

Optical Topology Programming

Foundations, Measurements, and Applications

“Dr. Matthew Nance-Hall”!!



Introduction



A world map with numerous colored lines representing submarine cable networks. The lines are most dense in Europe, North America, and Asia. Labels for various countries and regions are visible, including Canada, Greenland, Iceland, Norway, Finland, Sweden, Russia, Poland, Germany, France, Spain, Italy, Turkey, India, and others. The map also shows major bodies of water like the Indian Ocean and the Pacific Ocean.

Networks

The backbone of the Internet

Reliance on networked applications is rising.

Web search, GPS navigation, video streaming, ride hailing, food delivery, telehealth, video conferencing.

The rise of artificial intelligence (AI) based applications is expected to accelerate demand further.

Large language models, autonomous vehicles, robotics, virtual assistants, medical diagnosis, etc.

In 2020, the average daily traffic demand world-wide was 2.5 quintillion, or 2.5 *billion billion* bytes.¹

From 2019 to 2023, estimated annual growth of traffic was 30%.²



Inter-network and Intra-network

Inter-network

- Traffic exchanged between one **autonomous** network and another.
- A home router handles inter-network traffic to/from a home and the Internet.
- AWS's routers handle video streaming traffic from Prime Video's servers to customers' homes.

Intra-network

- Concerns traffic exchanged within a single autonomous network.
- A campus network's **core routers and switches** handle the transfer of data within the organization.
- AWS's core routers and switches handle the traffic between AWS's data centers around the world.

Scope: Enterprise Intra-network Management

Traffic confined to a **single backbone network** managed by a single entity.

Owner manages the entire network stack

- Optical Fibers

- Routers and Switches

- Network Software Controller

Can be as large as a global cloud provider or as small as the UO campus network.





Trends in Enterprise Networks



Trend 1: Cloudification

The backbone of our digital lives



A **cloud** is a *private network* of data centers, distributed around the globe, that hosts services for their customers.

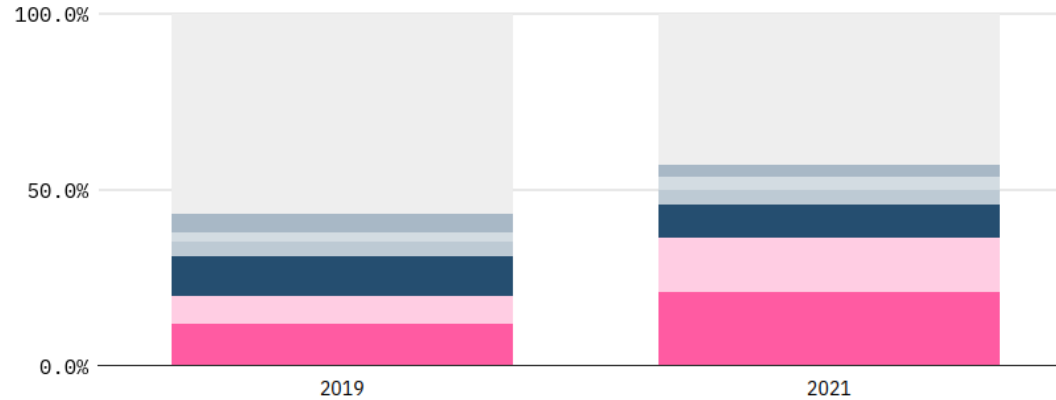
They host **popular applications**.
e.g., Netflix, Pinterest, Etsy, Uber,
or just about any website or app you can think of.

They also host **private services** for their customers.
e.g., supply chain management, fraud detection,
logistics, human resources and accounting,
or just about any business need you can think of.

Big Tech took up more than half of total network traffic last year

% of total network traffic in 2021 compared to 2019

● Google ● Facebook ● Netflix ● Apple ● Amazon ● Microsoft ● Other



Source: [Sandvine](#)

TECHMONITOR

This represented a 33% increase in total traffic from 2019, and the latest figures show that it is the first time where the total network traffic of these six companies was larger than all other service providers combined.



57%

More than half of all network traffic in 2021 was attributed to **six** entities.

All six represent services running on major cloud networks.



“We need another 1000x [capacity]
over the coming 20 years, but
we don't know how to do that.”

- Amin Vahdat, 2022
Vice President and General Manager
Machine Learning, Systems, and Cloud AI @ Google

Enterprise Network Trend 2



Evolving Threats

DDoS attacks

are inexpensive to launch.
cause significant loss of revenue.

These attacks cost small to medium sized enterprises ~ \$5,600/min [3].



BUY DDOS ATTACK 4 HOURS

★★★★★ (29 customer reviews)

\$99.00

DDoS attack Per Hour

1

Category: [Hacking Services](#) Tags: [buy DDoS attack](#), [ddos attack bot](#), [ddos attack botnet](#), [ddos attack buy](#), [ddos attack cost](#), [ddos attack definition](#), [ddos attack discord](#), [report a ddos attack](#), [stop a ddos attack](#)

All your Internet connections will be taken offline by massive Distributed Denials of Service flooding starting March 30 if you don't pay protection fee - exactly 0.50 Bitcoins (roughly \$600.00) to the address "1EMJ3L2o4exgpD1pxNPf2HxxHKBTf3MDJC"

Summary of Trends

Trend 1

Network applications are driving demand on backbone enterprise networks.

Trend 2

This threat looms over any enterprise network, adding a compounding factor to the "normal" traffic demand that the network should satisfy.



State of the Art



Traffic Engineering

Decentralized cloud wide-area network traffic engineering with BLASTSHIELD

Umesh Krishnaswamy Rachee Singh Nikolaj Bjørner Himanshu Raj
Microsoft

Semi-Oblivious Traffic Engineering: The Road Not Taken

Praveen Kumar Yang Yuan Chris Yu Nate Foster Robert Kleinberg
Cornell Cornell CMU Cornell Cornell
Petr Lapukhov Chium Lin Lim Robert Soulé
Facebook Facebook Università della Svizzera italiana

Intrusion Detection

Machine Learning Based Intrusion Detection System for Software Defined Networks

Atiku Abubakar and Bernard Pranggono*
Department of Engineering and Mathematics,
Sheffield Hallam University, Sheffield, S1 1WB, U.K.
*B.Pranggono@shu.ac.uk

Traffic Scrubbing

Protocol Scrubbing: Network Security Through Transparent Flow Modification

David Watson, Matthew Smart, G. Robert Malan, Member, IEEE, and Farnam Jahani, Member, IEEE

Poseidon: Mitigating Volumetric DDoS Attacks with Programmable Switches

Menghao Zhang*, Guangyu Li*, Shicheng Wang*, Chang Liu*, Ang Chen†, Hongxin Hu*, Guofei Gu*, Qi Li*, Mingwei Xu*, Jianping Wu*

*Institute for Network Sciences and Cyberspace & Department of Computer Science and Technology, Tsinghua University
†Beijing National Research Center for Information Science and Technology (BNRIST)
‡Rice University §School of Computing, Clemon University ¶SUCCESS Lab, Texas A&M University

Software Defined Everything*

Bringing programmability into the network.

Load Balancing

A High-Speed Load-Balancer Design with Guaranteed Per-Connection-Consistency

Tom Barbette Chen Tang Haoran Yao Dejan Kostić
Gerald Q. Maguire Jr. Panagiotis Papadimitratos Marco Chiesa
KTH Royal Institute of Technology

Firewalls

Language-Independent Synthesis of Firewall Policies

Chiara Bodei, Pierpaolo Degano, Letterio Galletta Riccardo Focardi, Mauro Tempesta, Lorenzo Veronese
Dipartimento di Informatica, Università di Pisa, Italy DAIS, Università Ca' Foscari Venezia, Italy
{chiara.degano,galletta}@di.unipi.it {focardi,tempesta}@unive.it, 852058@stud.unive.it

Limitations Based on Implicit Assumptions

Today's best practices in networks use programmability to scale traffic management and network defense.

