PCF: Provably Resilient Flexible Routing

Chuan Jiang, Sanjay Rao, Mohit Tawarmalani

Purdue University

ACM SIGCOMM 2020

Background

- The network performance requirements are increasingly stringent.
    - Over a 5 year period, traffic has been increased 100X and performance must be met 99.99% of time (vs. 99% of the time)[1].

- Failures of network components are routine and they have great impact on network performance.

[1] Hong et al, B4 and after: managing hierarchy, partitioning, and asymmetry for availability and scale in google's software-defined WAN. SIGCOMM 2018.

Background

- The network performance requirements are increasingly stringent.
  - Over a 5 year period, traffic has been increased 100X and performance must be met 99.99% of time (vs. 99% of the time)[1].

- Failures of network components are routine and they have great impact on network performance.

Design the networks so that the desired traffic can be served over a ***target set of failures***.

[1] Hong et al, B4 and after: managing hierarchy, partitioning, and asymmetry for availability and scale in google's software-defined WAN. SIGCOMM 2018.

Congestion-free routing

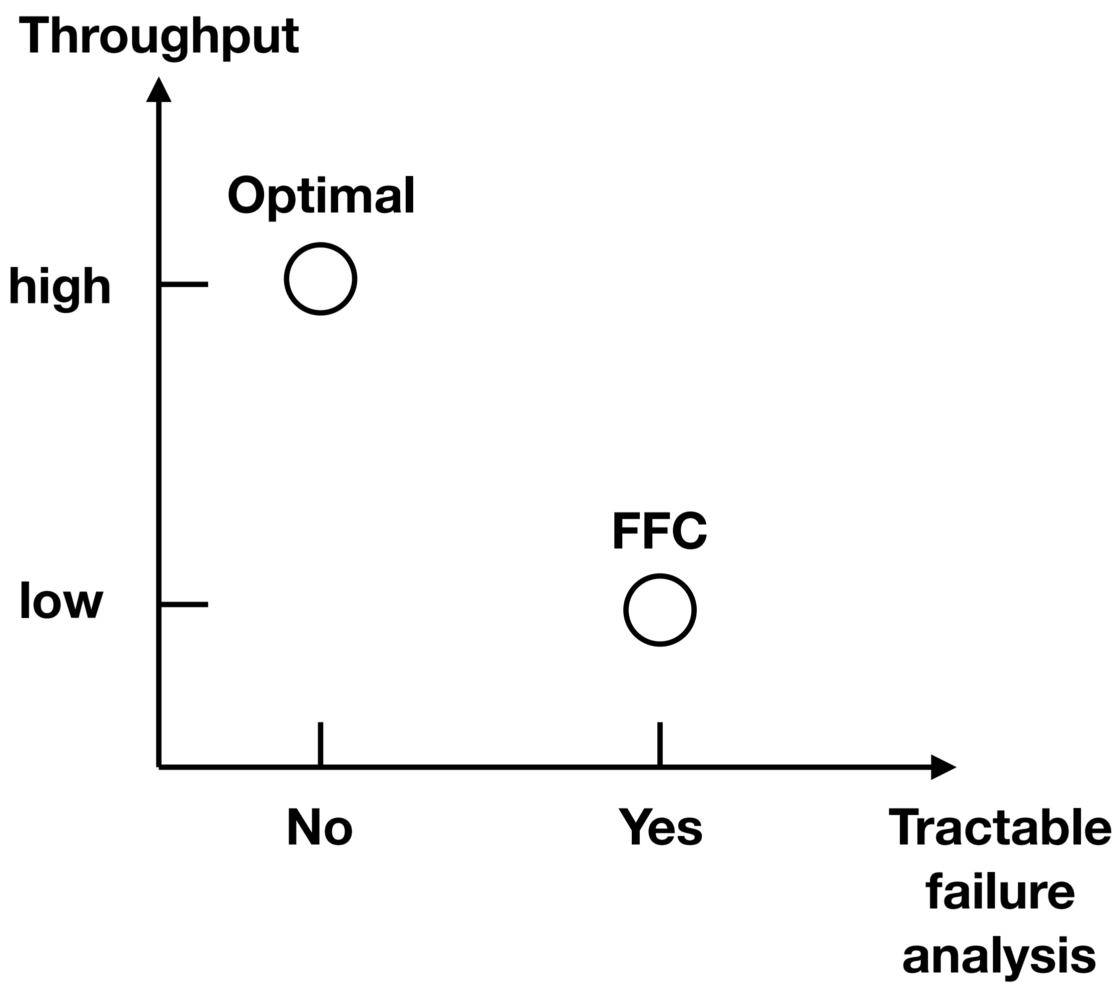
- Traditional traffic engineering: links may be overloaded upon failures[1, 2]
- Many works[3, 4, 5] have been developed to design congestion-free mechanisms.
    - ***Guarantee*** a given throughput can be sustained under failures.
    - ***Tractable*** models to deal with ***large state space*** of failure scenarios (e.g, f simultaneous link failures)
    - Typically involve ***light-weight*** online operations on failures
- FFC[3] is the state-of-the-art mechanism and uses tunnel-based forwarding.
    - A set of pre-selected tunnels and traffic demand are provided to FFC.
    - It computes reservations on tunnels so that throughput can be guaranteed across failures.

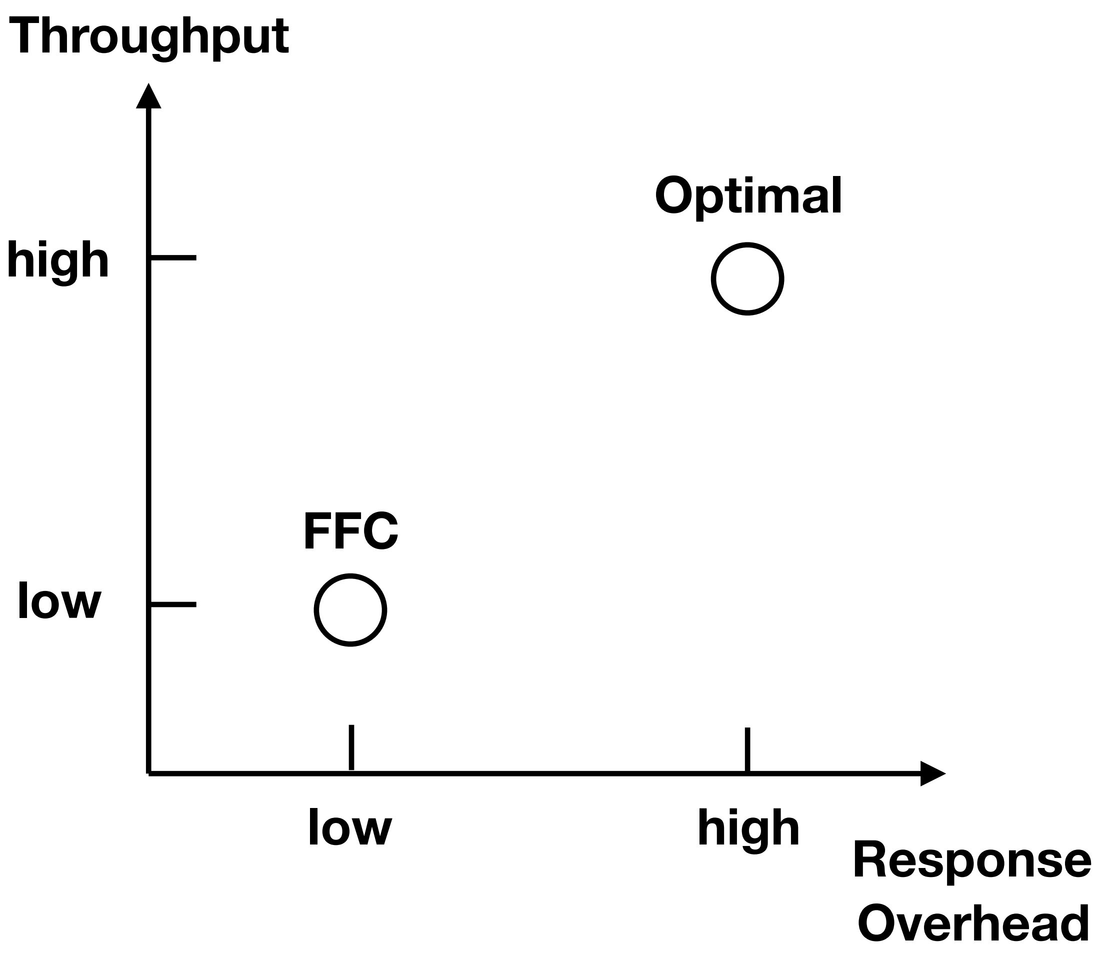
[1] Hong et al, Achieving high utilization with software-driven WAN, SIGCOMM 2013.
[2] Jain et al, B4: Experience with a globally- deployed software defined wan, SIGCOMM 2013.
[3] Liu et al, Traffic engineering with forward fault correction, SIGCOMM 2014.
[4] Sinha et al, Network design for tolerating multiple link failures using Fast Re-route (FRR), *DRCN* 2014.
[5] Wang et al, R3: resilient routing reconfiguration, SIGCOMM 2010.

Congestion-free routing vs. optimal routing

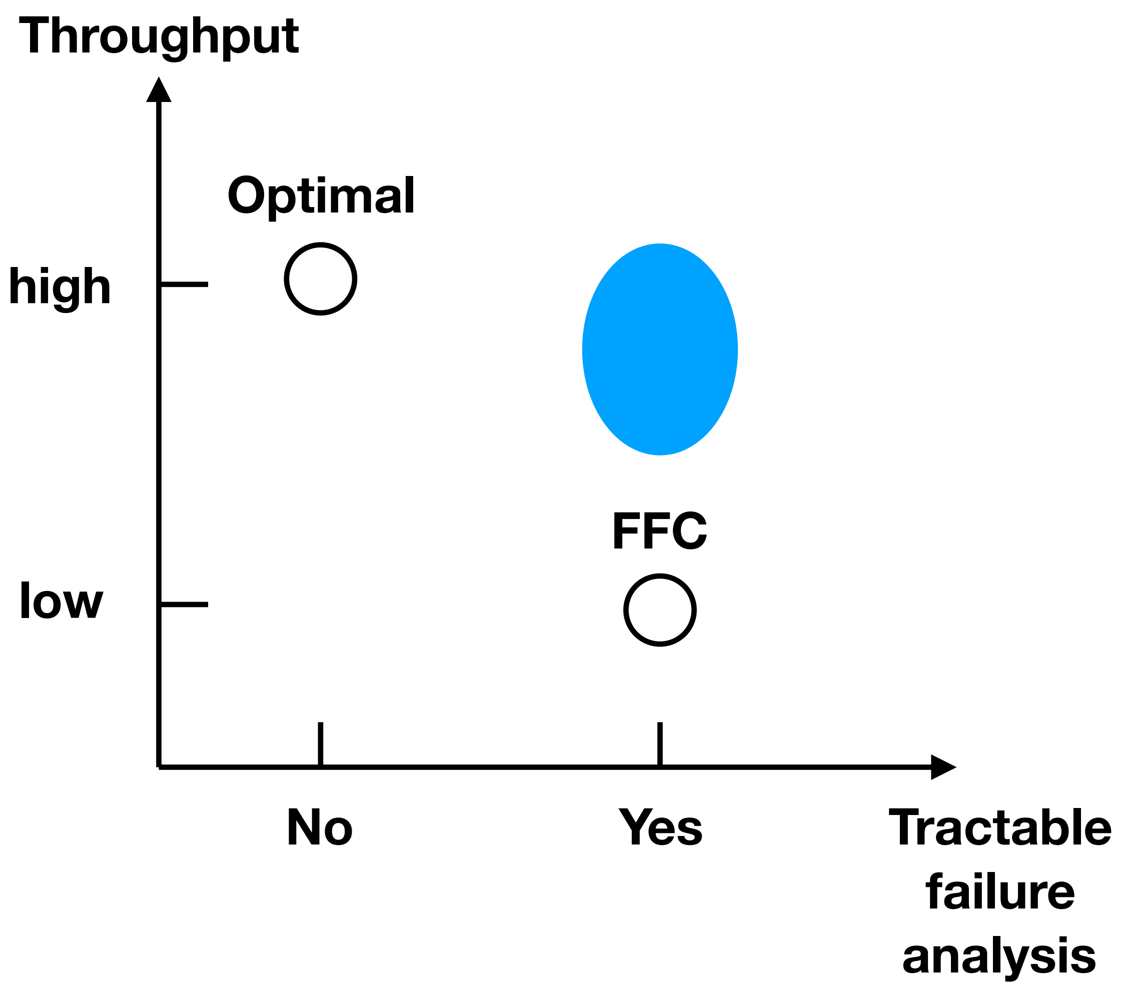
- FFC's mechanism is ***not flexible enough*** and its throughput can be very ***conservative***.
- Optimal mechanism
    - ***Most flexible***
    - It recomputes the best routing online for each scenario each time when a failure occurs, which always provide the ***best throughput***.
    - It brings ***higher response overhead*** related to online operations.
    - It is ***intractable*** to provide a performance guarantee under failures.

