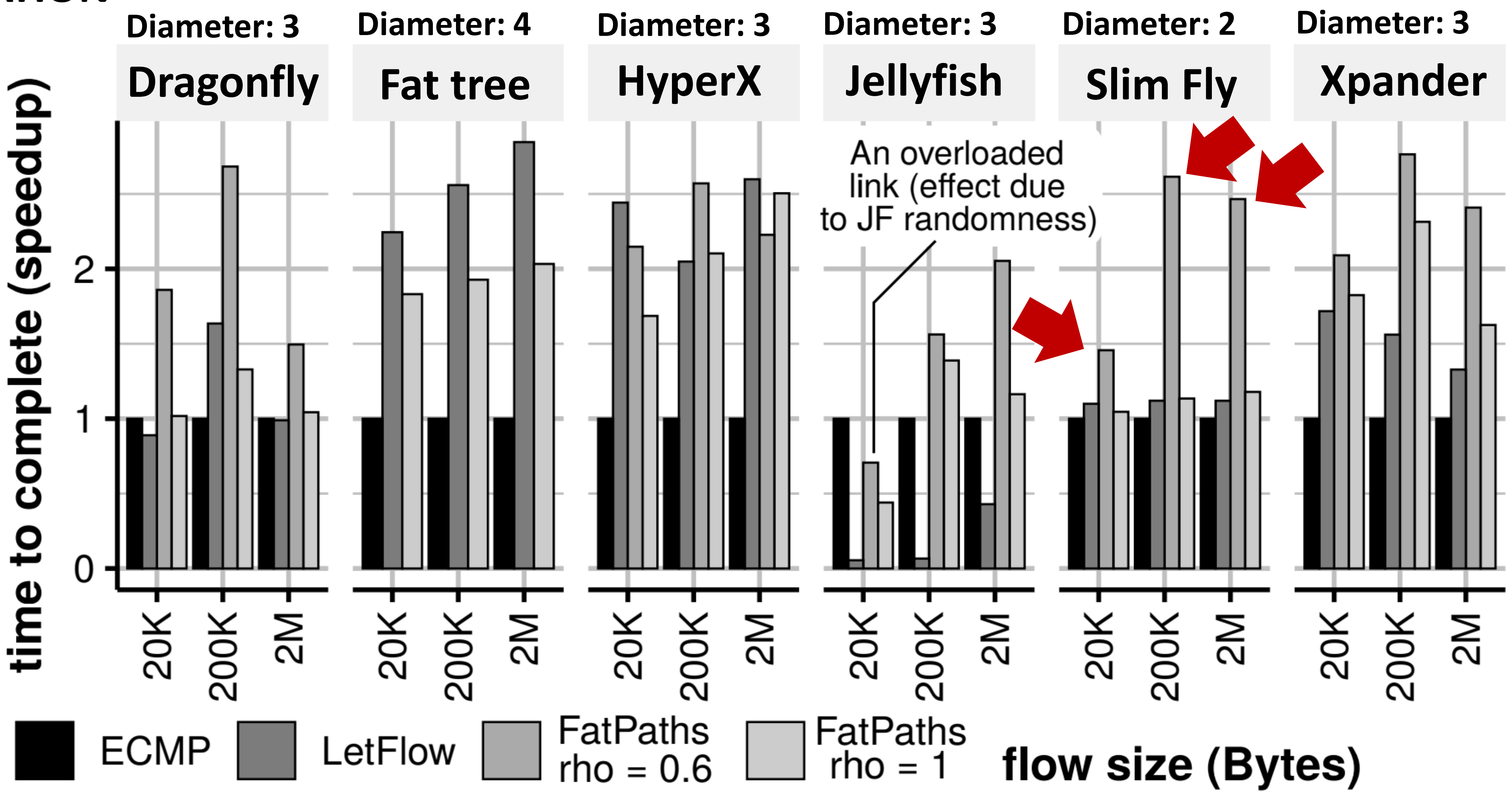


ECMP: traditional static load balancing
LetFlow: recent adaptive load balancing

Speedups are measured over ECMP
rho: fraction of links kept in a layer

EVALUATION

N ≈ 10,000; comparable cost; random uniform traffic; "full TCP" setting

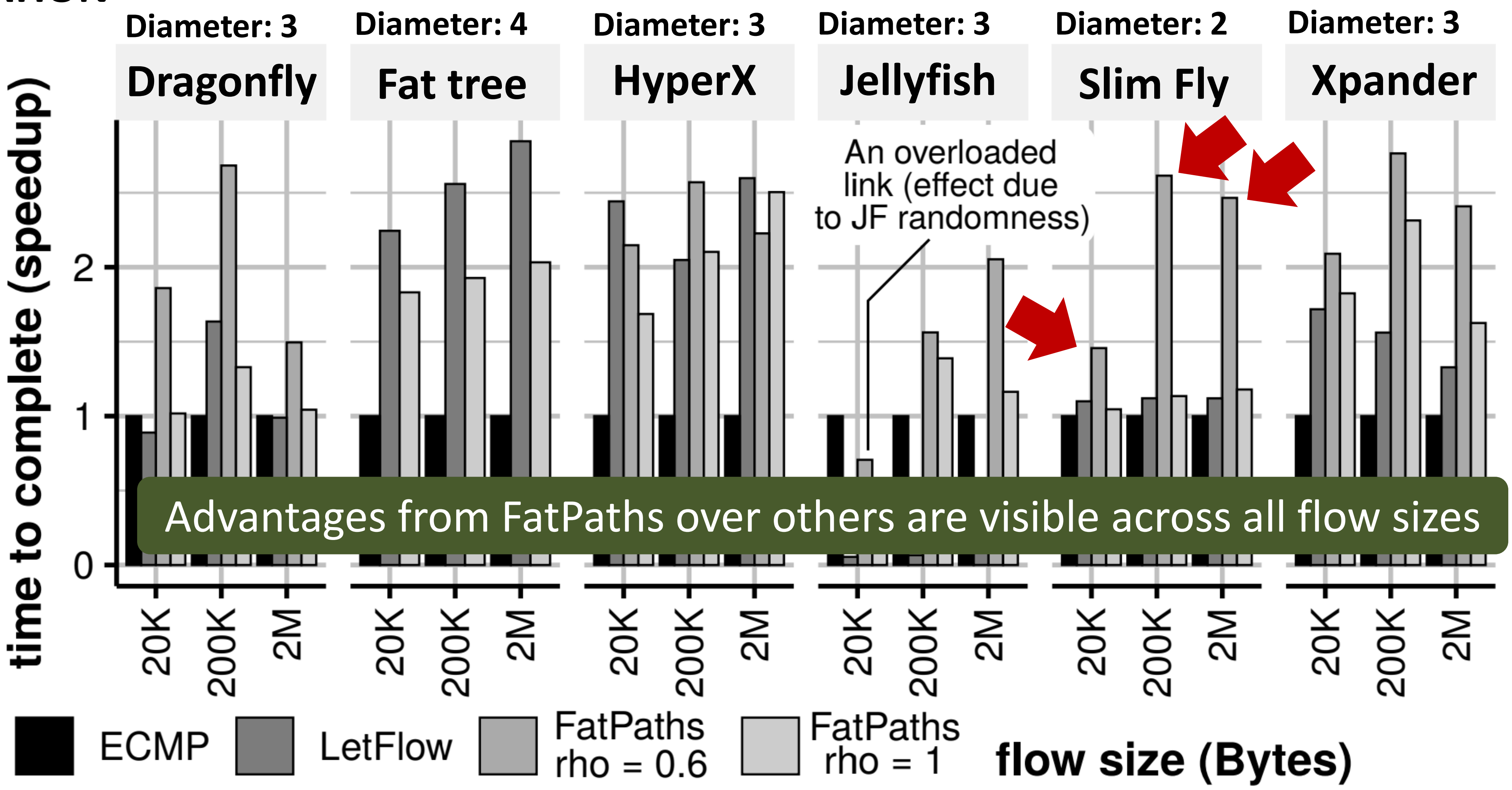


ECMP: traditional static load balancing
LetFlow: recent adaptive load balancing

Speedups are measured over ECMP
rho: fraction of links kept in a layer

EVALUATION

N ≈ 10,000; comparable cost; random uniform traffic; "full TCP" setting



ECMP: traditional static load balancing
LetFlow: recent adaptive load balancing

Speedups are measured over ECMP
rho: fraction of links kept in a layer

Survey of routing protocols

Survey of routing protocols

Focus: path diversity in routing

SURVEY OF ROUTING PROTOCOLS

Focus: path diversity in routing

Routing Scheme (Name, Abbreviation, Reference)	Stack Layer	Supported path diversity aspect						
		SP	NP	SM	MP	DP	ALB	AT
(1) SIMPLE ROUTING PROTOCOLS (often used as building blocks):								
Valiant load balancing (VLB) [17]	L2–L3	👎	👍	👎	👎	👎	👎	👍
Simple Spanning Tree (ST) [18]	L2	👎^{S}	👎^{S}	👎	👎	👎	👎	👍
Simple routing, e.g., OSPF [19]–[22]	L2, L3	👍	👎	👎	👎	👎	👎	👍
UGAL [23]	L2–L3	👍	👍	👎	👎	👎	👍	👍
ECMP [11], OMP [24], Pkt. Spraying [25]	L2, L3	👍	👎	👎	👍	👎	👎	👍
(2) ROUTING ARCHITECTURES:								
DCell [26]	L2–L3	👎	👍	👎	👎	👎	👎	👎
Monsoon [27]	L2, L3	🖒	🖒	👎	🖒	👎	👎	👎
PortLand [9]	L2	👍	👎	👎	👍	👎	👎	👎
DRILL [28], LocalFlow [29], DRB [30]	L2	👍	👎	👎	👍	👎	👍	👎
VL2 [31]	L3	👍	👎	👎	👍	👎	🖒	👎
Architecture by Al-Fares et al. [32]	L2–L3	👍	👎	👎	👍	👍	👍	👎
BCube [33]	L2–L3	👍	👎	👎	👍	👍	👎	👎
SEATTLE [34], others* [35]–[38]	L2	👍	👎	👎	👎	👎	👎	👍
VIRO [39]	L2–L3	👎^{S}	👎^{S}	👎	👎	👎	👎	👍
Ethernet on Air [40]	L2	👎^{S}	👎^{S}	👎	🖒^{R}	👎	👎	👍
PAST [41]	L2	🖒^{S}	🖒^{S}	👎	👎	👍	👎	👍
MLAG, MC-LAG, others [42]	L2	🖒	🖒	👎	🖒^{R}	👎	👎	👍
MOOSE [43]	L2	👍	👎	👎	🖒	👎	👎	👍
MPA [44]	L3	👍	👍	👎	👍	👎	👎	👍
AMP [45]	L3	👍	👎	👎	👍	👎	👍	👍