Limitations Based on Implicit Assumptions

Today's best practices in networks use programmability to scale traffic management and network defense.

Key observation:

These change *the forwarding behavior only* and ignore the topological behavior of the network completely.

Limitations Based on Implicit Assumptions

Today's best practices in networks use programmability to scale traffic management and network defense.

Key observation:

These change *the forwarding behavior only* and ignore the topological behavior of the network completely.

What is critically lacking is a framework to change the topological and forwarding behavior to enable more agile and robust services.

Limitations Based on Implicit Assumptions

Today's best practices in networks use programmability to scale traffic management and network defense.

Key observation:

These change *the forwarding behavior only* and ignore the topological behavior of the network completely.

What is critically lacking is a framework to change the topological and forwarding behavior to enable more agile and robust services.

Why are we here?

Limitations Based on Implicit Assumptions

Today's best practices in networks use programmability to scale traffic management and network defense.

Key observation:

These change *the forwarding behavior only* and ignore the topological behavior of the network completely.

What is critically lacking is a framework to change the topological and forwarding behavior to enable more agile and robust services.

Why are we here?

1. Operator's mindset: "The benefit of dynamically changing topology is unclear."

Limitations Based on Implicit Assumptions

Today's best practices in networks use programmability to scale traffic management and network defense.

Key observation:

These change *the forwarding behavior only* and ignore the topological behavior of the network completely.

What is critically lacking is a framework to change the topological and forwarding behavior to enable more agile and robust services.

Why are we here?

1. **Operator's mindset:** "The benefit of dynamically changing topology is unclear."
2. **Academic's mindset:** "Optimizing routing+topology is NP-hard, not worth the effort."

Limitations Based on Implicit Assumptions

Today's best practices in networks use programmability to scale traffic management and network defense.

Key observation:

These change *the forwarding behavior only* and ignore the topological behavior of the network completely.

What is critically lacking is a framework to change the topological and forwarding behavior to enable more agile and robust services.

Why are we here?

1. **Operator's mindset:** "The benefit of dynamically changing topology is unclear."
2. **Academic's mindset:** "Optimizing routing+topology is NP-hard, not worth the effort."
3. **Common denominator:** "This is infeasible, if it was possible we'd be doing it already."

Limitations Based on Implicit Assumptions

Today's best practices in networks use programmability to scale traffic management and network defense.

Key observation:

These change *the forwarding behavior only* and ignore the topological behavior of the network completely.

What is critically lacking is a framework to change the topological and forwarding behavior to enable more agile and robust services.

Why are we here?

1. **Operator's mindset:** "The benefit of dynamically changing topology is unclear."
2. **Academic's mindset:** "Optimizing routing+topology is NP-hard, not worth the effort."
3. **Common denominator:** "This is infeasible, if it was possible we'd be doing it already."

This thinking has carried over for the last 20 years. While networks have been "getting by" without considering topology, modern applications & threats compell us to revisit these assumptions.

Problem 1: “A Dynamic Topology is Infeasible”

The Optical-Packet Network Chasm

Optical and packet networks have evolved separately.

Exciting work is happening in both domains, but these developments rarely overlap.

Optical networking researchers and operators don't think applications need the optical network to be programmable.

Networking researchers assume that optical links need tens of minutes or hours to be brought online.



Problem 2: “Adapting Topology to suit Traffic is Impractical”

Jointly Optimizing Routing & Topology

Input: A set of *short-term* demands between different network endpoints.

Number of connected labeled graphs with n nodes.

n	$a(n)$
0	1
1	1
2	1
3	4
4	38
5	728
6	26704
7	1866256
8	251548592
9	66296291072
10	34496488594816
11	35641657548953344
12	73354596206766622208
13	301272202649664088951808
14	2471648811030443735290891264
15	40527680937730480234609755344896
16	1328578958335783201008338986845427712

Source: <https://oeis.org/A001187>

Problem 2: “Adapting Topology to suit Traffic is Impractical”

Jointly Optimizing Routing & Topology

short-term: a time horizon of 5 minutes or less.

Input: A set of *short-term* demands between different network endpoints.

Number of connected labeled graphs with n nodes.

n	$a(n)$
0	1
1	1
2	1
3	4
4	38
5	728
6	26704
7	1866256
8	251548592
9	66296291072
10	34496488594816
11	35641657548953344
12	73354596206766622208
13	301272202649664088951808
14	2471648811030443735290891264
15	40527680937730480234609755344896
16	1328578958335783201008338986845427712

Source: <https://oeis.org/A001187>

Problem 2: “Adapting Topology to suit Traffic is Impractical”

Jointly Optimizing Routing & Topology

short-term: a time horizon of 5 minutes or less.

Input: A set of *short-term* demands between different network endpoints.

Output: The *best* set of optical links & network paths to satisfy all demands.

Number of connected labeled graphs with n nodes.

n	$a(n)$
0	1
1	1
2	1
3	4
4	38
5	728
6	26704
7	1866256
8	251548592
9	66296291072
10	34496488594816
11	35641657548953344
12	73354596206766622208
13	301272202649664088951808
14	2471648811030443735290891264
15	40527680937730480234609755344896
16	1328578958335783201008338986845427712

Source: <https://oeis.org/A001187>

Problem 2: “Adapting Topology to suit Traffic is Impractical”

Jointly Optimizing Routing & Topology

short-term: a time horizon of 5 minutes or less.

Input: A set of *short-term* demands between different network endpoints.

best: an application-specific objective.

Output: The *best* set of optical links & network paths to satisfy all demands.

Number of connected labeled graphs with n nodes.

n	$a(n)$
0	1
1	1
2	1
3	4
4	38
5	728
6	26704
7	1866256
8	251548592
9	66296291072
10	34496488594816
11	35641657548953344
12	73354596206766622208
13	301272202649664088951808
14	2471648811030443735290891264
15	40527680937730480234609755344896
16	1328578958335783201008338986845427712

Source: <https://oeis.org/A001187>

Problem 2: “Adapting Topology to suit Traffic is Impractical”

Jointly Optimizing Routing & Topology

short-term: a time horizon of 5 minutes or less.

Input: A set of *short-term* demands between different network endpoints.

best: an application-specific objective.

Output: The *best* set of optical links & network paths to satisfy all demands.

This problem is NP-Hard.

- The sheer number of possible connected graphs with n vertices is astronomical.

With 16 nodes, we’re already considering more than 10^{36} graphs.

Number of connected labeled graphs with n nodes.

n	$a(n)$
0	1
1	1
2	1
3	4
4	38
5	728
6	26704
7	1866256
8	251548592
9	66296291072
10	34496488594816
11	35641657548953344
12	73354596206766622208
13	301272202649664088951808
14	2471648811030443735290891264
15	40527680937730480234609755344896
16	1328578958335783201008338986845427712

Source: <https://oeis.org/A001187>

Problem 3: “The Benefit of Dynamically Changing Topology is Unclear”

The prevailing mindset

Networking is worlds away from where it was 20 years ago.

Problem 3: “The Benefit of Dynamically Changing Topology is Unclear”

The prevailing mindset

Networking is worlds away from where it was 20 years ago.

Modern demand from applications such as AI and machine learning, in combination with ever-more pervasive threats against network infrastructure such as DDoS, compel us to revisit this assumption.

Problem 3: “The Benefit of Dynamically Changing Topology is Unclear”

The prevailing mindset

Networking is worlds away from where it was 20 years ago.