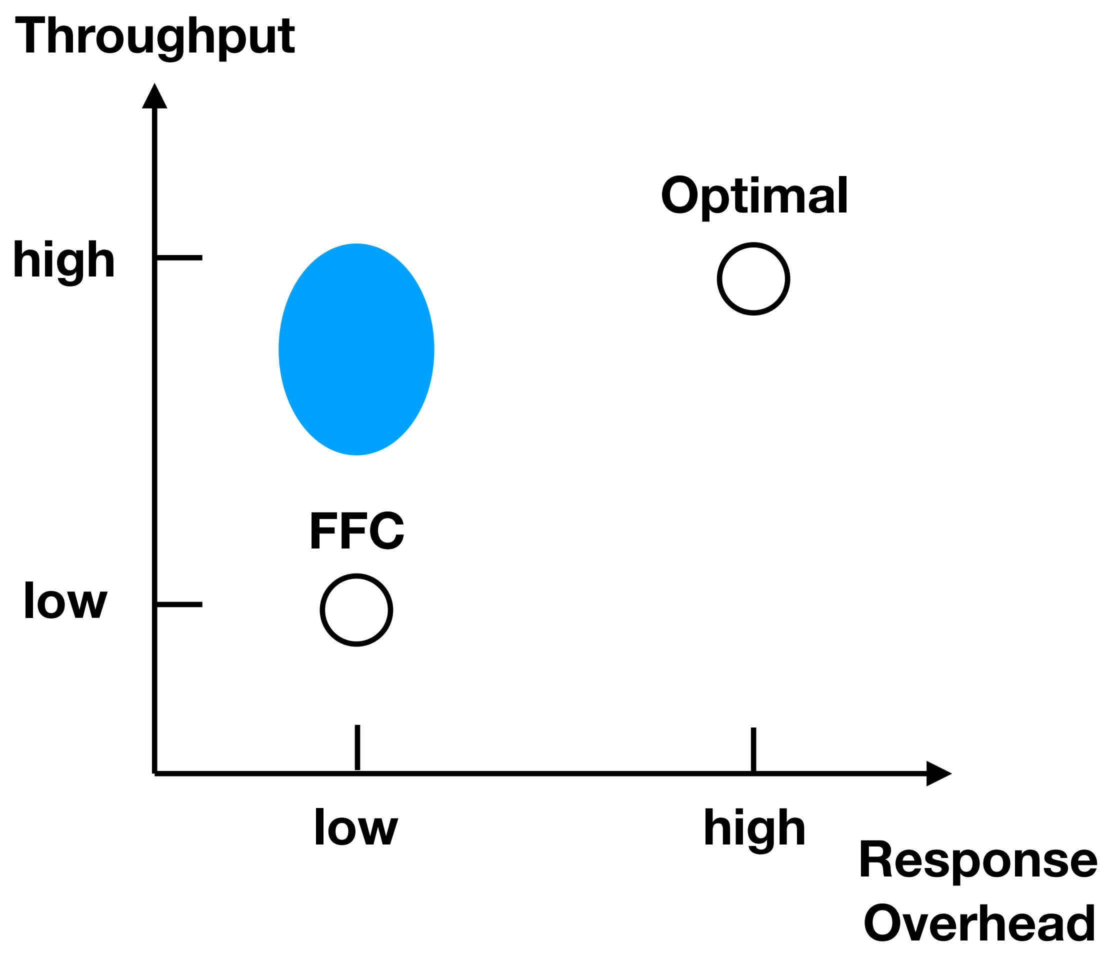
Bridge the gap !





Bridge the gap !





- Our goal is to design a new mechanism which sustains **high throughput** with **low response overhead** while providing **tractable failure analysis**.

● Desired area for new mechanisms

Contributions

- We show that existing congestion-free schemes perform much worse than optimal.
    - FFC's performance can be arbitrarily worse than optimal.
    - FFC's performance can degrade with an increase in the number of tunnels.
- We propose a set of novel mechanism called ***PCF (Provably Congestion-free and resilient Flexible routing)***.
    - PCF ensures the network is **provably congestion-free** under failures.
    - PCF performs **closer to the network's intrinsic capability**.

Contributions

- We show that existing congestion-free schemes perform much worse than o

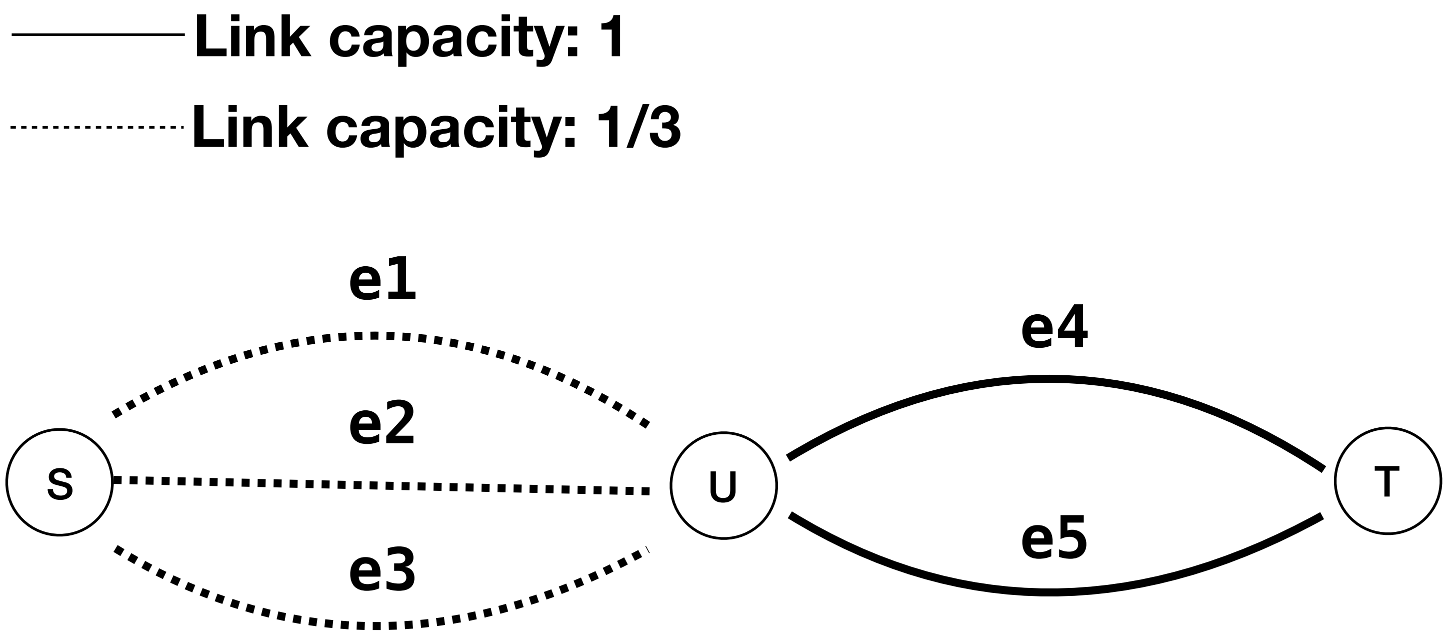
  f

  tunnels.

PCF's schemes can sustain higher throughput than FFC by a factor of **upto 1.5X** on average across the topologies, while providing a benefit of **2.6X** in some cases.

- We propose a set of novel mechanism called ***PCF (Provably Congestion-free and resilient Flexible routing)***.
  - PCF ensures the network is **provably congestion-free** under failures.
  - PCF performs **closer to the network's intrinsic capability**.

Example - Topology overview



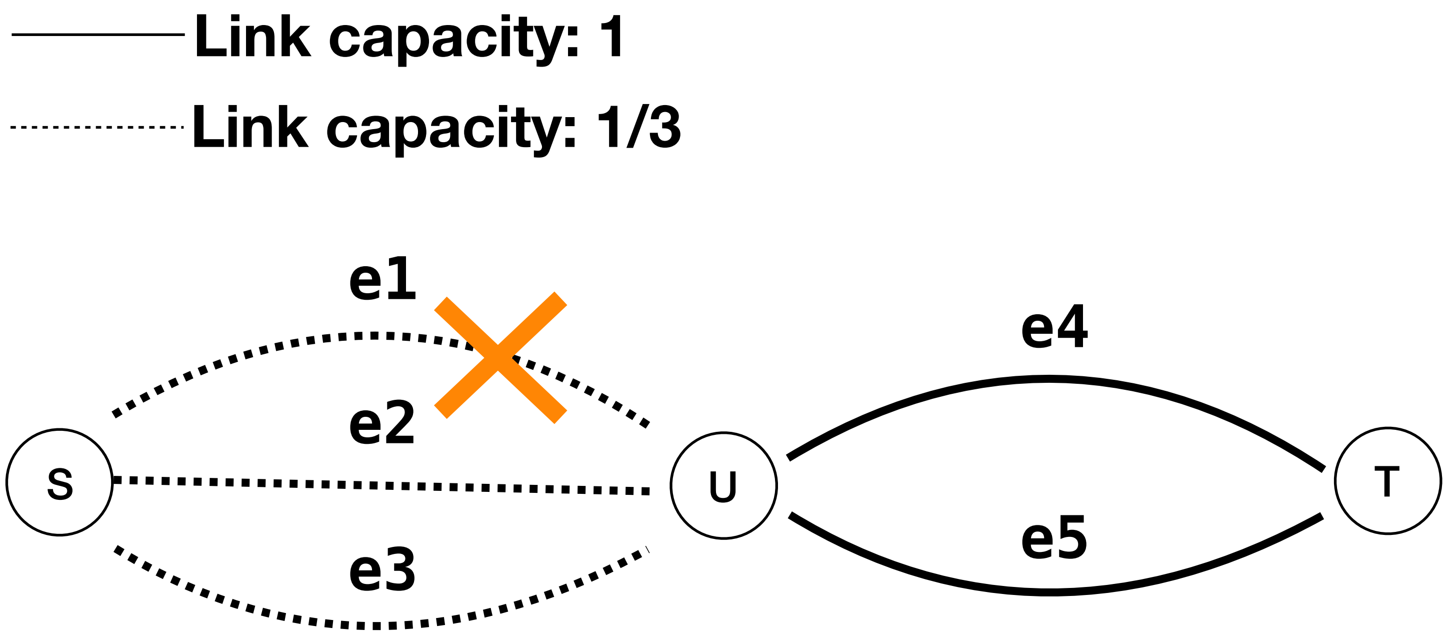
── Link capacity: 1

┈┈ Link capacity: 1/3

Tunnels:

```
l1 – e1,e4
l2 – e1,e5
l3 – e2,e4
l4 – e2,e5
l5 – e3,e4
l6 – e3,e5
```

How well can the network perform?