MSTP [46], GOE [47], Viking [48]	L2	👎^{S}	👎^{S}	👎	👍	👎	👎	👍
SPB [49], TRILL [50], Shadow MACs [51]	L2	👍	👎^{R}	👎	👍	👎	👎	👍
SPAIN [52]	L2	🖒^{S}	🖒^{S}	🖒^{S}	👍	👍	👎	👍
(3) Schemes for exposing/encoding paths (can be combined with FatPaths):								
XPath [53]	L3	👍	🖒	🖒	👍	👍	🖒	👍
Source routing for flexible DC fabric [54]	L3	👍	🖒^{R}	🖒^{R}	👎	👎	👎	🖒†
(3) FatPaths [This work]	L2–L3	👍	👍	👍	👍	👍	👍	👍

Survey of routing protocols

Focus: path diversity in routing

(1) Taxonomy of „path diversity support” in routing

Routing Scheme (Name, Abbreviation, Reference)	Stack Layer	Supported path diversity aspect						
		SP	NP	SM	MP	DP	ALB	AT
(1) SIMPLE ROUTING PROTOCOLS (often used as building blocks):								
Valiant load balancing (VLB) [17]	L2–L3	👎	👍	👎	👎	👎	👎	👍
Simple Spanning Tree (ST) [18]	L2	👎^{S}	👎^{S}	👎	👎	👎	👎	👍
Simple routing, e.g., OSPF [19]–[22]	L2, L3	👍	👎	👎	👎	👎	👎	👍
UGAL [23]	L2–L3	👍	👍	👎	👎	👎	👍	👍
ECMP [11], OMP [24], Pkt. Spraying [25]	L2, L3	👍	👎	👎	👍	👎	👎	👍
(2) ROUTING ARCHITECTURES:								
DCell [26]	L2–L3	👎	👍	👎	👎	👎	👎	👎
Monsoon [27]	L2, L3	◐	◐	👎	◐	👎	👎	👎
PortLand [9]	L2	👍	👎	👎	👍	👎	👎	👎
DRILL [28], LocalFlow [29], DRB [30]	L2	👍	👎	👎	👍	👎	👍	👎
VL2 [31]	L3	👍	👎	👎	👍	👎	◐	👎
Architecture by Al-Fares et al. [32]	L2–L3	👍	👎	👎	👍	👍	👍	👎
BCube [33]	L2–L3	👍	👎	👎	👍	👍	👎	👎
SEATTLE [34], others* [35]–[38]	L2	👍	👎	👎	👎	👎	👎	👍
VIRO [39]	L2–L3	👎^{S}	👎^{S}	👎	👎	👎	👎	👍
Ethernet on Air [40]	L2	👎^{S}	👎^{S}	👎	◐^{R}	👎	👎	👍
PAST [41]	L2	◐^{S}	◐^{S}	👎	👎	👍	👎	👍
MLAG, MC-LAG, others [42]	L2	◐	◐	👎	◐^{R}	👎	👎	👍
MOOSE [43]	L2	👍	👎	👎	◐	👎	👎	👍
MPA [44]	L3	👍	👍	👎	👍	👎	👎	👍
AMP [45]	L3	👍	👎	👎	👍	👎	👍	👍
MSTP [46], GOE [47], Viking [48]	L2	👎^{S}	👎^{S}	👎	👍	👎	👎	👍
SPB [49], TRILL [50], Shadow MACs [51]	L2	👍	👎^{R}	👎	👍	👎	👎	👍
SPAIN [52]	L2	◐^{S}	◐^{S}	◐^{S}	👍	👍	👎	👍
(3) Schemes for exposing/encoding paths (can be combined with FatPaths):								
XPath [53]	L3	👍	◐	◐	👍	👍	◐	👍
Source routing for flexible DC fabric [54]	L3	👍	◐^{R}	◐^{R}	👎	👎	👎	◐†
(3) FatPaths [This work]	L2–L3	👍	👍	👍	👍	👍	👍	👍