Modern demand from applications such as AI and machine learning, in combination with ever-more pervasive threats against network infrastructure such as DDoS, compel us to revisit this assumption.

To boldly go where no enterprise network has gone before.



Problem 3: “The Benefit of Dynamically Changing Topology is Unclear”

The prevailing mindset

Networking is worlds away from where it was 20 years ago.

Modern demand from applications such as AI and machine learning, in combination with ever-more pervasive threats against network infrastructure such as DDoS, compel us to revisit this assumption.

To boldly go where no enterprise network has gone before.

To unlock the benefits of dynamically changing the topology for scaling network capacity to meet demand and shoring up its defense capability.



Thesis Statement

This thesis advances the state-of-the-art in network management by challenging the prevailing notion that the joint optimization of optical and packet layers is impractical. It does so through two key contributions. (1) Establishing theoretical and empirical foundations for optical topology programming (OTP). (2) Demonstrating the advantages of OTP in enhancing network security (e.g., combating network reconnaissance, volumetric DDoS) and network management (e.g., scaling traffic engineering) applications.



The Optical Layer Network

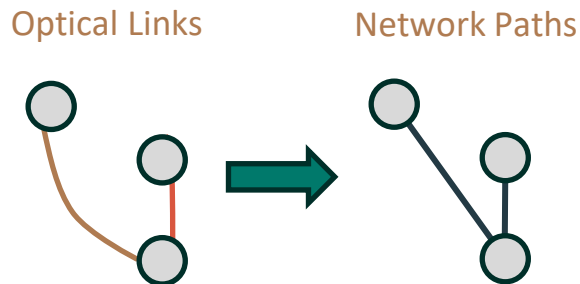


Definitions

Optical Link: Physical link between two endpoints. Can traverse zero or more nodes between endpoints. Also known as **wavelength**.

Network Path: One or more optical links between two nodes.

Capacity or Bandwidth of a network path is proportional to the number of optical links.

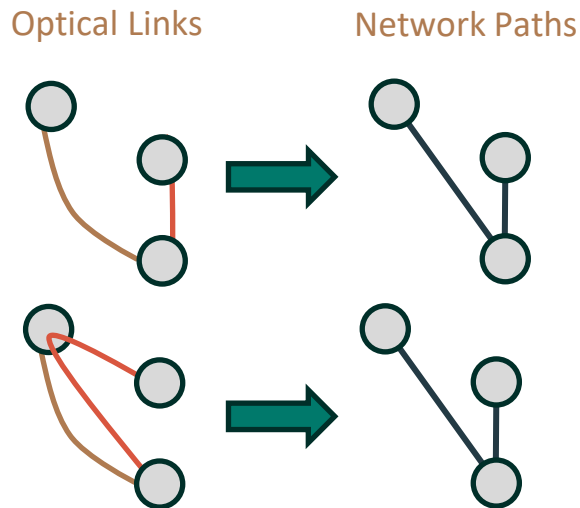


Definitions

Optical Link: Physical link between two endpoints. Can traverse zero or more nodes between endpoints. Also known as **wavelength**.

Network Path: One or more optical links between two nodes.

Capacity or Bandwidth of a network path is proportional to the number of optical links.

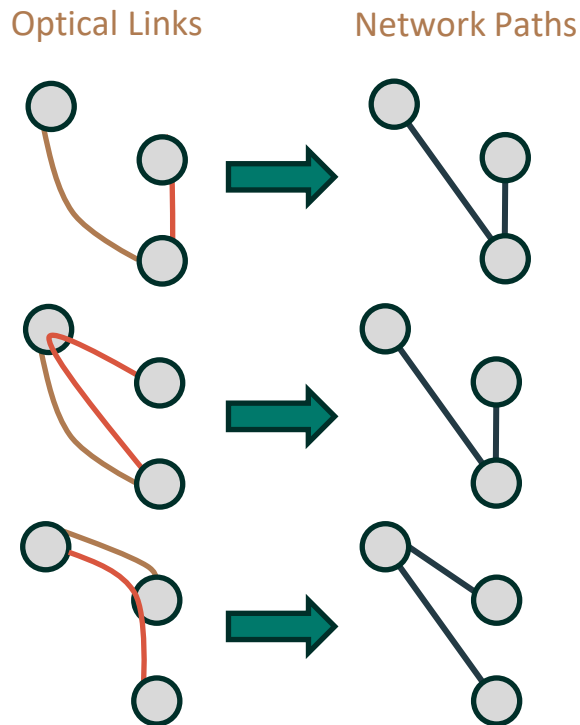


Definitions

Optical Link: Physical link between two endpoints. Can traverse zero or more nodes between endpoints. Also known as **wavelength**.

Network Path: One or more optical links between two nodes.

Capacity or Bandwidth of a network path is proportional to the number of optical links.



Optical Topology Programming (OTP)

A mechanism to opportunistically (re)allocate or move optical links toward improving the performance or security of that network.

Benefits and Intuition:

Bandwidth on demand for existing network paths.

New network paths for opportunistically forwarding traffic.

How we get there:

Foundations to address the practicality problem: Find an efficient solution to an NP-hard problem.

Measurements to address the feasibility problem: Show that OTP is feasible in production networks today.

Applications to demonstrate the benefits of OTP: Develop applications to scale capacity & address threats.



Foundations for OTP



Joint optimization for routing & topology is NP-Hard.

Joint optimization for routing & topology is NP-Hard.
What can we do about this?

Joint optimization for routing & topology is NP-Hard.

What can we do about this?

Simplify the search.

Joint optimization for routing & topology is NP-Hard.

What can we do about this?

Simplify the search.

There is an incomprehensibly large number of possible network topologies to choose from...

Joint optimization for routing & topology is NP-Hard.

What can we do about this?

Simplify the search.

There is an incomprehensibly large number of possible network topologies to choose from...
but let's be realistic – we don't need to consider all of them.

How to Jointly Optimize Routing + Topology

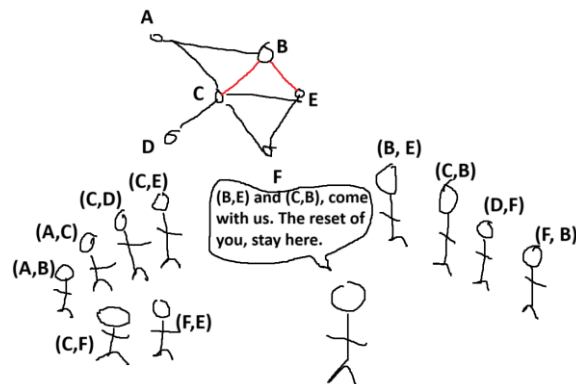
With John Willson Matt Nance-Hall

1. Choose a set of candidate links in the network.

How to Jointly Optimize Routing + Topology

With John-Willson Matt Nance-Hall

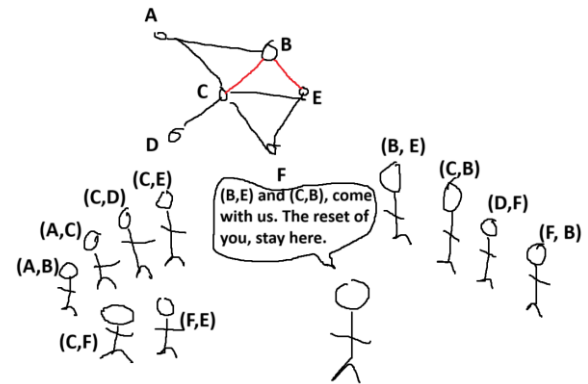
1. Choose a set of candidate links in the network.



How to Jointly Optimize Routing + Topology

With John-Willson Matt Nance-Hall

1. Choose a set of candidate links in the network.
2. Enumerate the possible paths between all nodes using the candidate links.