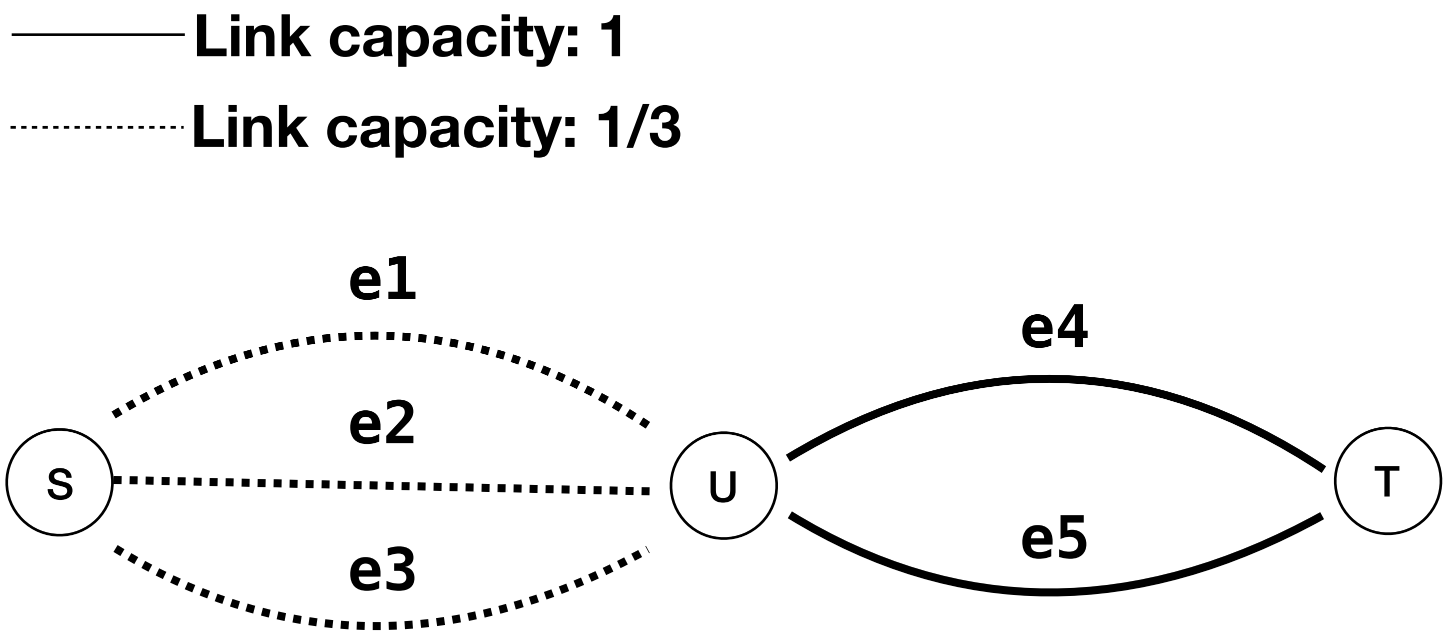


Tunnels:

l1 – e1,e4
l2 – e1,e5
l3 – e2,e4
l4 – e2,e5
l5 – e3,e4
l6 – e3,e5

- Single link failure
- Respond to failure optimally
- 2/3 unit of traffic can always be sent

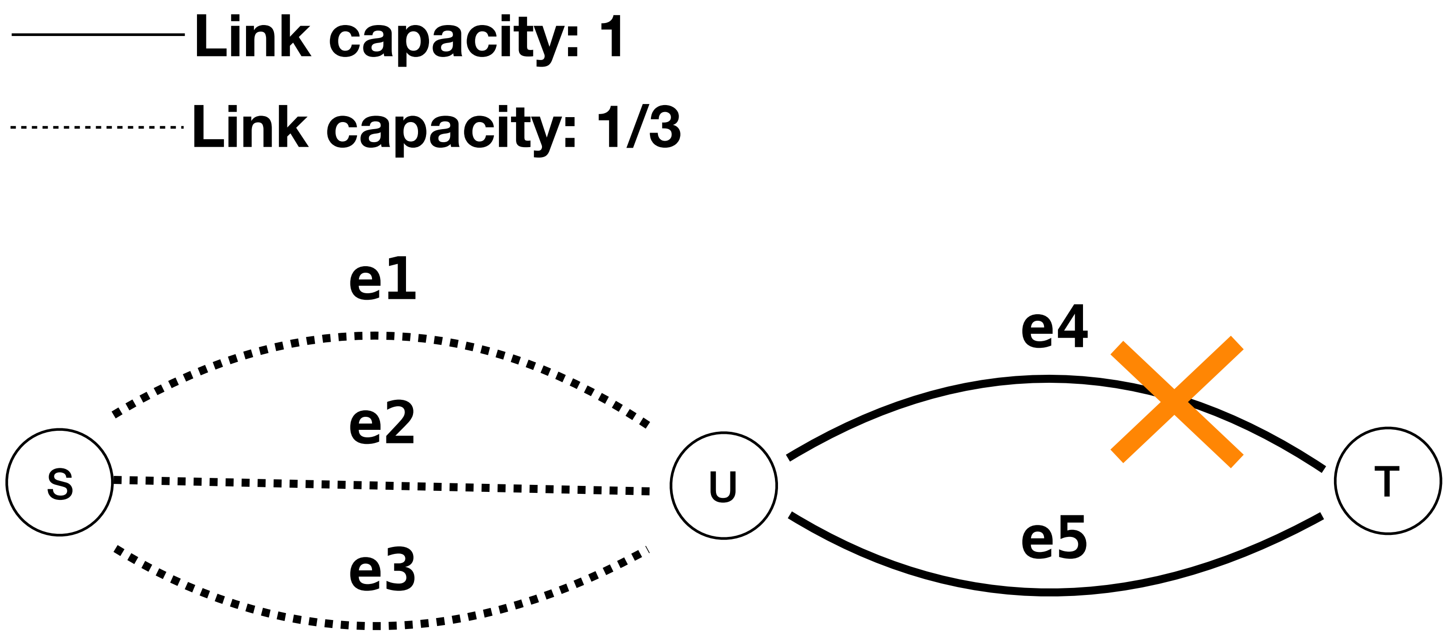
How well can FFC perform?



Reservation on tunnels:

```
l1 - e1,e4: 1/6
l2 - e1,e5: 1/6
l3 - e2,e4: 1/6
l4 - e2,e5: 1/6
l5 - e3,e4: 1/6
l6 - e3,e5: 1/6
```

How well can FFC perform?

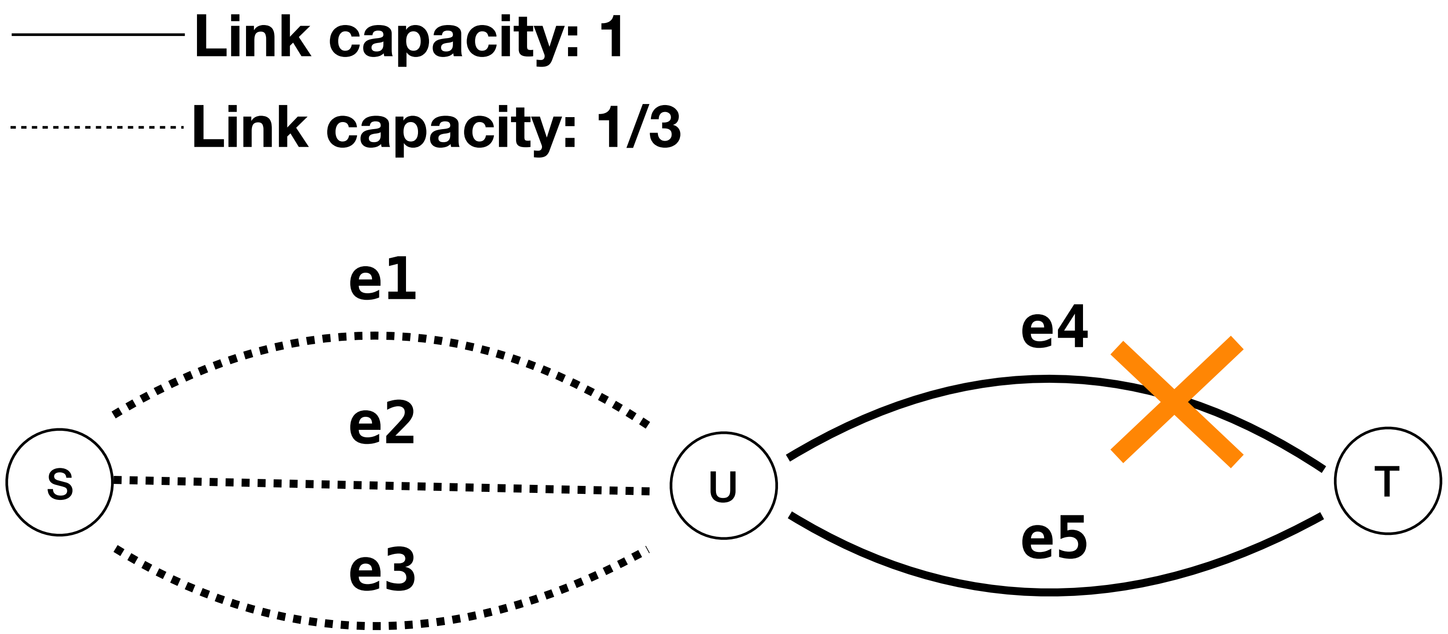


Reservation on tunnels:

~~l1 - e1,e4: 1/6~~
l2 - e1,e5: 1/6
~~l3 - e2,e4: 1/6~~
l4 - e2,e5: 1/6
~~l5 - e3,e4: 1/6~~
l6 - e3,e5: 1/6

Remaining tunnels can only carry 1/2 !

How well can FFC perform?