SURVEY OF ROUTING PROTOCOLS

Focus: path diversity in routing

(1) Taxonomy of „path diversity support" in routing

(2) Analysis of routing protocols

Routing Scheme (Name, Abbreviation, Reference)	Stack Layer	Supported path diversity aspect						
		SP	NP	SM	MP	DP	ALB	AT
(1) SIMPLE ROUTING PROTOCOLS (often used as building blocks):								
Valiant load balancing (VLB) [17]	L2–L3	👎	👍	👎	👎	👎	👎	👍
Simple Spanning Tree (ST) [18]	L2	👎^{S}	👎^{S}	👎	👎	👎	👎	👍
Simple routing, e.g., OSPF [19]–[22]	L2, L3	👍	👎	👎	👎	👎	👎	👍
UGAL [23]	L2–L3	👍	👍	👎	👎	👎	👍	👍
ECMP [11], OMP [24], Pkt. Spraying [25]	L2, L3	👍	👎	👎	👍	👎	👎	👍
(2) ROUTING ARCHITECTURES:								
DCell [26]	L2–L3	👎	👍	👎	👎	👎	👎	👎
Monsoon [27]	L2, L3	🖒	🖒	👎	🖒	👎	👎	👎
PortLand [9]	L2	👍	👎	👎	👍	👎	👎	👎
DRILL [28], LocalFlow [29], DRB [30]	L2	👍	👎	👎	👍	👎	👍	👎
VL2 [31]	L3	👍	👎	👎	👍	👎	🖒	👎
Architecture by Al-Fares et al. [32]	L2–L3	👍	👎	👎	👍	👍	👍	👎
BCube [33]	L2–L3	👍	👎	👎	👍	👍	👎	👎
SEATTLE [34], others* [35]–[38]	L2	👍	👎	👎	👎	👎	👎	👍
VIRO [39]	L2–L3	👎^{S}	👎^{S}	👎	👎	👎	👎	👍
Ethernet on Air [40]	L2	👎^{S}	👎^{S}	👎	🖒^{R}	👎	👎	👍
PAST [41]	L2	🖒^{S}	🖒^{S}	👎	👎	👍	👎	👍
MLAG, MC-LAG, others [42]	L2	🖒	🖒	👎	🖒^{R}	👎	👎	👍
MOOSE [43]	L2	👍	👎	👎	🖒	👎	👎	👍
MPA [44]	L3	👍	👍	👎	👍	👎	👎	👍
AMP [45]	L3	👍	👎	👎	👍	👎	👍	👍
MSTP [46], GOE [47], Viking [48]	L2	👎^{S}	👎^{S}	👎	👍	👎	👎	👍
SPB [49], TRILL [50], Shadow MACs [51]	L2	👍	👎^{R}	👎	👍	👎	👎	👍
SPAIN [52]	L2	🖒^{S}	🖒^{S}	🖒^{S}	👍	👍	👎	👍
(3) Schemes for exposing/encoding paths (can be combined with FatPaths):								
XPath [53]	L3	👍	🖒	🖒	👍	👍	🖒	👍
Source routing for flexible DC fabric [54]	L3	👍	🖒^{R}	🖒^{R}	👎	👎	👎	🖒†
(3) FatPaths [This work]	L2–L3	👍	👍	👍	👍	👍	👍	👍