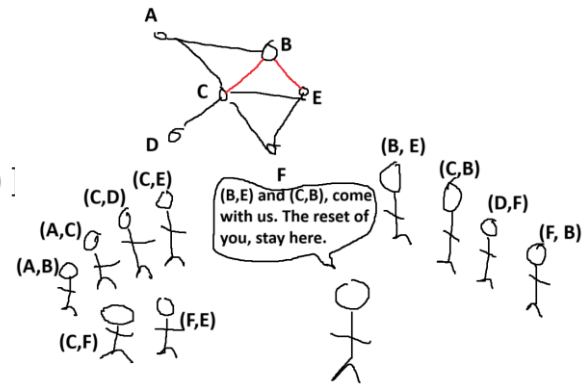


How to Jointly Optimize Routing + Topology

With John-Willson Matt Nance-Hall

1. Choose a set of candidate links in the network.
 2. Enumerate the possible paths between all nodes using the candidate links.
- Use the shortest path length in the original graph as a cutoff.

E.g., **B** -> **C** has length 2 in the original graph [**(B, A), (A, C)**]
so, include [**(B, C)**] and [**(B, E), (E, C)**] but not
[**(B, E), (E, F), (F, C)**].



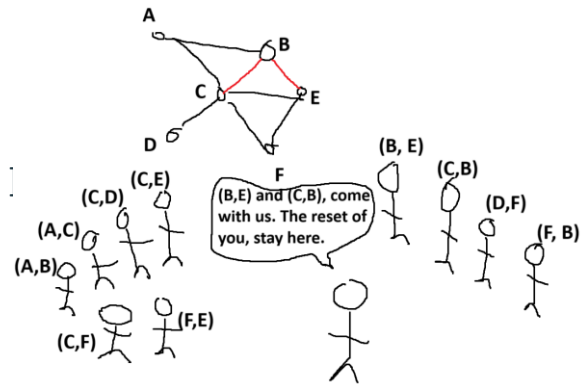
How to Jointly Optimize Routing + Topology

With John-Willson Matt Nance-Hall

1. Choose a set of candidate links in the network.
2. Enumerate the possible paths between all nodes using the candidate links.
 - Use the shortest path length in the original graph as a cutoff.

E.g., **B** -> **C** has length 2 in the original graph [**(B, A)**, **(A, C)**]
so, include [**(B, C)**] and [**(B, E)**, **(E, C)**] but not
[**(B, E)**, **(E, F)**, **(F, C)**].

3. For each pair of nodes, (s, t) , call the links from these paths $\mathcal{I}^{s \rightarrow t}$
So, for **B** -> **C** this is { **(B, C)**, **(B, A)**, **(A, C)**, **(B, E)**, **(E, C)** }



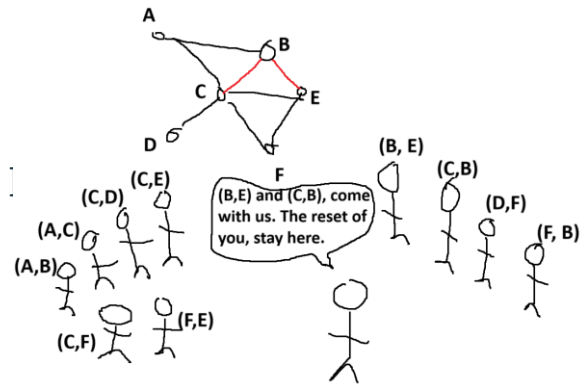
How to Jointly Optimize Routing + Topology

With John-Willson Matt Nance-Hall

1. Choose a set of candidate links in the network.
2. Enumerate the possible paths between all nodes using the candidate links.
 - Use the shortest path length in the original graph as a cutoff.

E.g., **B** -> **C** has length 2 in the original graph [**(B, A), (A, C)**]
so, include [**(B, C)**] and [**(B, E), (E, C)**] but not
[**(B, E), (E, F), (F, C)**].

3. For each pair of nodes, (s, t) , call the links from these paths $\mathcal{F}^{s \rightarrow t}$
So, for **B** -> **C** this is { **(B, C), (B, A), (A, C), (B, E), (E, C)** }
4. Restrict all (s, t) flows for any solution to only include $\mathcal{F}^{s \rightarrow t}$ links.



Foundations Summary

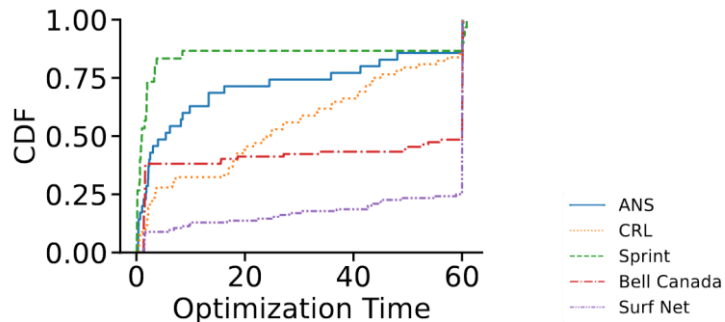
Restricting the forwarding links available to flows enables us to solve joint routing + topology problems in *less than 1 minute* for graphs with up to 50 nodes.

Foundations Summary

Restricting the forwarding links available to flows enables us to solve joint routing + topology problems in less than 1 minute for graphs with up to 50 nodes.

Significant because the worlds largest cloud backbones have 50 nodes or fewer.

Network	Nodes	Links
Sprint	11	18
ANS	18	25
CRL	33	38
Bell Canada	48	65
SurfNet	50	68






Measurements

Benchmarking The Optical Layer



Optical link add-time:
The time required to add an optical link to
an existing set of links.

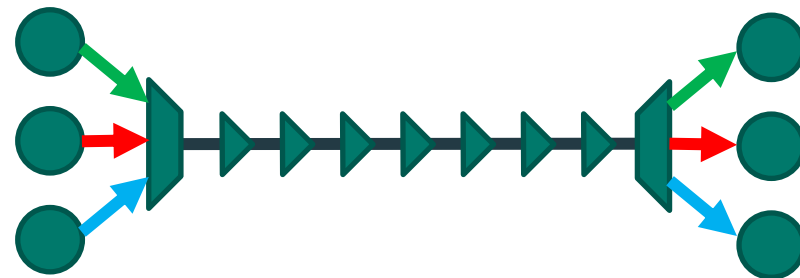


Which factors impact add-time for optical link the most?

How can these factors be reduced?



Lab testbed

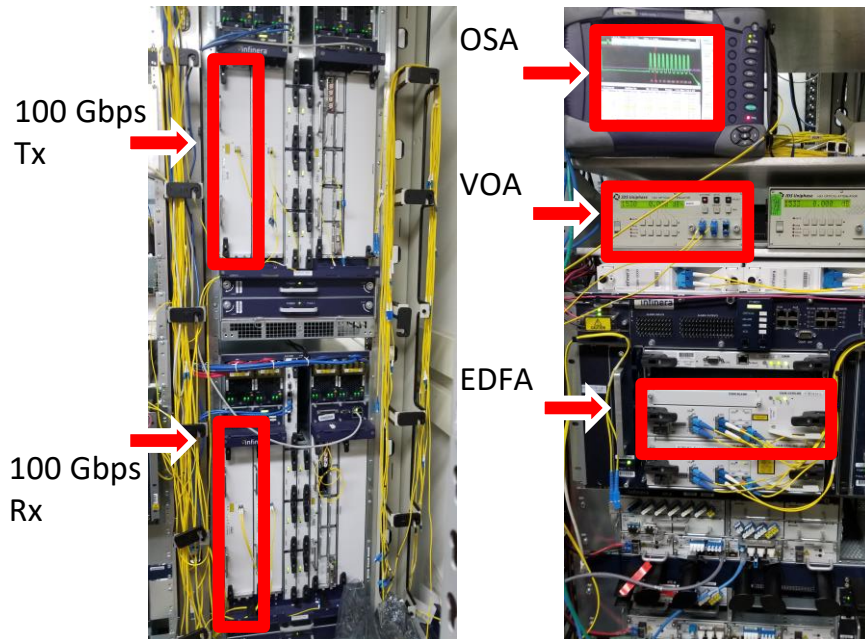


6x 100 Gigabit per second (Gbps) Transponders.
Transmit (Tx) and receive (Rx) optical signals.