Link capacity: 1

Link capacity: 1/3

Reservation on tunnels:

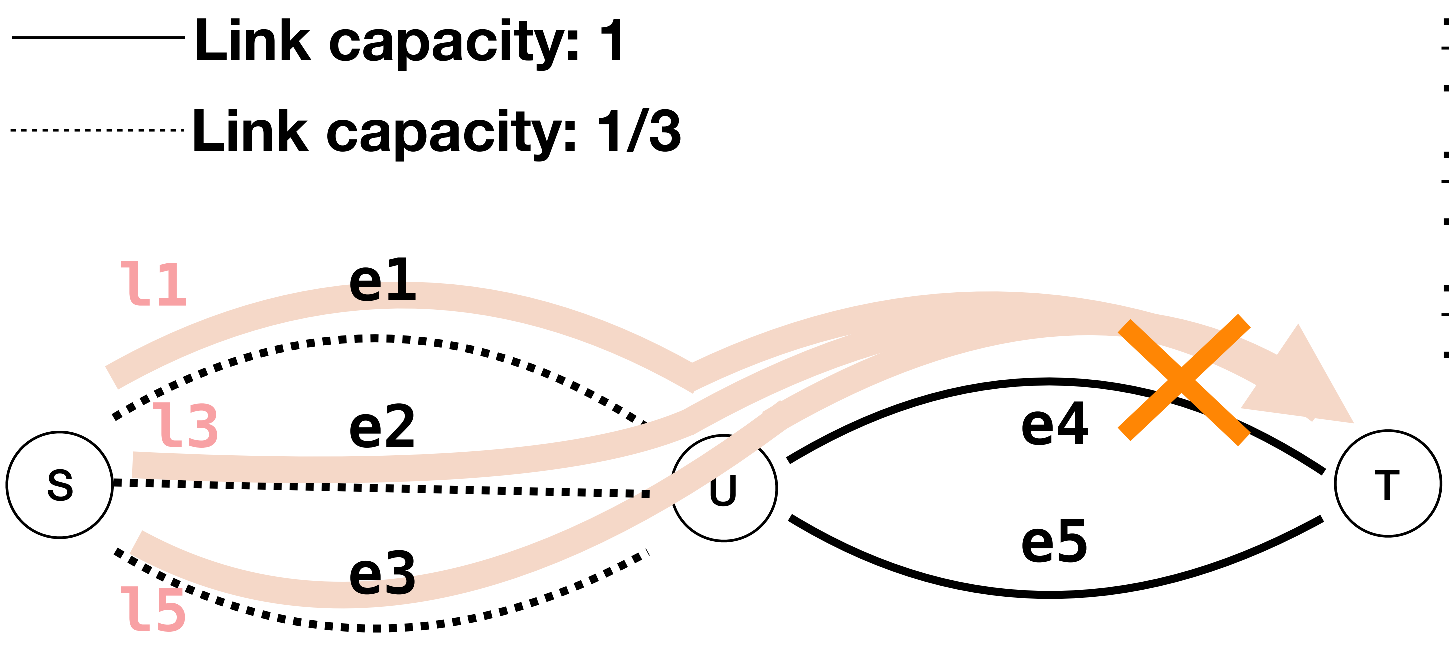
~~l1 - e1,e4: 1/6~~
l2 - e1,e5: 1/6
~~l3 - e2,e4: 1/6~~
l4 - e2,e5: 1/6
~~l5 - e3,e4: 1/6~~
l6 - e3,e5: 1/6

Remaining tunnels can only carry 1/2 !

FFC’s performance guarantee: 1/2

Optimal scheme: 2/3

Underlying reason



Reservation on tunnels:

~~l1 – e1,e4: 1/6~~
l2 – e1,e5: 1/6
~~l3 – e2,e4: 1/6~~
l4 – e2,e5: 1/6
~~l5 – e3,e4: 1/6~~
l6 – e3,e5: 1/6

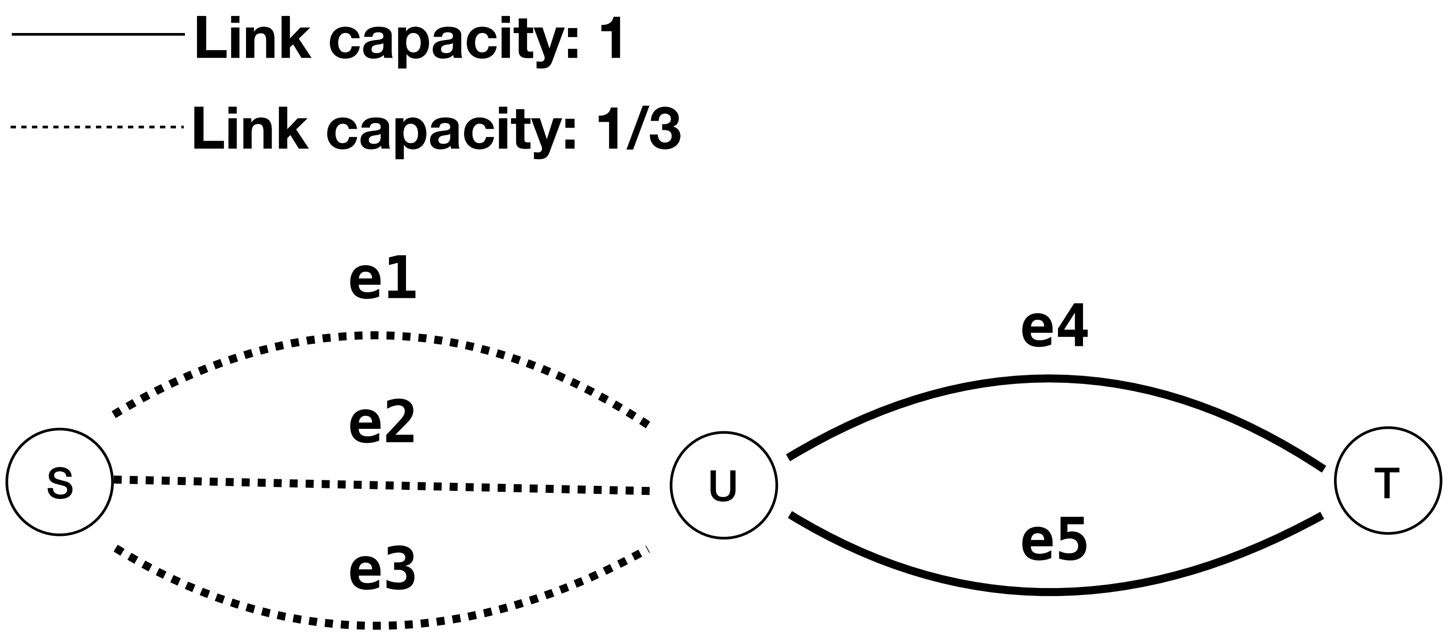
- FFC’s reservations are made at the granularity of entire tunnel.
  - e4 fails -> l1, l3, l5 fail -> reserved capacity on e1, e2, e3 is lost !
- PCF can solve this issue. For this example, it can achieve **optimal throughput**.

PCF’s solution

- FFC doesn’t provide enough flexibility in network response.
- Optimal mechanism has the most flexibility, but doesn’t provide tractable failure analysis.

- PCF carefully introduces flexibility in network response to simultaneously meet three objectives:
  - High throughput, tractable failure analysis, low response overhead
  - Introduce an abstraction called **logical sequence**

PCF's solution - Logical sequence



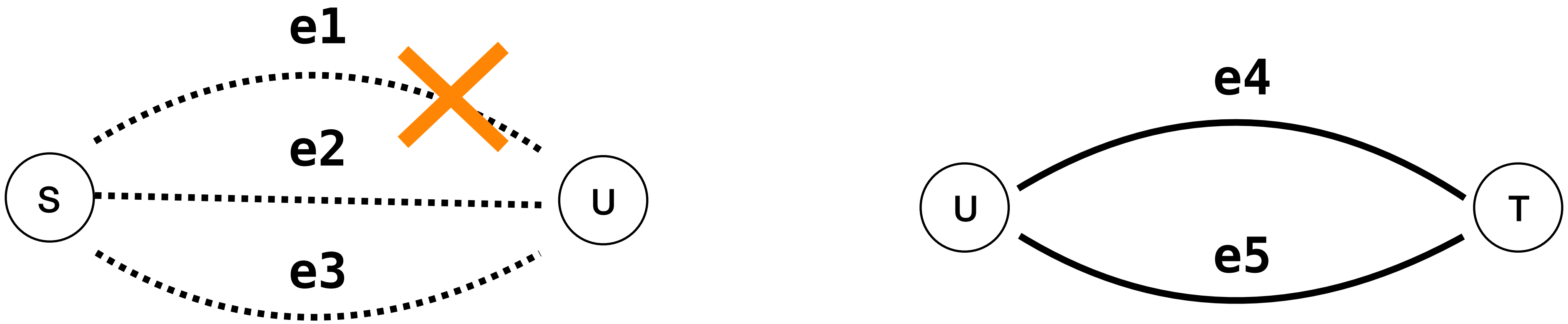
Tunnels:
l1 – e1
l2 – e2
l3 – e3
l4 – e4
l5 – e5

- **Logical sequence: S-U-T**
- Traffic is **independently** routed in the two segments (S-U and U-T) of the logical sequence.
- On each segment, we want to make reservation to ensure that it works upon failures.

PCF’s solution - Logical sequence

— Link capacity: 1

┈ Link capacity: 1/3

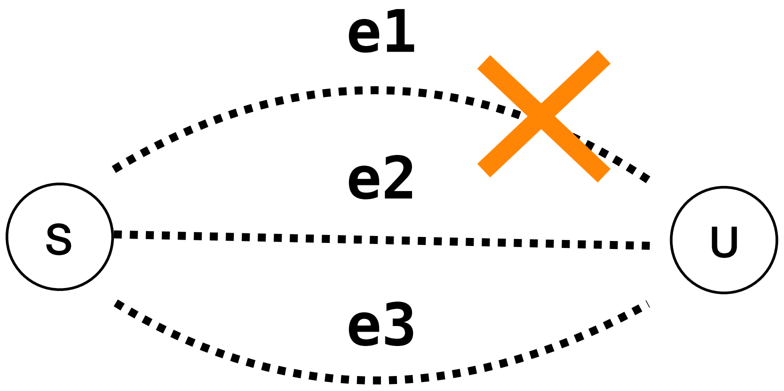


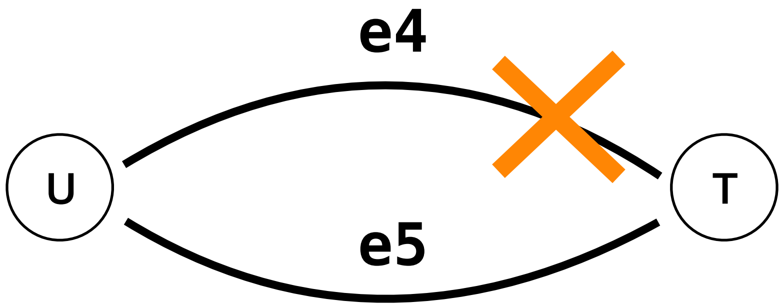
2/3 unit of traffic can be sent under single link failure.

PCF’s solution - Logical sequence

── **Link capacity: 1**

┄┄ **Link capacity: 1/3**

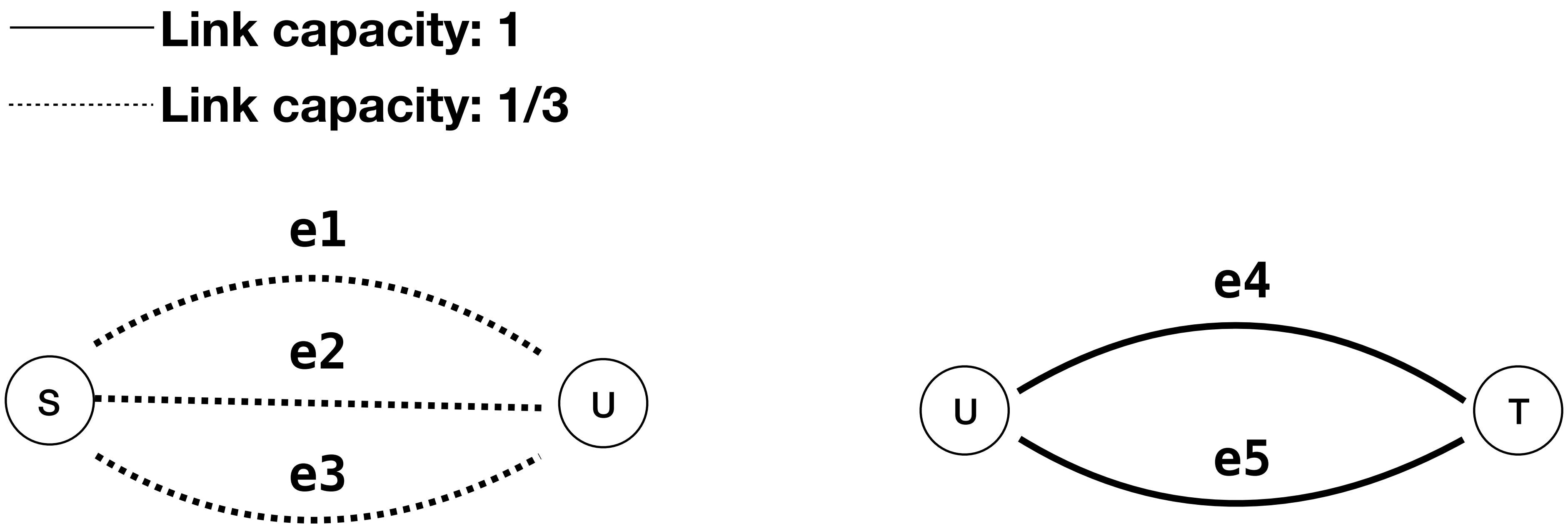




2/3 unit of traffic can be sent under single link failure.