Survey of routing protocols

Focus: path diversity in routing

(1) Taxonomy of „path diversity support” in routing

(2) Analysis of routing protocols

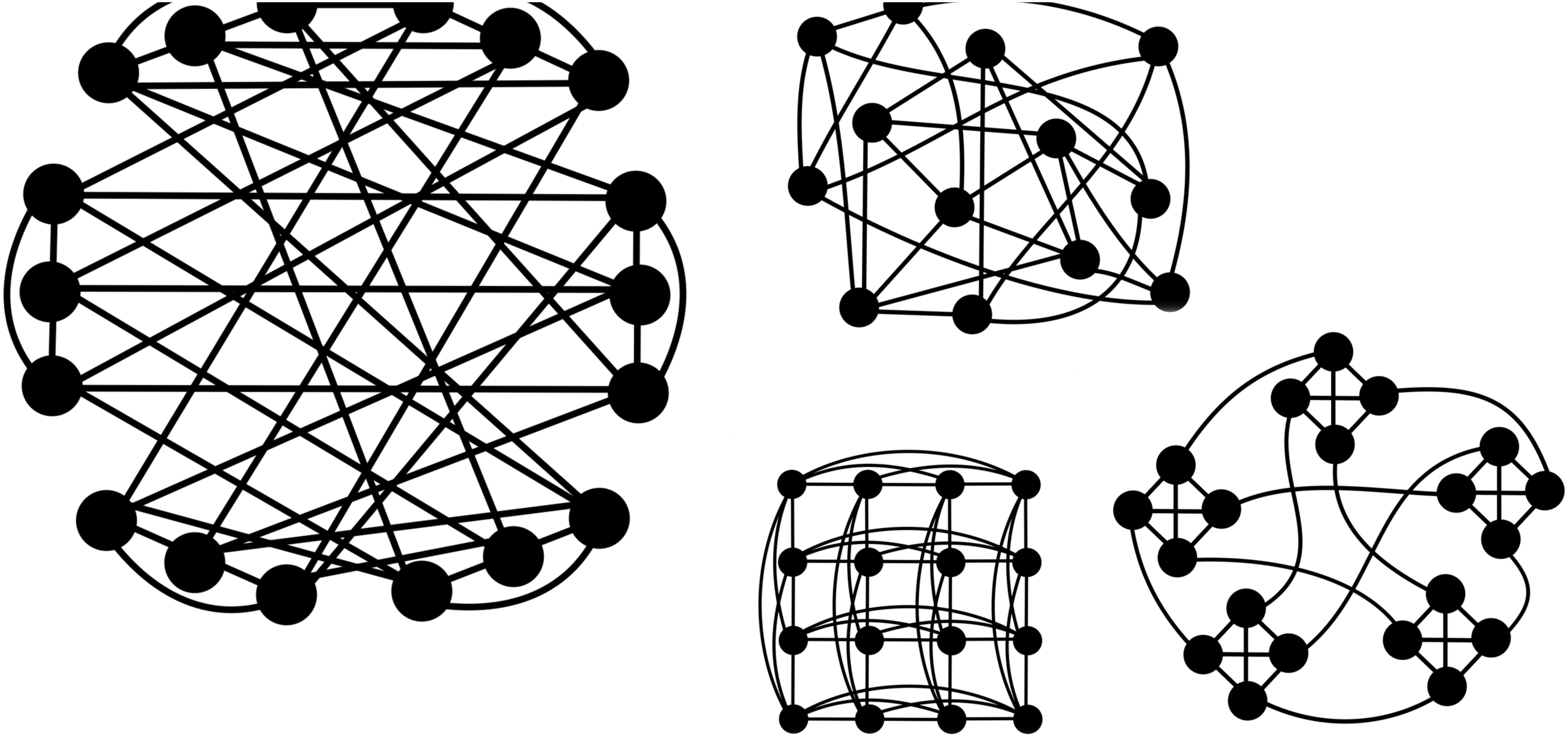
Routing Scheme (Name, Abbreviation, Reference)	Stack Layer	Supported path diversity aspect						
		SP	NP	SM	MP	DP	ALB	AT
(1) SIMPLE ROUTING PROTOCOLS (often used as building blocks):								
Valiant load balancing (VLB) [17]	L2–L3	👎	👍	👎	👎	👎	👎	👍
Simple Spanning Tree (ST) [18]	L2	👎^{S}	👎^{S}	👎	👎	👎	👎	👍
Simple routing, e.g., OSPF [19]–[22]	L2, L3	👍	👎	👎	👎	👎	👎	👍
UGAL [23]	L2–L3	👍	👍	👎	👎	👎	👍	👍
ECMP [11], OMP [24], Pkt. Spraying [25]	L2, L3	👍	👎	👎	👍	👎	👎	👍
(2) ROUTING ARCHITECTURES:								
DCell [26]	L2–L3	👎	👍	👎	👎	👎	👎	👎
Monsoon [27]	L2, L3	(👍)	(👍)	👎	(👍)	👎	👎	👎
PortLand [9]	L2	👍	👎	👎	👍	👎	👎	👎
DRILL [28], LocalFlow [29], DRB [30]	L2	👍	👎	👎	👍	👎	👍	👎
VL2 [31]	L3	👍	👎	👎	👍	👎	(👍)	👎
Architecture by Al-Fares et al. [32]	L2–L3	👍	👎	👎	👍	👍	👍	👎
BCube [33]	L2–L3	👍	👎	👎	👍	👍	👎	👎
SEATTLE [34], others* [35]–[38]	L2	👍	👎	👎	👎	👎	👎	👍
VIRO [39]	L2–L3	👎^{S}	👎^{S}	👎	👎	👎	👎	👍
Ethernet on Air [40]	L2	👎^{S}	👎^{S}	👎	(👍)^{R}	👎	👎	👍
PAST [41]	L2	(👍)^{S}	(👍)^{S}	👎	👎	👍	👎	👍
MLAG, MC-LAG, others [42]	L2	(👍)	(👍)	👎	(👍)^{R}	👎	👎	👍
MOOSE [43]	L2	👍	👎	👎	(👍)	👎	👎	👍
MPA [44]	L3	👍	👍	👎	👍	👎	👎	👍
AMP [45]	L3	👍	👎	👎	👍	👎	👍	👍
MSTP [46], GOE [47], Viking [48]	L2	👎^{S}	👎^{S}	👎	👍	👎	👎	👍
SPB [49], TRILL [50], Shadow MACs [51]	L2	👍	👎^{R}	👎	👍	👎	👎	👍
SPAIN [52]	L2	(👍)^{S}	(👍)^{S}	(👍)^{S}	👍	👍	👎	👍
(3) Schemes for exposing/encoding paths (can be combined with FatPaths):								
XPath [53]	L3	👍	(👍)	(👍)	👍	👍	(👍)	👍
Source routing for flexible DC fabric [54]	L3	👍	(👍)^{R}	(👍)^{R}	👎	👎	👎	(👍)†
(3) FatPaths [This work]	L2–L3	👍	👍	👍	👍	👍	👍	👍