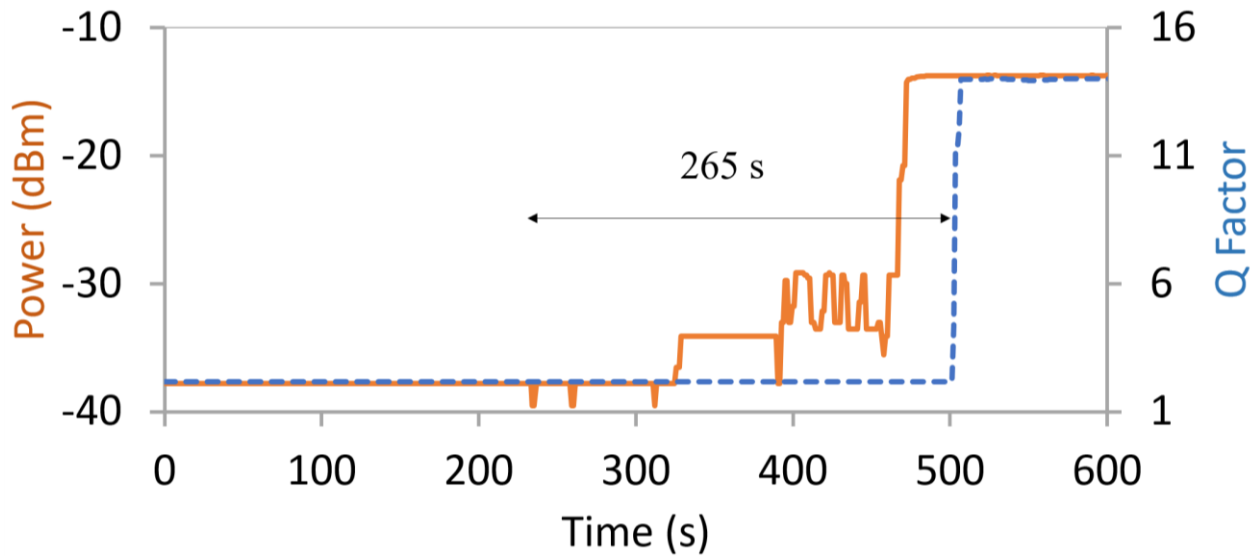
7x erbium-doped fiber amplifiers (EDFAs).
Deployed roughly every 100 km or 60 miles in a
production WAN.

1x variable optical attenuator (VOA)
To instantly add/drop link.

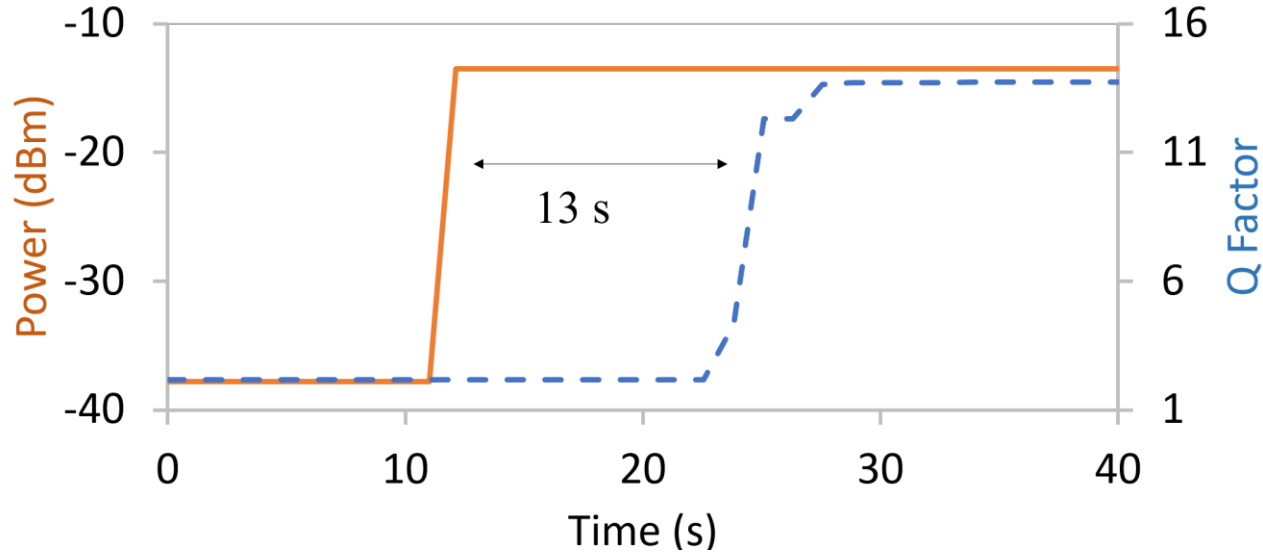
1x Optical spectrum analyzer (OSA)
To visualize optical channels



Default Time to Add 100 Gbps Optical Link



Improved Time to Add 100 Gbps Optical Link





20x Faster

- Optical signal add-time decreased 20x by disallowing 'automatic' power adjustment in favor of 'manual'.



Applications





GreyLambda:

A Framework to Scale Traffic Engineering Using OTP

Background

- Enterprise networks are expensive and therefore higher utilization = higher return on investment
- Enterprise networks need to be *overprovisioned* to ensure network availability in severe events e.g., flash crowds & fiber cuts
- Overprovisioning, or designing and building the network such that on average utilization is low to provide insurance against severe events, works directly against achieving higher utilization.

The Network Balancing Act

Temporal vs. Spatial requirements of traffic engineering (TE).



Temporal Requirement

- Forwarding paths should be computed for all demand pairs quickly, ideally every 5 minutes.

Spatial Requirement

- Forwarding paths should be diverse and balanced to ensure high utilization, even during severe network events.

The Network Balancing Act

Instances of TE Systems on Either Side of the Spectrum



Temporal Requirement

Equal-Cost Multi Path (ECMP) Routing.

Very fast to compute, $O(1)$.

Doesn't do well in severe network events.

Spatial Requirement

Optimal Multi-commodity Flow (MCF) Routing.

Much slower to compute, $O(n^2)$.

Adapts to severe network events.

The Network Balancing Act

Instances of TE Systems on Either Side of the Spectrum



Temporal Requirement

Equal-Cost Multi Path (ECMP) Routing.

Very fast to compute, $O(1)$.

Doesn't do well in severe network events.

Spatial Requirement

Optimal Multi-commodity Flow (MCF) Routing.

Much slower to compute, $O(n^2)$.

Adapts to severe network events.