1 unit of traffic can be sent under single link failure.

PCF’s solution - Logical sequence

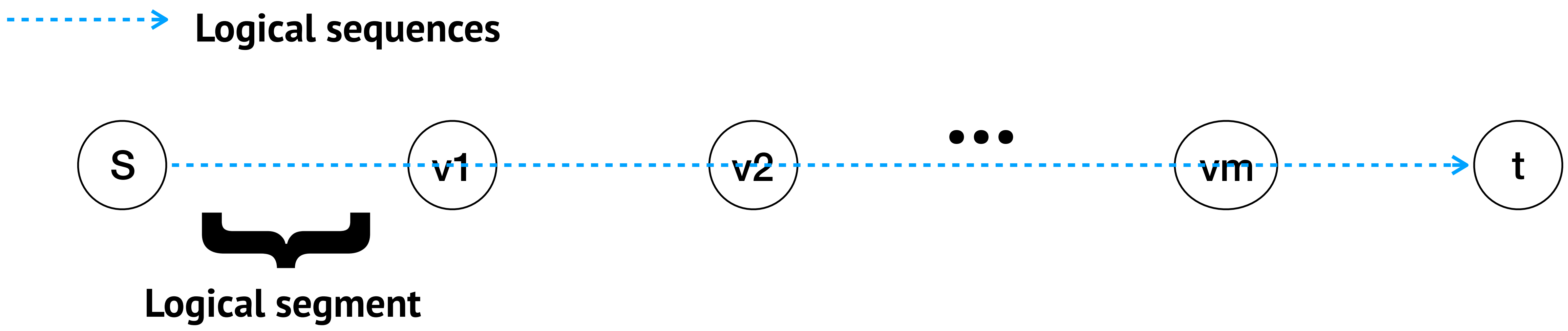


2/3 unit of traffic can be sent under single link failure.

1 unit of traffic can be sent under single link failure.

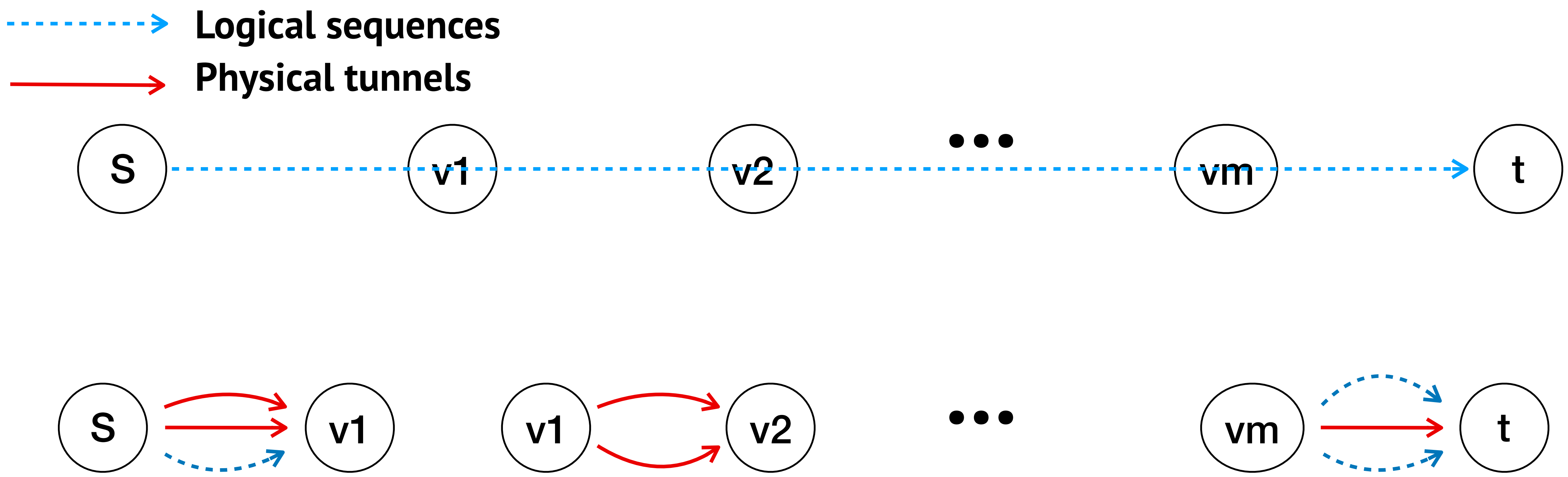
We can reserve 2/3 unit on the logical sequence S-U-T.
This reservation is always available under single link failure.
Performance guarantee: 2/3 (optimal)

PCF’s solution - Logical sequence



- Logical sequence: a sequence of nodes from s to t
- Logical hops: s, v1, v2, v3,…,vm, t
- Logical segments: s-v1, v1-v2, v2-v3, …, vm-t
- Traffic needs to traverse the logical hops.
- Logical hops don’t require direct link between them.

PCF’s solution - Logical sequence



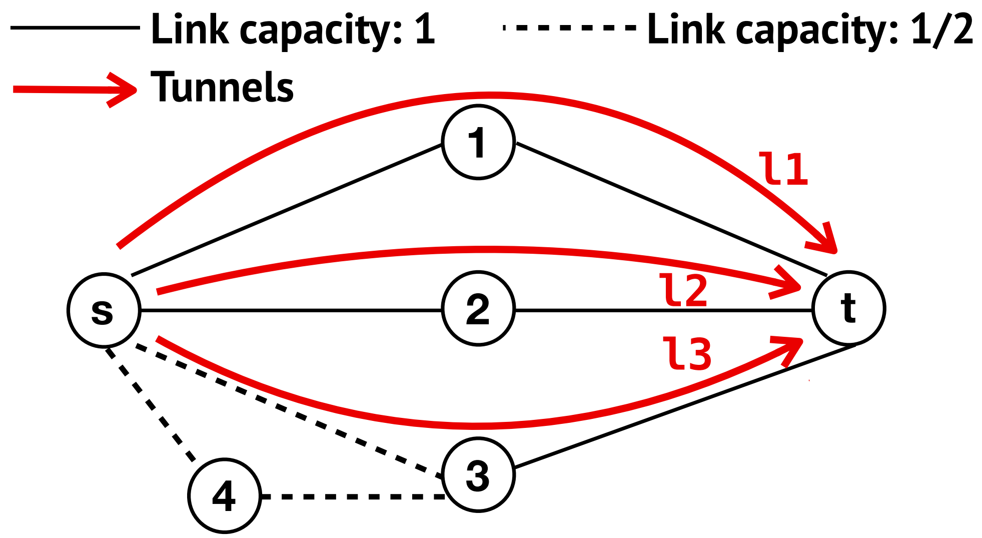
- Reserve on s-v1, v1-v2, v2-v3, …, vm-t independently.
- The reservation can be made on underlying physical tunnels or other logical sequences.
- We also consider **conditional logical sequence** which is only active under certain conditions (e.g. a set of links fail).

Logical sequence - model

- Goal: Determine the reservation on each physical tunnel and logical sequence

- Objective: Maximize allocated throughput
- Constraints:
  - Link capacity constraints
  - For any node pair s-t, and under any failure scenario
    - ensure sufficient reservation on physical tunnels and logical sequences from s to t
    - to sustain the throughput from s to t, and other logical sequences.

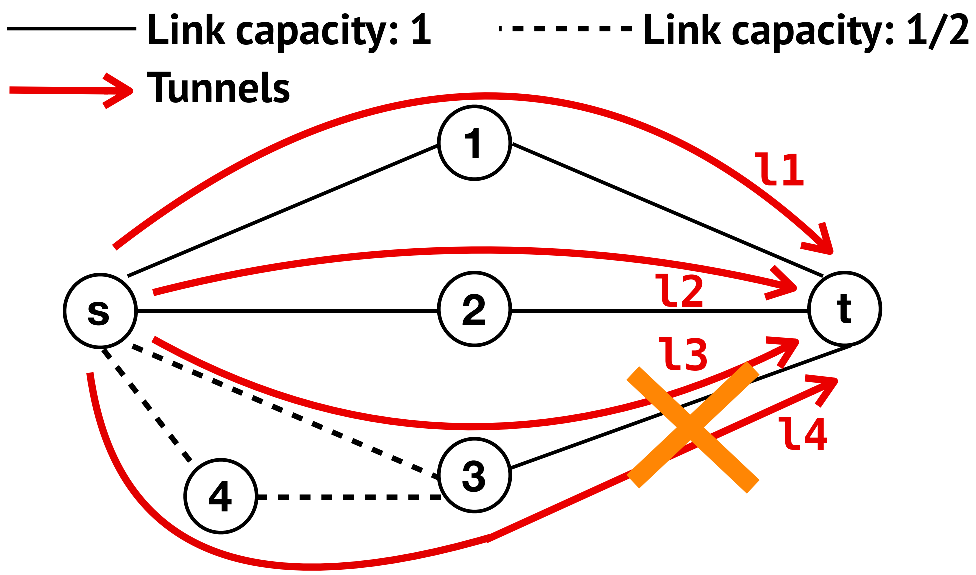
FFC - can deteriorate with more tunnels



Provided tunnels	Maximum Number of tunnels sharing a common link	Estimated number of tunnel failures under single link failure
l1, l2, l3	1	1

- FFC estimates the maximum number of tunnel failures, then considers all combinations of so many tunnel failures.

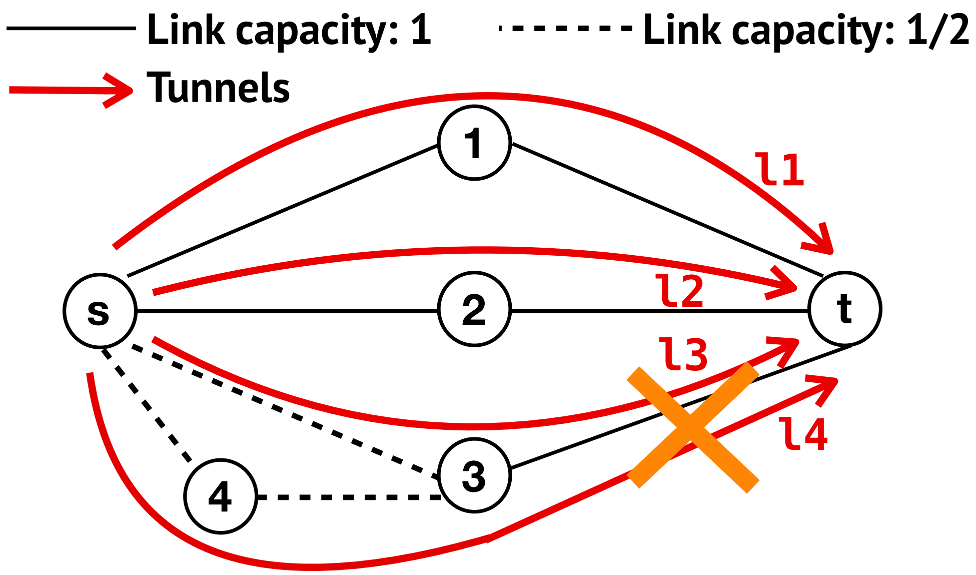
FFC - can deteriorate with more tunnels



Provided tunnels	Maximum Number of tunnels sharing a common link	Estimated number of tunnel failures under single link failure
l1, l2, l3, l4	2	2

- With 4 tunnels, FFC plans for all 2 tunnel failures, including failing l1 and l2 at the same time.
- If l1 and l2 die at the same time, which will **never occur under single link failure**, the performance will be very low.
- **Providing more tunnels to FFC may hurt the performance!**

FFC - can deteriorate with more tunnels



Provided tunnels	Maximum Number of tunnels sharing a common link	Estimated number of tunnel failures under single link failure
l1, l2, l3, l4	2	2

- With 4 tunnels, FFC plans for all 2 tunnel failures including failing l1

- If
si

PCF solves this issue by modeling the fact that when one link fails, l1 and l2 can not die at the same time.