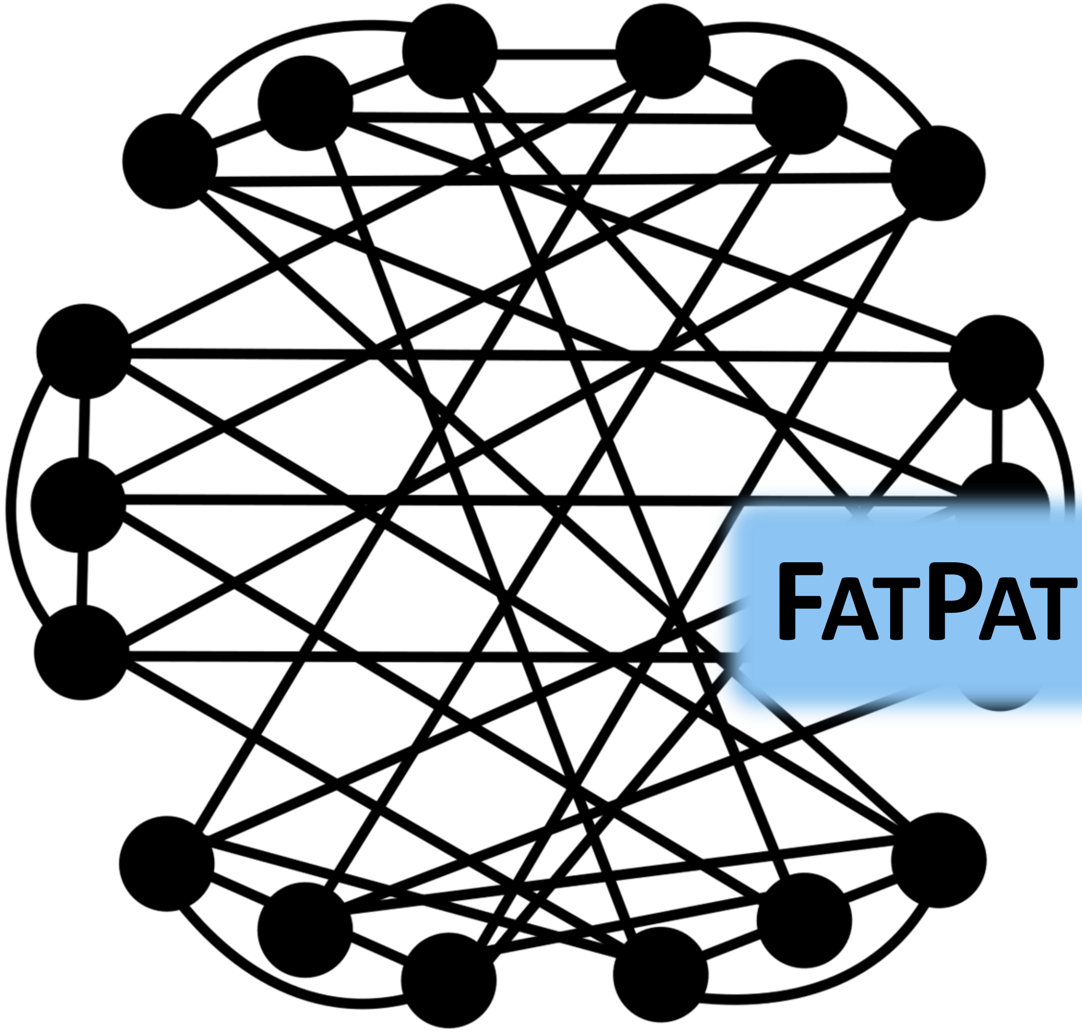
1

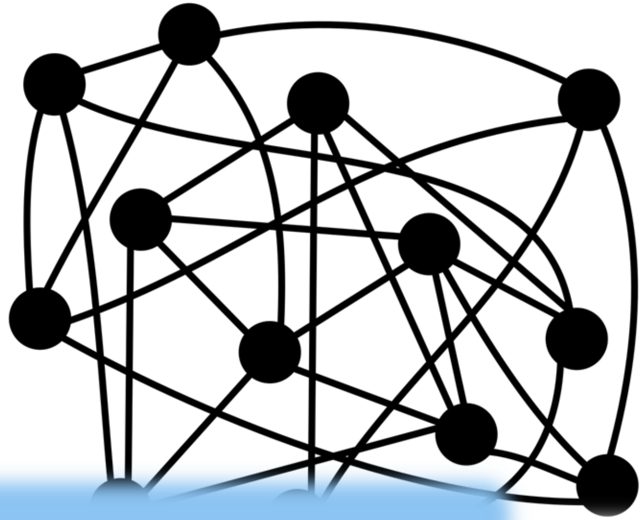
High-Performance Routing with Multipathing and Path Diversity in Supercomputers and Data Centers

Maciej Besta[1], Jens Domke[2], Marcel Schneider[1], Marek Konieczny[3], Salvatore Di Girolamo[1], Timo Schneider[1], Ankit Singla[1], Torsten Hoefler[1]

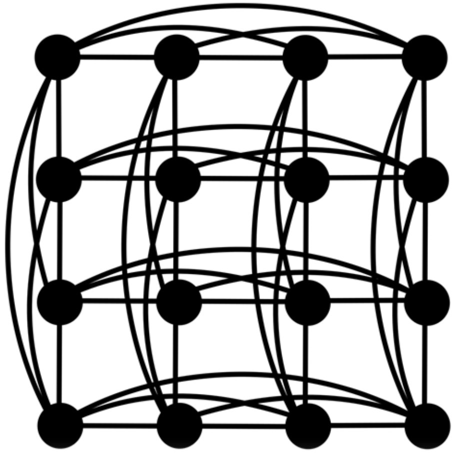
[1]Department of Computer Science, ETH Zurich; [2]RIKEN Center for Computational Science (R-CCS)
[3]Department of Computer Science, Electronics and Telecommunications; AGH-UST