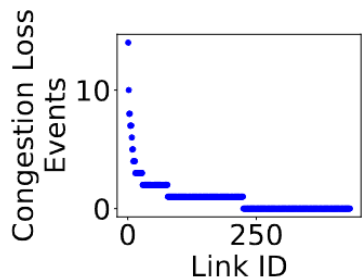
State-of-the art, Somewhere in Between

SMORE: MCF Heuristic with oblivious path selection

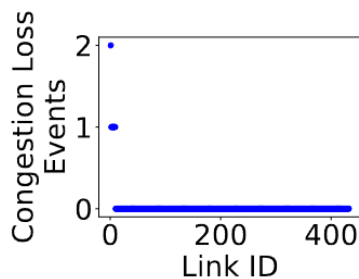
NCFlow: Parallelized MCF

Observation

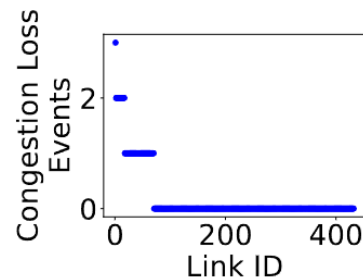
- Severe network events tend to disproportionately affect specific network links across TE implementations.
- We'll call these “High Rank” links.
- Graphs show “Total Congestion Loss Events per Link” in Microsoft’s Azure backbone with flash crowds and two link failures



(a) ECMP



(b) MCF



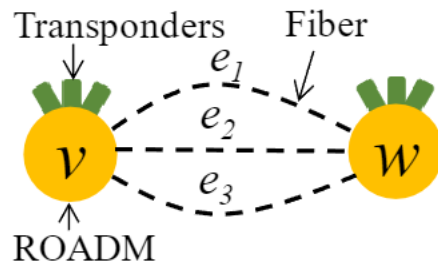
(c) SMORE

Implication

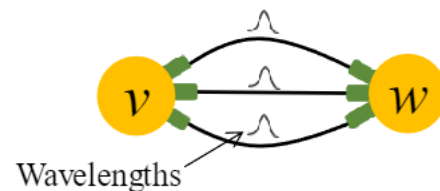
Suppose e_1 , e_2 , and e_3 are different optical paths from v to w .

In a typical TE system, the failure event of e_1 and e_2 means all the traffic from v to w must use e_3 's single optical link.

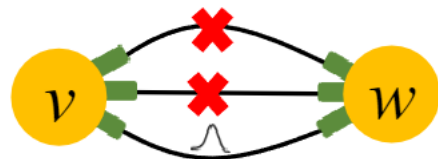
If we can migrate the wavelengths from the e_1 and e_2 paths onto e_3 , then we can mitigate congestion loss from the event.



(a)



(b)



(c)



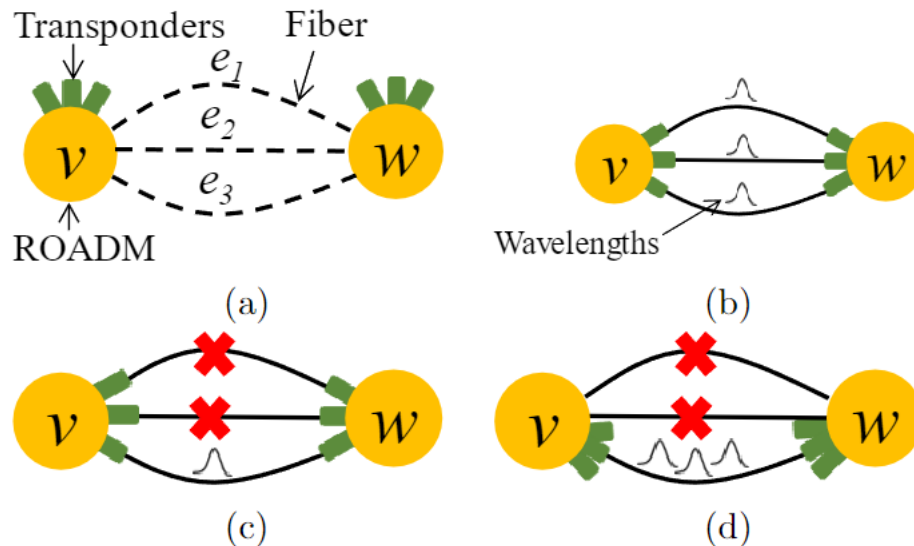
(d)

Implication

Suppose e_1 , e_2 , and e_3 are different optical paths from v to w .

In a typical TE system, the failure event of e_1 and e_2 means all the traffic from v to w must use e_3 's single optical link.

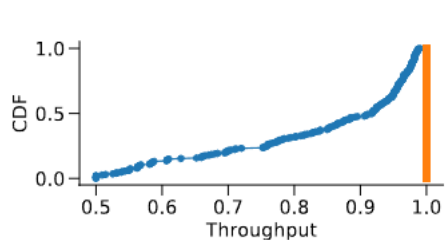
If we can migrate the wavelengths from the e_1 and e_2 paths onto e_3 , then we can mitigate congestion loss from the event.



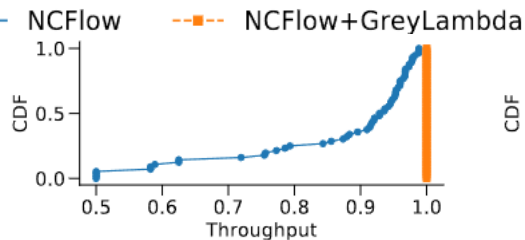
We can achieve this benefit by introducing *follow transponders* at only **high ranked links**

Results

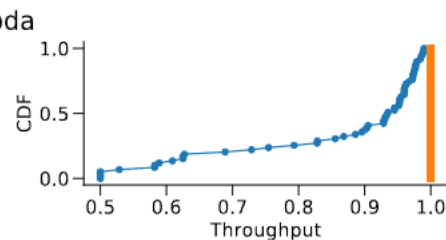
- GreyLambda completely mitigates congestion loss from severe network events in Microsoft's Azure backbone when paired with NCFlow.



(a) Flash Crowd



(b) Flash Crowd+1 Link Failure



(c) Flash Crowd+2 Link Failures

OTP for DDoS Defense

Multi-stage link-flood attack

Step 1: Reconnaissance

Step 2: Send Link-flood Traffic

OTP for DDoS Defense

Multi-stage link-flood attack

OTP for DDoS Defense

Multi-stage link-flood attack

Step 1: Reconnaissance

Step 2: Send Link-flood Traffic



Doppler:

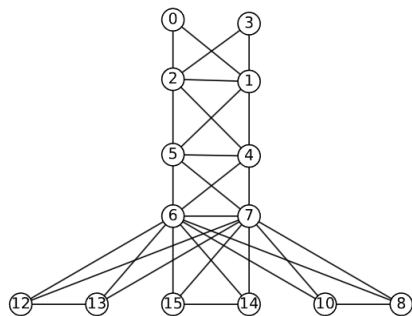
A Framework to Defend Against Network Reconnaissance Attacks

Background

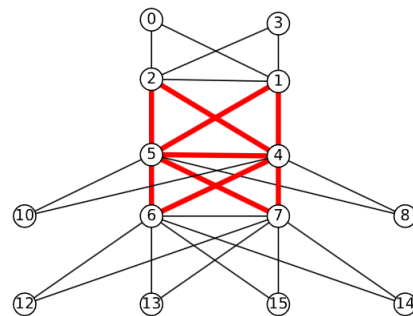
- DDoS attacks are more prevalent by the year, and attack methods deployed are increasingly cunning.
- Attackers are becoming more adept at performing *network reconnaissance*, i.e., mapping the target network to find vulnerable *bottleneck links*.
- Recent work has focused on thwarting reconnaissance by obfuscating **traceroute** probes sent through the network.
- We show an advanced reconnaissance attack, which doesn't use **traceroute**, and a method to defend against it with OTP.

The Ricci Attack

- Application of a recently discovered method for *cloud tomography*, or mapping cloud backbones.
- Originally wasn't considered as an attack vector for learning enterprise network topology.
- We were the first to apply it to the LFA attack loop.
- Uses min RTT delay measurements to discover bottleneck links in a network.
- *Requires accurate min RTT measurements between pairs of network nodes.*



(a) Ground Truth topology from network operator.