- **Providing more tunnels to FFC may hurt the performance!**

Theoretical results

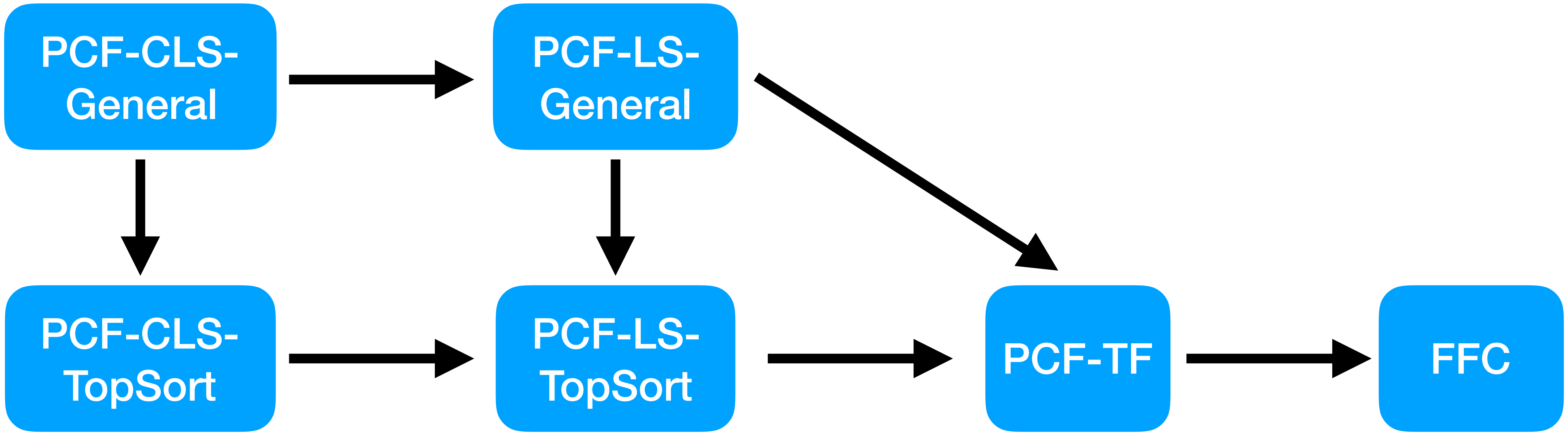
- Proposition: PCF’s performance **does not degrade with additional tunnels**, and performs **at least as well as FFC**.

- Proposition: There exist topologies for which (i) FFC’s throughput is **arbitrarily worse** than optimal even when **exponentially many** tunnels are used; and (ii) PCF’s throughput achieves the **optimal** with only **polynomially many** tunnels.

PCF - implementation

- When the logical sequences do not recursively depend on each other (satisfy a topological order):
    - Local proportional routing mechanism can be used.
    - Redistribute traffic on the active tunnels and logical sequences.
- In more general cases:
    - Use centralized controller to solve a linear system upon each failure
    - Solving a linear system is much easier than solving an optimization problem.
    - Amenable to distributed implementation in the future.

PCF - family of schemes



PCF-CLS-General

PCF-LS-General

PCF-CLS-TopSort

PCF-LS-TopSort

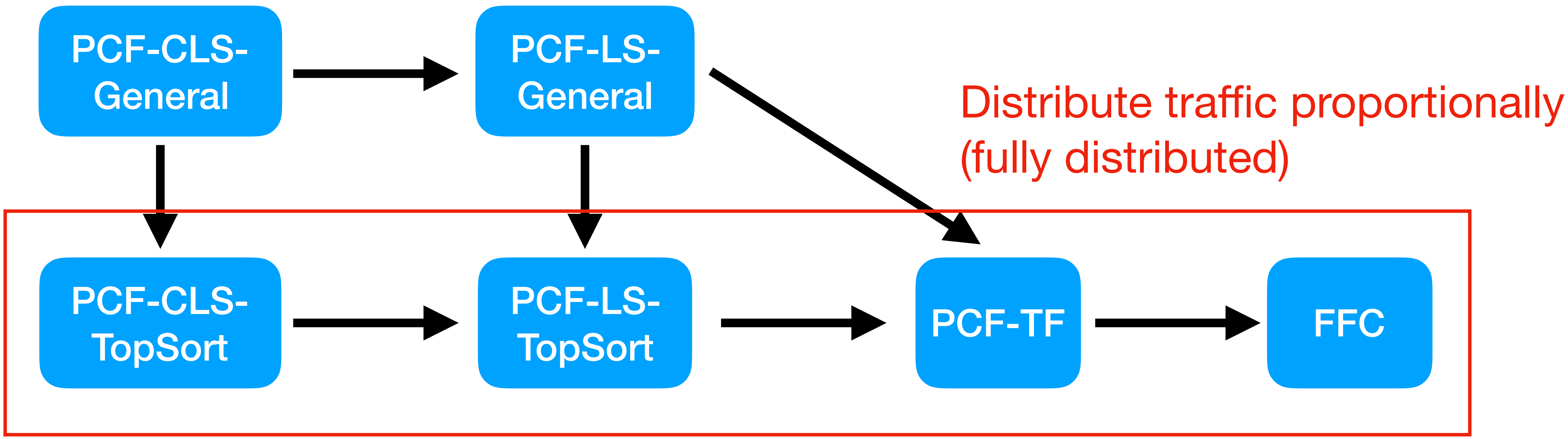
PCF-TF

FFC

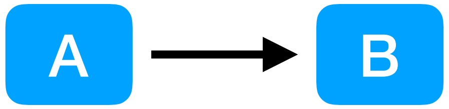
A → B A is provably better than B

All PCF schemes are associated with tractable models that guarantee the network is congestion-free under failures.

PCF - family of schemes



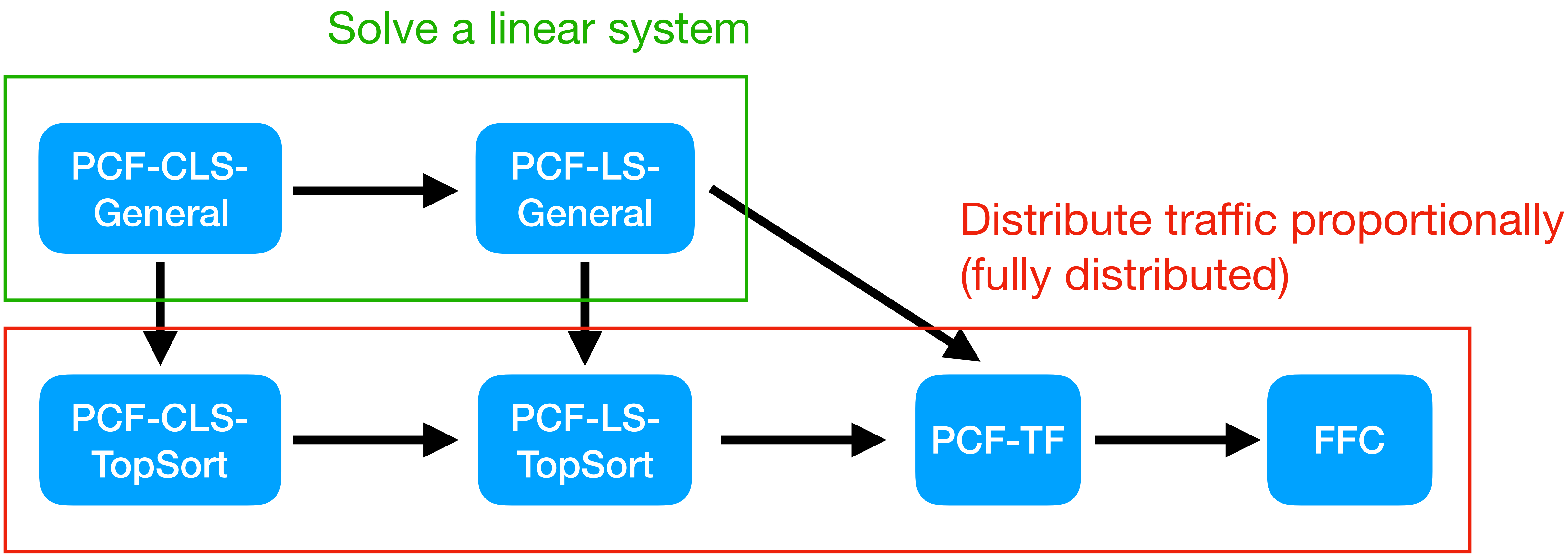
Distribute traffic proportionally (fully distributed)



A is provably better than B

All PCF schemes are associated with tractable models that guarantee the network is congestion-free under failures.

PCF - family of schemes



Solve a linear system

PCF-CLS-General

PCF-LS-General

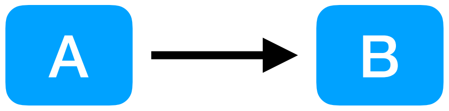
Distribute traffic proportionally (fully distributed)

PCF-CLS-TopSort

PCF-LS-TopSort

PCF-TF

FFC



A is provably better than B

All PCF schemes are associated with tractable models that guarantee the network is congestion-free under failures.

Evaluation - instantiating logical sequences

- PCF-LS - We chose topologically sorted sequences by using shortest paths.
- PCF-CLS - We additionally added sequences that are activated on the failure of a link.