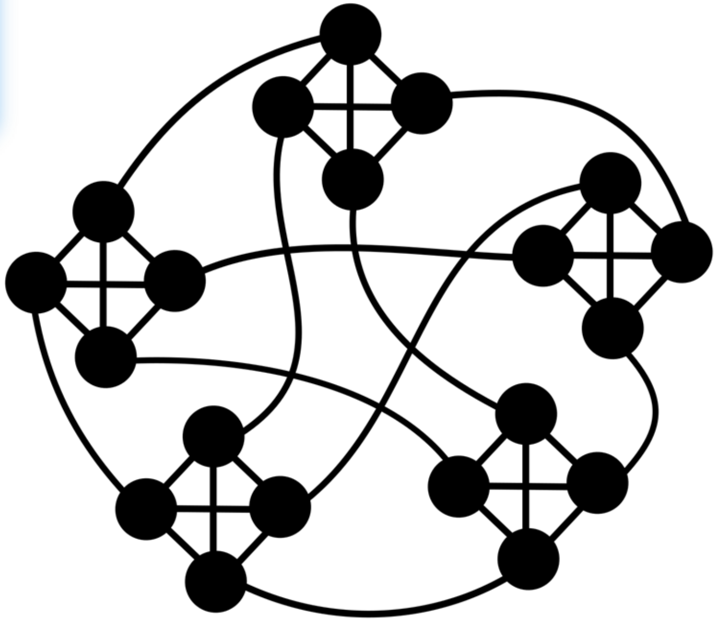




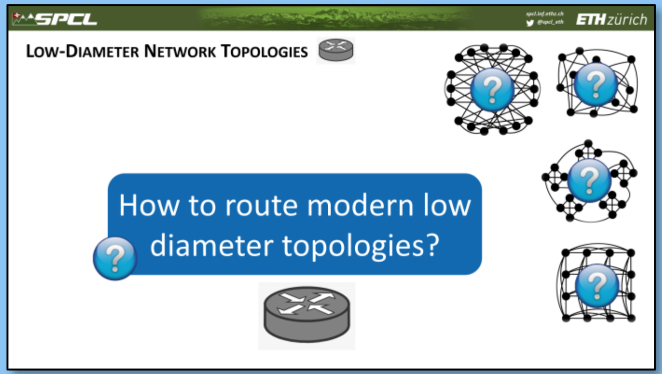


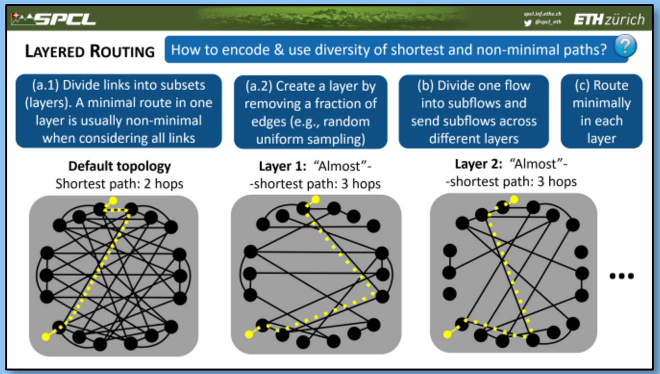
FatPaths Overview

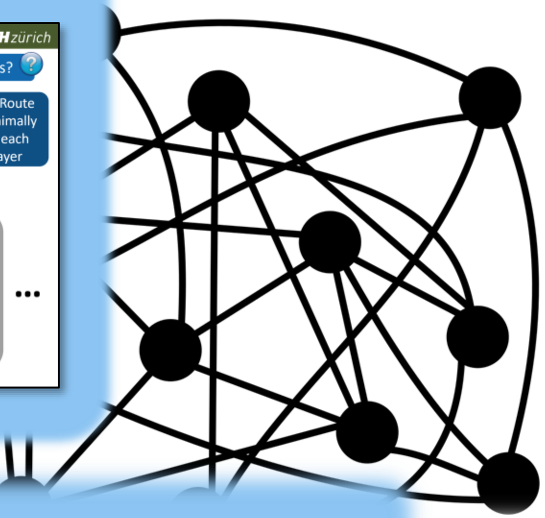


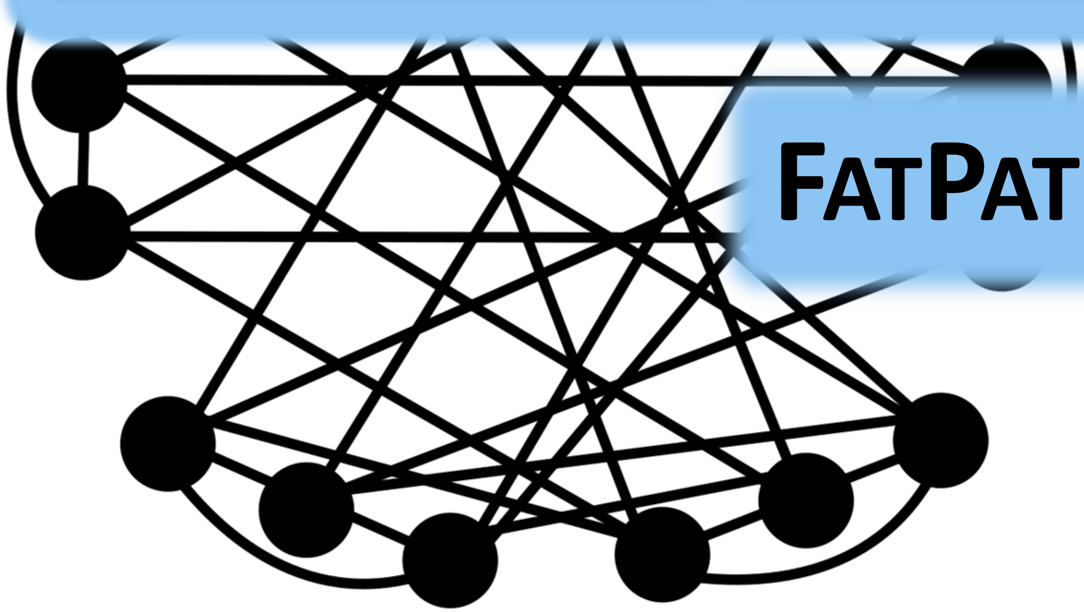


A ROUTING SCHEME FOCUSING ON LOW-DIAMETER TOPOLOGIES

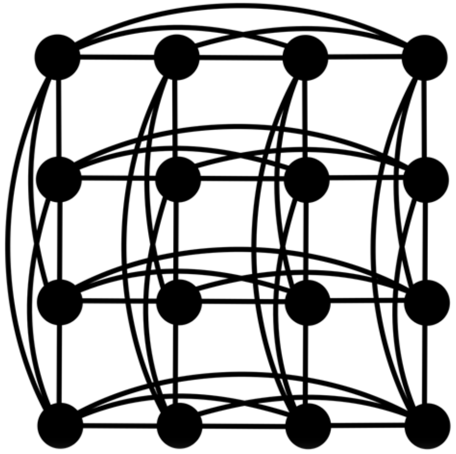