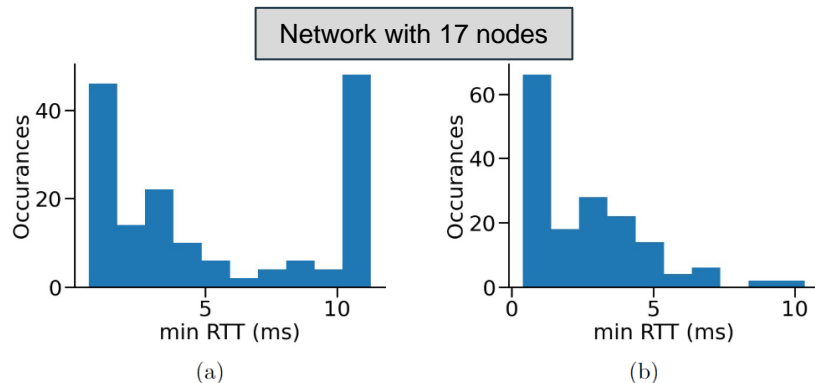


(b) Topology inferred with Ricci Attack

Ricci Attack: Time requirement

Adapt and change the optical topology faster than the attacker can perform the Ricci attack.

- Measurements in (a) collected over 25 minutes.
- Measurements in (b) collected over 1 hour and 36 minutes.

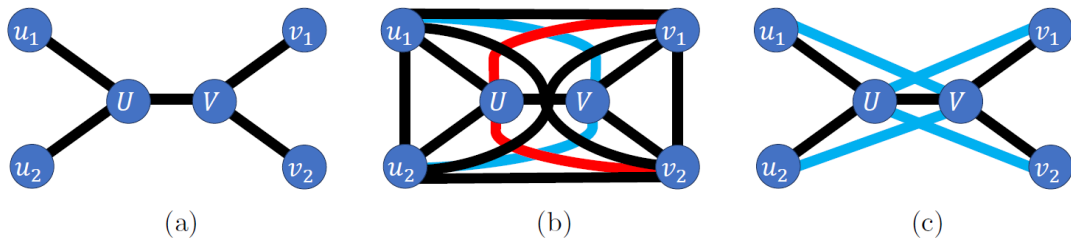


It can take tens of minutes to hours to get clean data necessary for the Ricci attack.

Doppler Defense

- Doppler proactively adapts and changes the optical topology of the network to subvert reconnaissance efforts.
- Challenging because of the vast potential choice for topology and connectivity options.

Doppler Defense: Candidate link selection



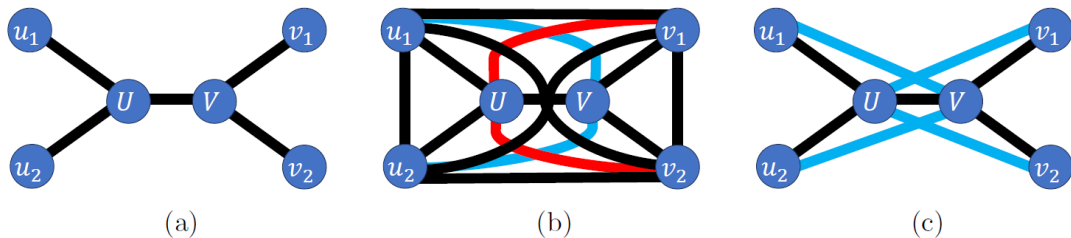
Strategy 1: *max*

Given a network with a subgraph like (a), we can consider the largest set of candidate links to be (b), i.e., a fully-connected graph around (U, V) and its neighbors.

Strategy 2: *conservative*

Candidate links are those from U to V 's neighbors and V to U 's neighbors.

Doppler Defense: Candidate link selection



Strategy 1: *max*

Given a network with a subgraph like (a), we can consider the largest set of candidate links to be (b), i.e., a fully-connected graph around (U, V) and its neighbors.

Strategy 2: *conservative*

Candidate links are those from U to V 's neighbors and V to U 's neighbors.

This choice dramatically influences the **set of possible solutions** and the **speediness of finding one**.

Doppler Evaluation

Tested the Ricci attack and Doppler defense on four networks.

Experiment Parameters

- Candidate link and path selection (*conservative* or *max*)
- Transponders placement (top 100, 90, ..., 10, 0% of nodes)
- Quantity of fallow transponders where placed (1, 2, or 3)
- Time (30 seconds, 1 minute, 5 minutes)

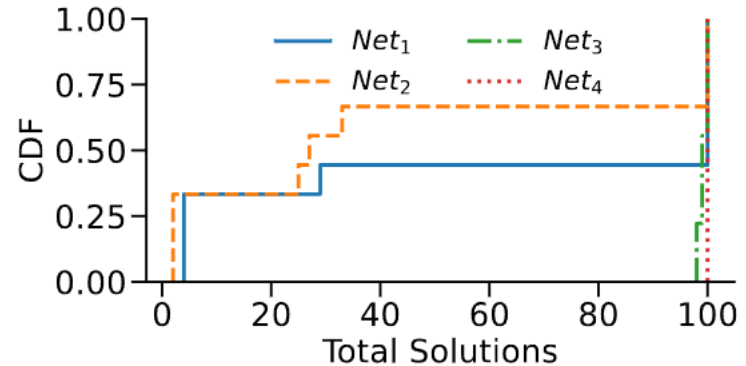
Investigating:

1. How does Doppler fare with no fallow transponders?
2. How similar are Doppler topologies to each other?
3. How does Doppler perform with low time constraints?

Network	L2+L3 Switches	Open Router Proxy	Mappable?
<i>Network₁</i>	14	×	✓
<i>Network₂</i>	12	×	✓
<i>Network₃</i>	17	✓	✓
<i>Network₄</i>	10	✓	✓

Doppler with No Fallow Transponders

- Doppler finds multiple alternative topologies for all 4 networks under all configurations.
- **Networks 1 and 2** found fewer solutions specifically with *max* link and path selection and a **30 second** time constraints.
- Doppler completed the solution pool for all networks with the *conservative* link and path selection strategy.
- The *conservative* strategy introduced 65 to 70% fewer variables than *max*



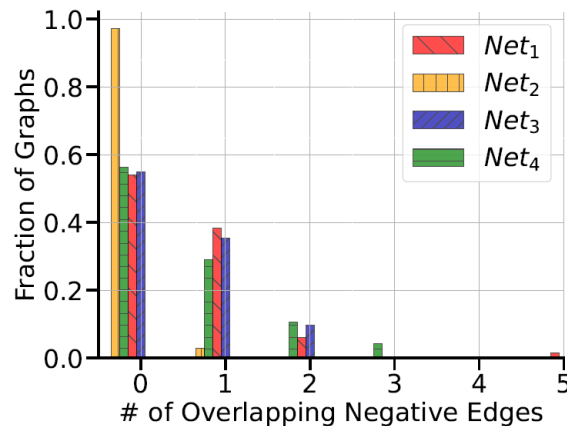
Reconnaissance: Before and After Doppler

Comparison of Doppler topologies

- Majority of graphs from Doppler have 0 overlapping bottleneck edges.
- Less than 40% have 1 or more overlapping bottleneck edges.