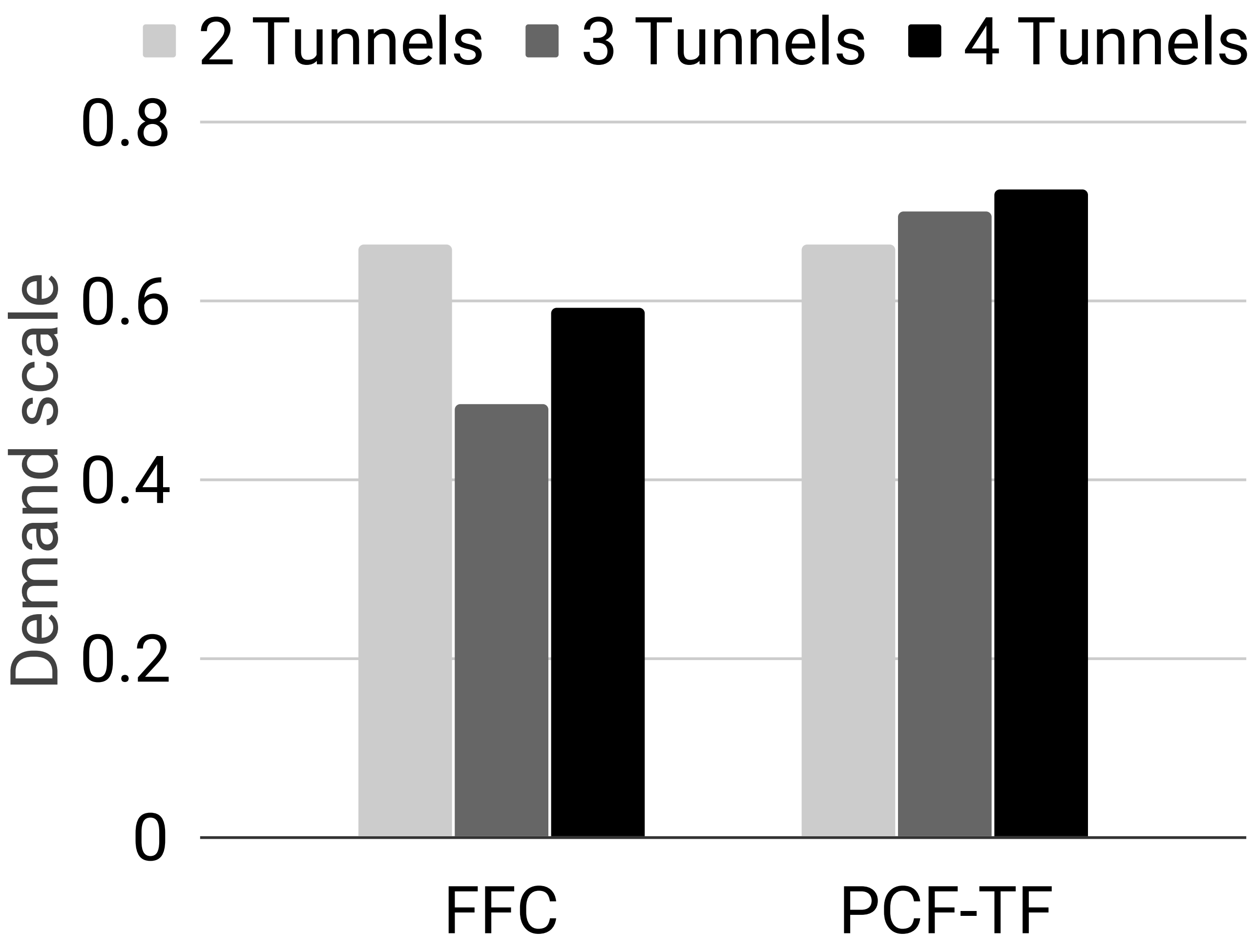
Evaluation - setup

- Physical tunnels: as disjoint as possible
- 21 topologies (the largest topology has 151 links)
- Traffic matrix: gravity model
- Metric: demand scale (the factor by which the traffic demand of all pairs can be scaled)

Benefits of the better failure model

Deltacom topology, single link failure

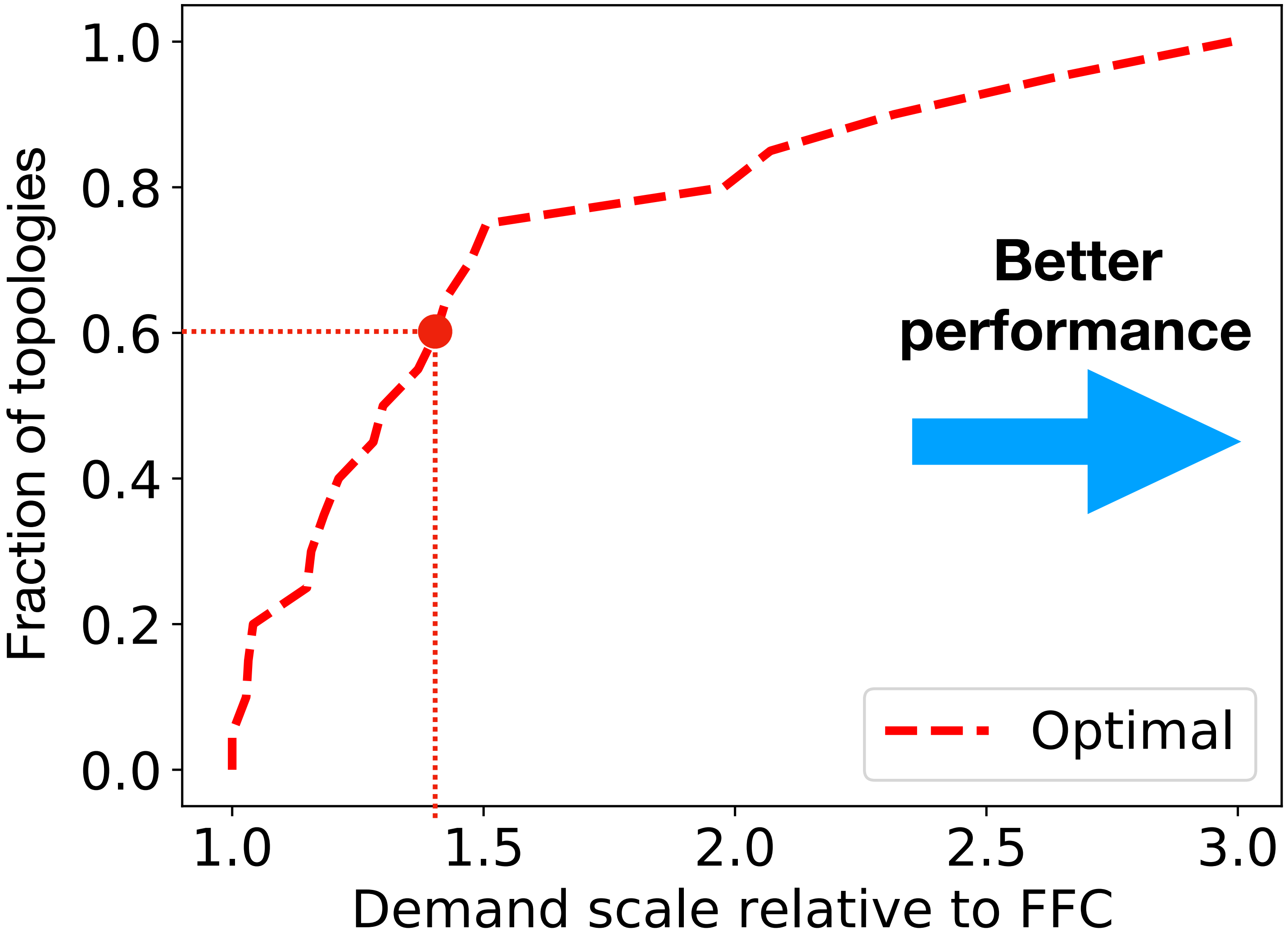
- FFC’s performance is worse with 3 and 4 tunnels than with only 2 tunnels.
- PCF performs better as tunnels are added.



PCF vs. FFC on multiple topologies

21 topologies, up to 3 link failures

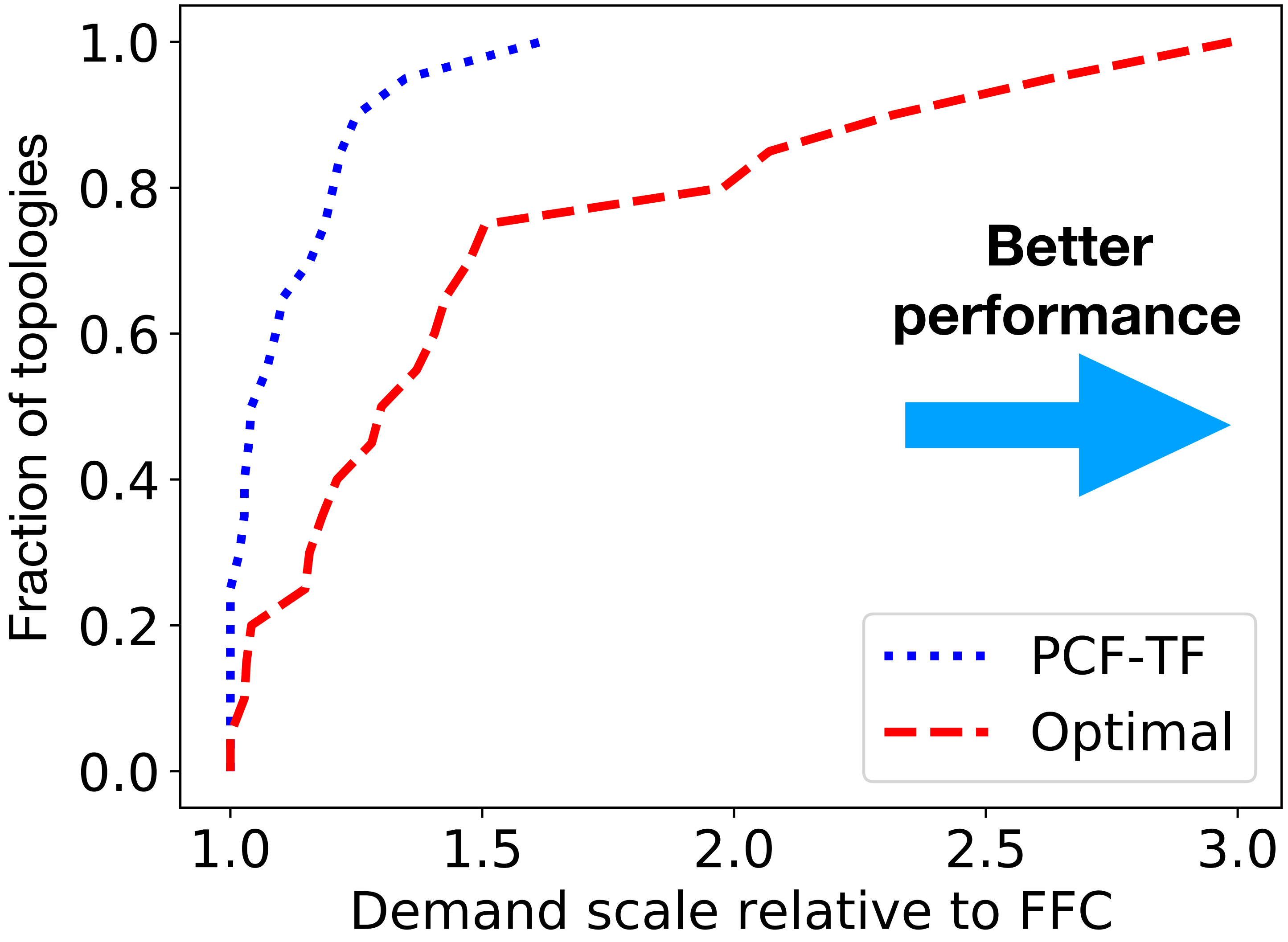
- Optimal scheme gives much higher throughput than FFC.
- For 40% of the topologies, the optimal scheme can sustain 40% more demand than FFC.