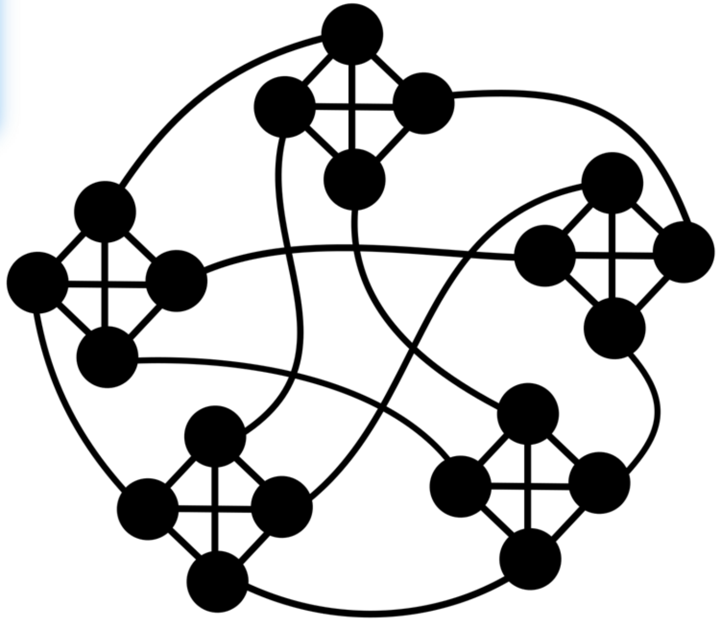




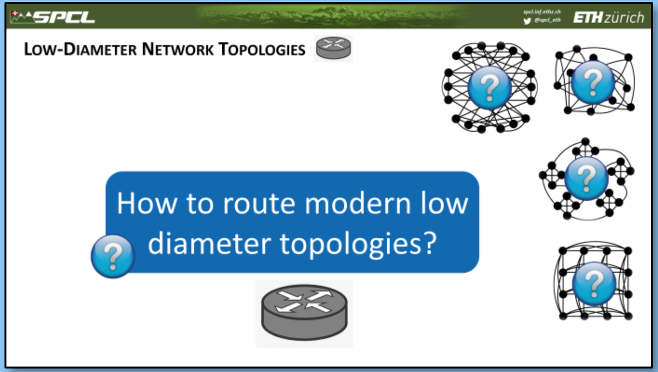


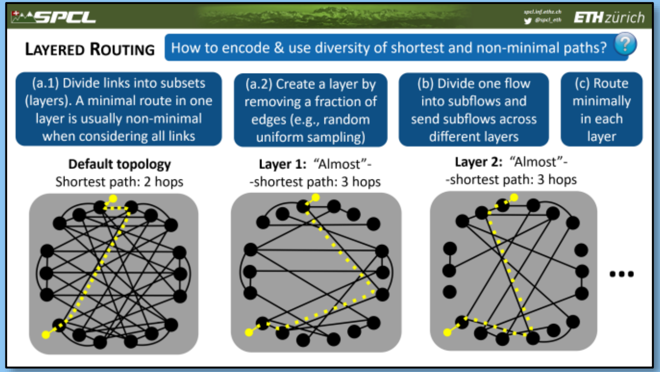
FatPaths Overview



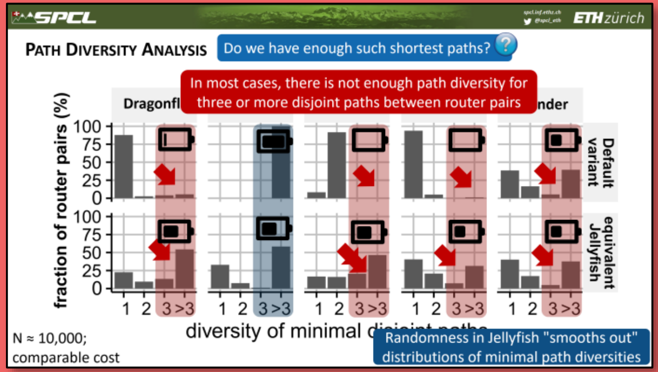


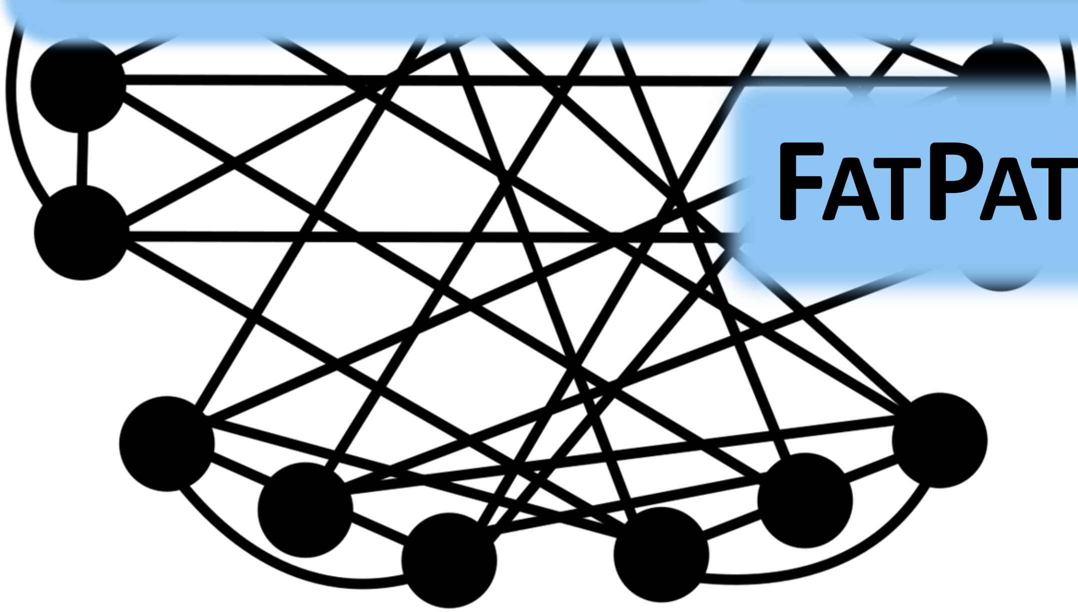
A routing scheme focusing on low-diameter topologies