PCF vs. FFC on multiple topologies

21 topologies, up to 3 link failures

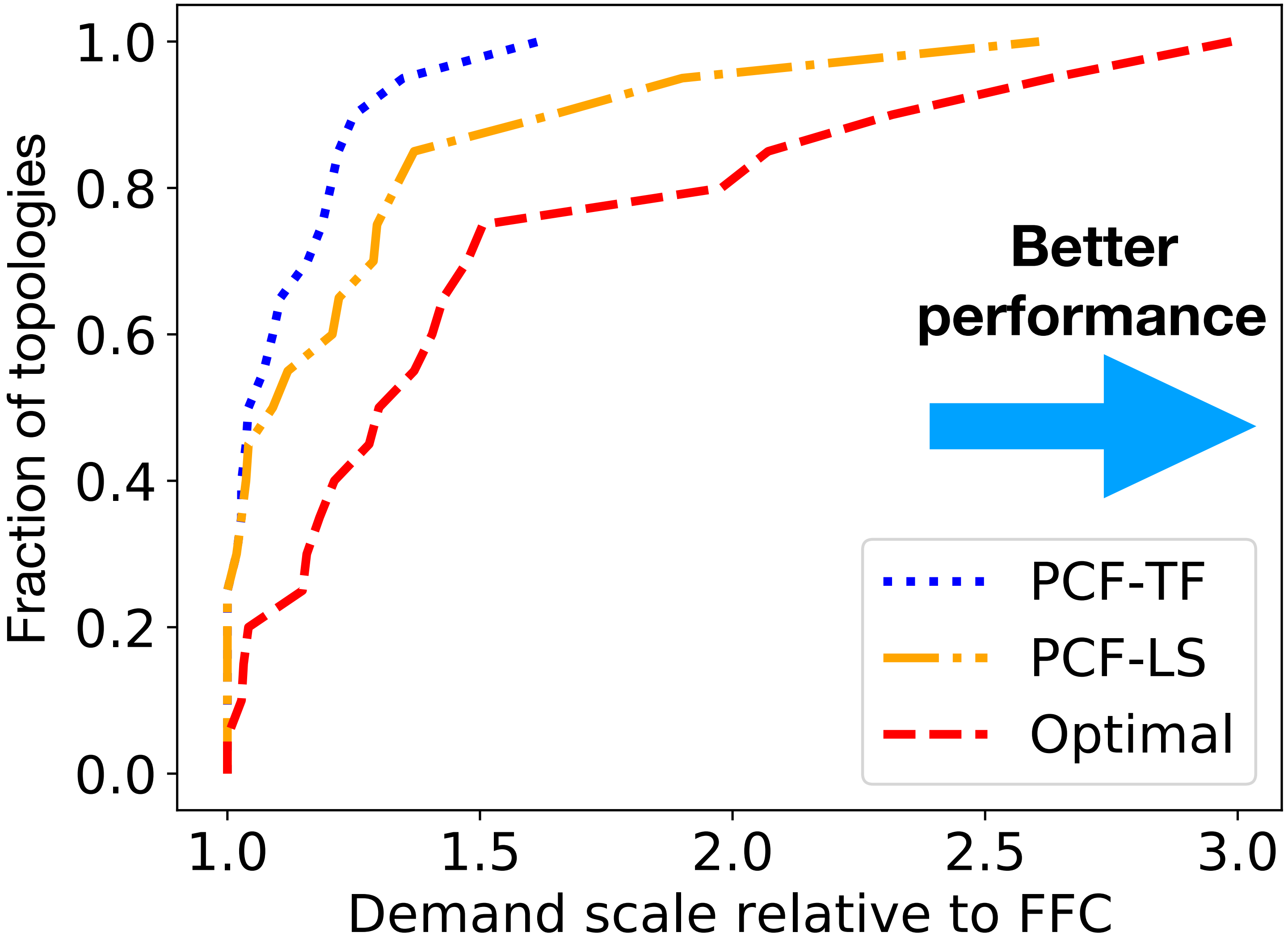
- PCF-TF improves over FFC by 11% on average and more than 50% in the best case.
- PCF-TF has the same response mechanism as FFC.



PCF vs. FFC on multiple topologies

21 topologies, up to 3 link failures

- PCF-LS improves over FFC by 25% on average, and performs 2.6x better in the best case.
- Fully distributed response mechanism

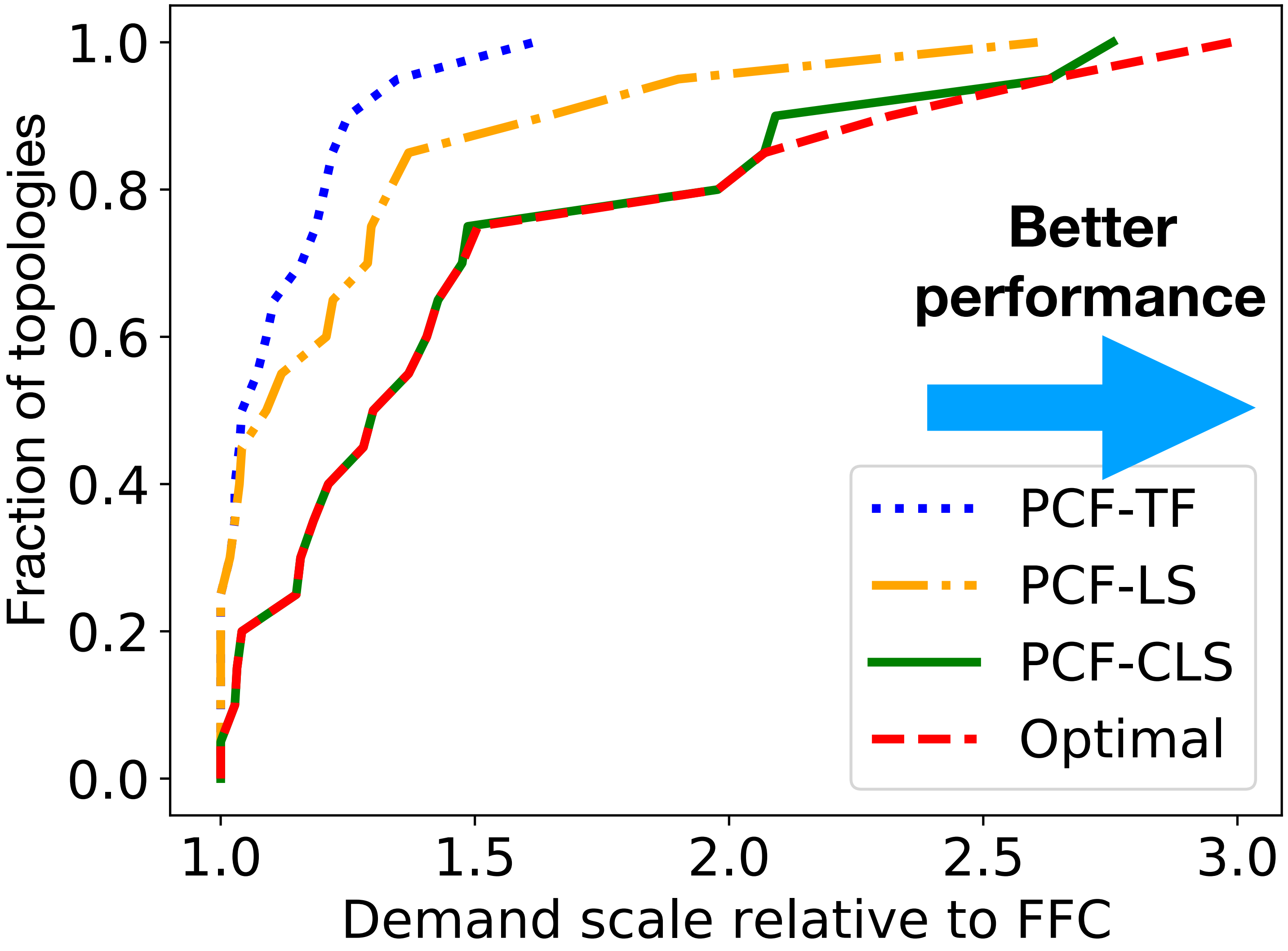


Better performance

PCF vs. FFC on multiple topologies

21 topologies, up to 3 link failures

- PCF-CLS improves over FFC by 50% on average, and matches the optimal for most cases.
- Only require linear system on failures



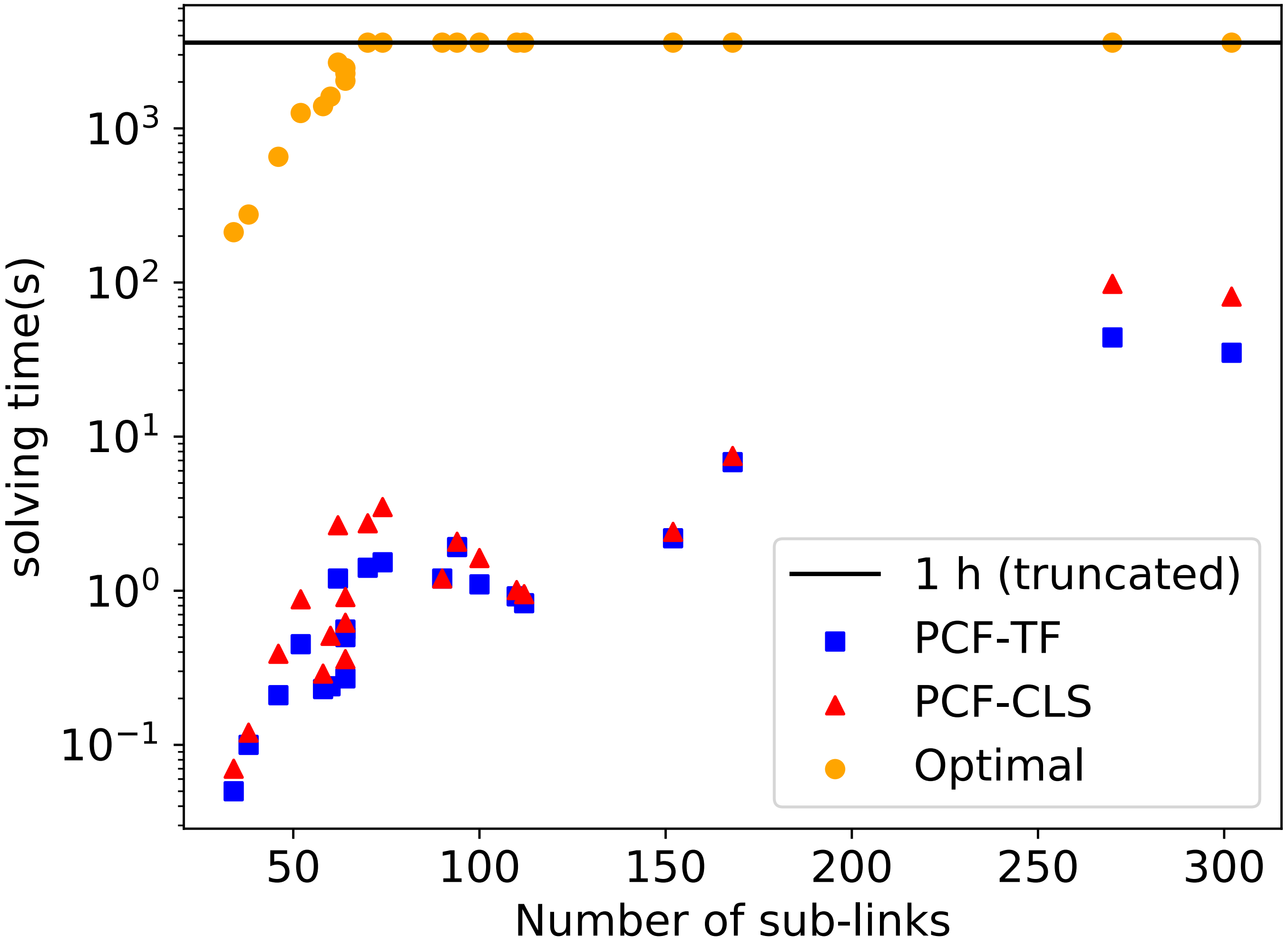
Other results

- Similar improvement over FFC are observed in other experiments
  - Evaluate on same topology over multiple different demands
  - Evaluate on other metric instead of demand scale
- An interesting heuristic shows feasibility of achieving nearly optimal performance for most topologies with completely local routing under single link failure.

Solving time

21 topologies, up to 3 link failures

- PCF schemes:
  - For most topologies, the solving times are under 10 seconds.
  - For the largest topology (302 links), the solving time is under 100 seconds.
- Optimal scheme:
  - Does not finish within one hour for many topologies.
  - For the largest topology, it took days to finish.



Conclusion

- We show that existing congestion-free schemes perform much worse than the network's intrinsic capability. We present the underlying reasons.
- We propose PCF in order to bridge the gap.
    - Carefully introduce flexibility in network response to achieve:
        - High throughput, tractable failure analysis, low response overhead
    - Formal results show that PCF is provably better than FFC.
    - PCF achieves up to 50% improvement over FFC on average across 21 topologies.

Thanks!

Chuan Jiang: jiang486@purdue.edu
Sanjay Rao: sanjay@ecn.purdue.edu