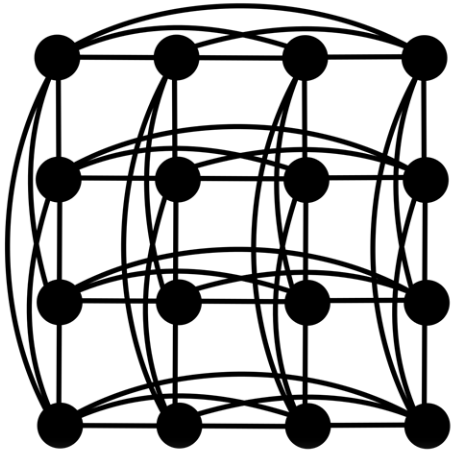


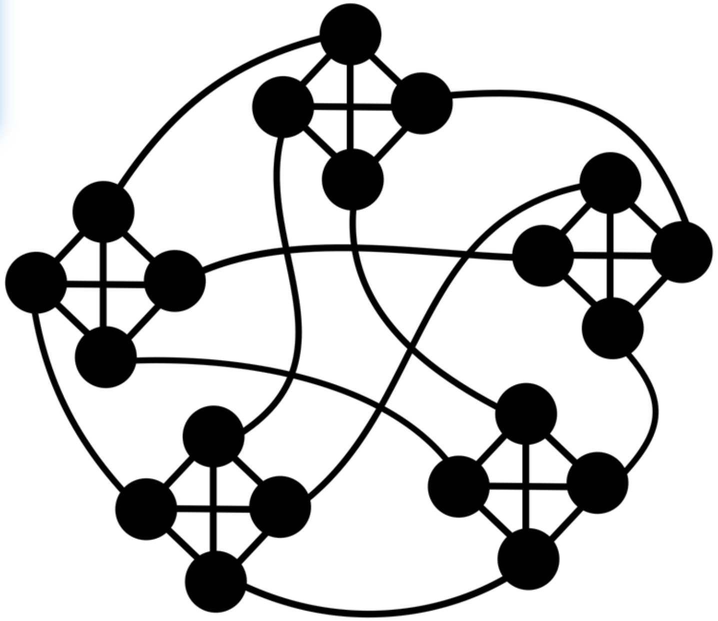
Broad analysis of path diversity





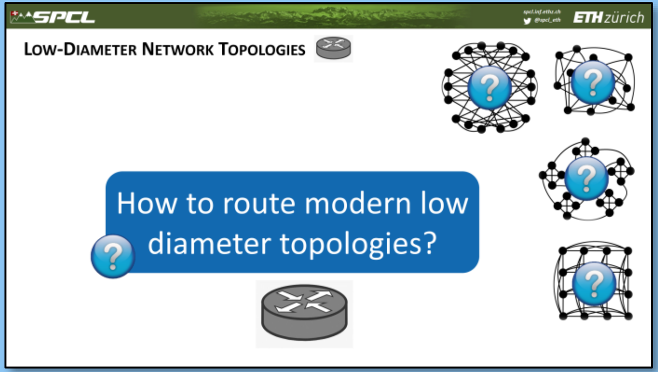
FatPaths Overview

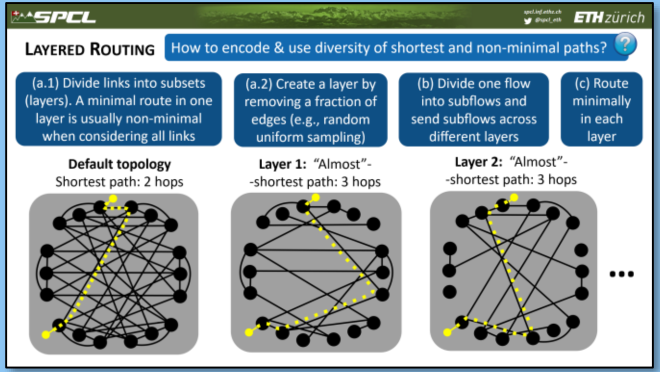




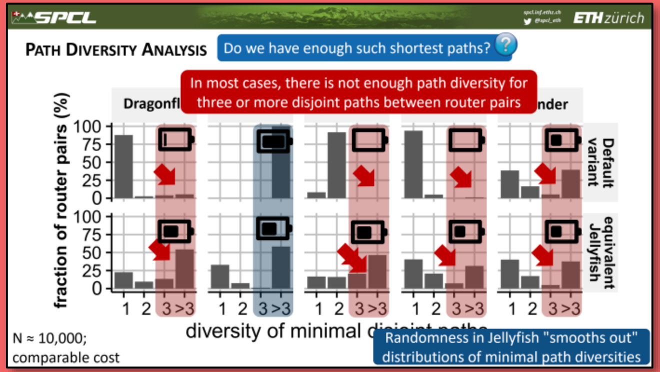
FatPaths Overview

A routing scheme focusing on low-diameter topologies

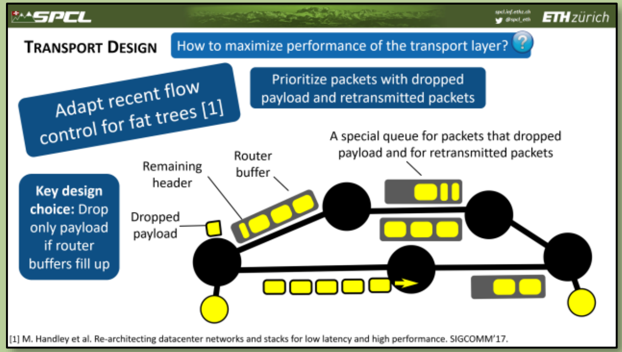


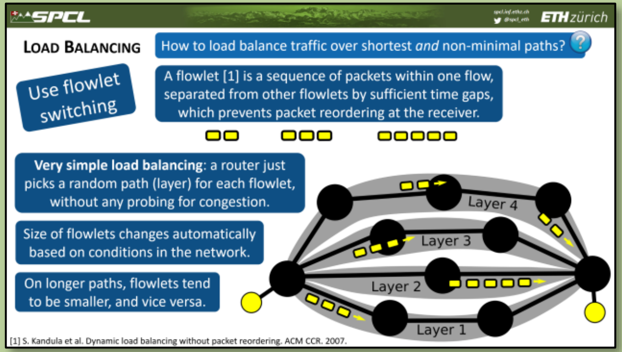


Broad analysis of path diversity



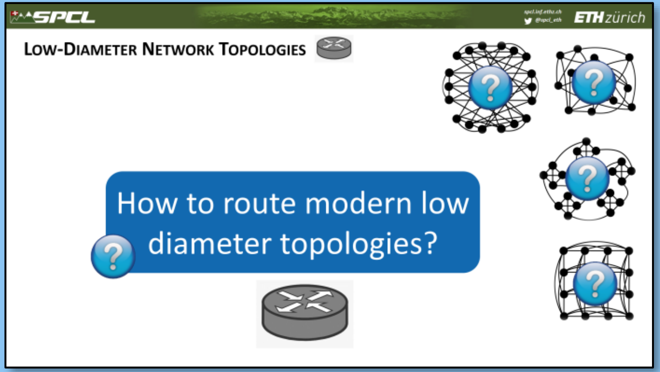
High-performance networking architecture

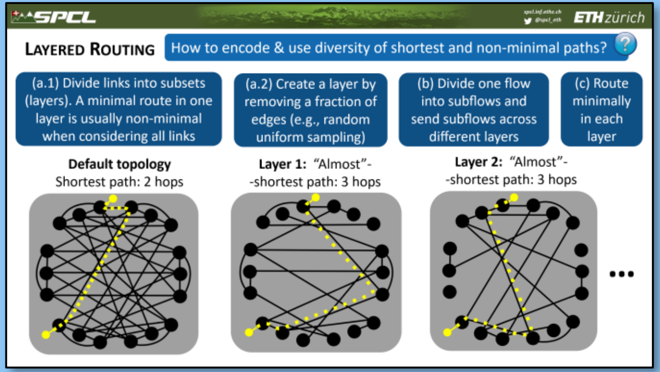




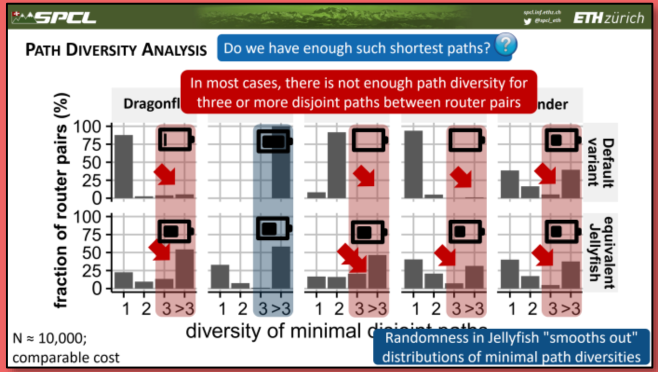
FatPaths Overview

A routing scheme focusing on low-diameter topologies

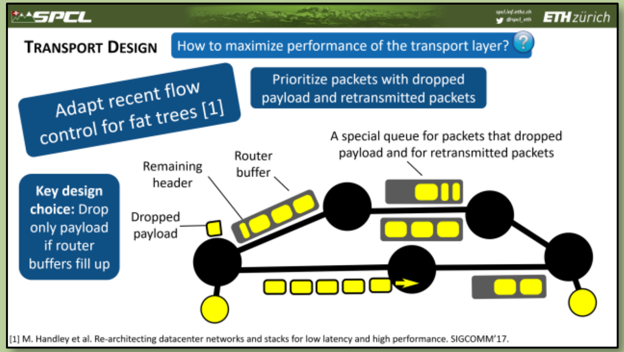


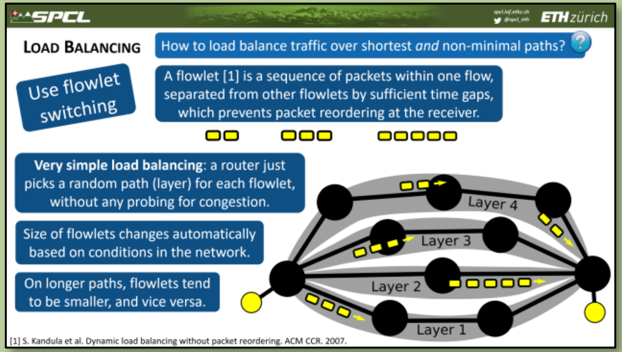


Broad analysis of path diversity



High-performance networking architecture





Rich evaluation, theoretical analysis, taxonomy