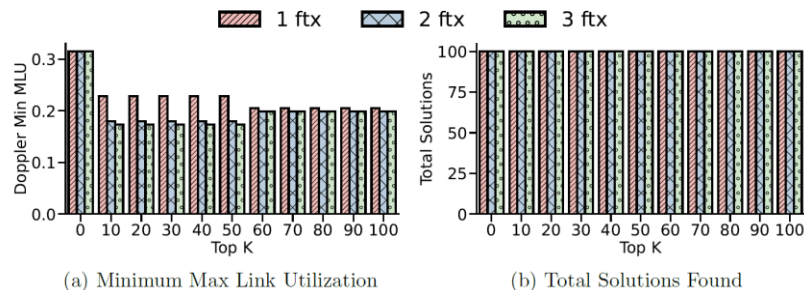
Key takeaway

- There is high variance in the Doppler topologies with respect to the location of a bottleneck link.



Doppler with a Low Time Limit

- We look at the performance of Doppler solutions specifically with a **30 second limit** to complete its solution pool.
- The graph shows **Network 3**, but results for all networks were similar.
 - Doppler maintained low Max Link Utilization for solutions across operating parameters.
 - Doppler found 100 feasible solutions.





ONSET:

A Framework to Combat Terabit Link Flood Attacks

Background

Recall Doppler, a pro-active solution to make network reconnaissance more challenging.

ONSET directly combats an active, ongoing link-flood attack through optical topology programming.

Software define networking has been used to address the LFA threat in the past.

Ripple:

- Aims to detect traffic from LFA attackers directly and drop that malicious traffic.

Downside:

- LFA traffic is notoriously hard to detect.
- As attackers are getting more sophisticated their ability to blend their attacks with normal traffic is increasing.

Goal

Develop a framework for LFA defense that can be applied to legacy networks (i.e., without SDN traffic engineering or defenses) and modern networks with state-of-the-art SDN based defense.

Challenges

- Foundations Challenge (How to jointly optimize topology + Routing). ✓

Challenges

- Foundations Challenge (How to jointly optimize topology + Routing). ✓
- Managing Network Performance, especially in non-SDN routed networks.

Challenges

- Foundations Challenge (How to jointly optimize topology + Routing). ✓
- Managing Network Performance, especially in non-SDN routed networks.

Why?

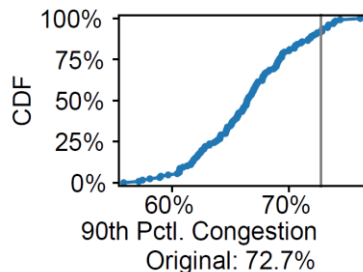
In a network with equal-cost multi-path (ECMP) routing, adding a new link to the network could sometimes increase network congestion.

Challenges

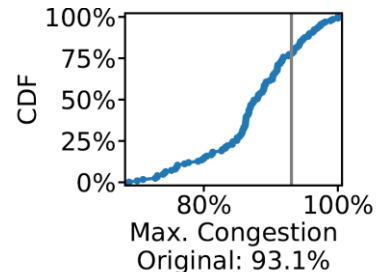
- Foundations Challenge (How to jointly optimize topology + Routing). ✓
- Managing Network Performance, especially in non-SDN routed networks.

Why?

In a network with equal-cost multi-path (ECMP) routing, adding a new link to the network could sometimes increase network congestion.



(a) CDF of 90th percentile congestion after adding different links.



(b) CDF of maximum congestion after adding different links.

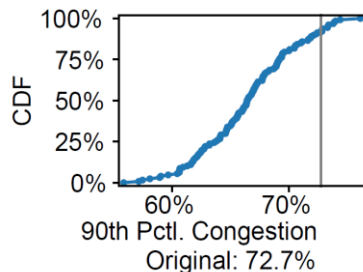
Challenges

- Foundations Challenge (How to jointly optimize topology + Routing). ✓
- Managing Network Performance, especially in non-SDN routed networks.

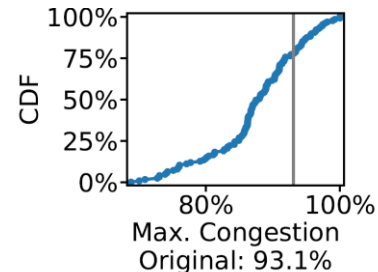
Why?

In a network with equal-cost multi-path (ECMP) routing, adding a new link to the network could sometimes increase network congestion.

~15% of link additions increased 90th percentile link congestion.
~25% increased max link congestion.



(a) CDF of 90th percentile congestion after adding different links.



(b) CDF of maximum congestion after adding different links.

Addressing Network Performance in non-SDN Networks

Binary links make ECMP difficult to model efficiently with numerical optimization.

They turn the mixed-integer **linear** programming problem into a mixed-integer **quadratic** problem.

Addressing Network Performance in non-SDN Networks

Binary links make ECMP difficult to model efficiently with numerical optimization.

They turn the mixed-integer **linear** programming problem into a mixed-integer **quadratic** problem.

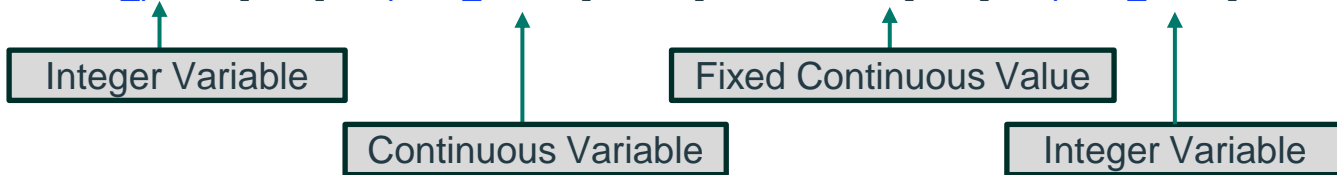
```
# flow_balance_constraint
for (s,t,i) in paths: # each path `i` from `s` to `t`
    model.addConstr(
        total_paths[s,t] * path_flows[s,t,i] == demand[s,t] * path_vars[s,t,i])
```

Addressing Network Performance in non-SDN Networks

Binary links make ECMP difficult to model efficiently with numerical optimization.

They turn the mixed-integer **linear** programming problem into a mixed-integer **quadratic** problem.

```
# flow_balance_constraint
for (s,t,i) in paths: # each path `i` from `s` to `t`
    model.addConstr(
        total_paths[s,t] * path_flows[s,t,i] == demand[s,t] * path_vars[s,t,i])
```



Addressing Network Performance in non-SDN Networks



Instead, we use the SDN optimization model (i.e., without a flow-balance constraint).

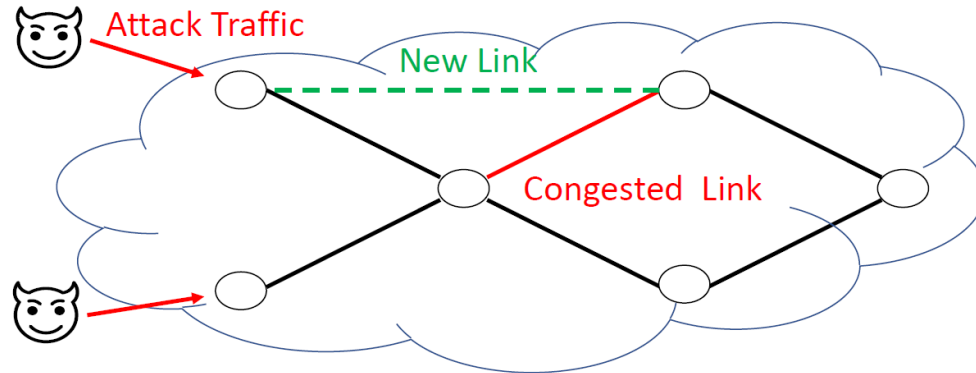
We use our technique from Doppler to find a large set of solutions.

Then solve the ECMP routing scheme on each solution in parallel.

The “winner” is the solution with the lowest maximum link utilization.

ONSET: An LFA Defense Framework Using Optical Topology Programming

1.  Topology Pruning
2.  Joint Topology & Routing Optimization



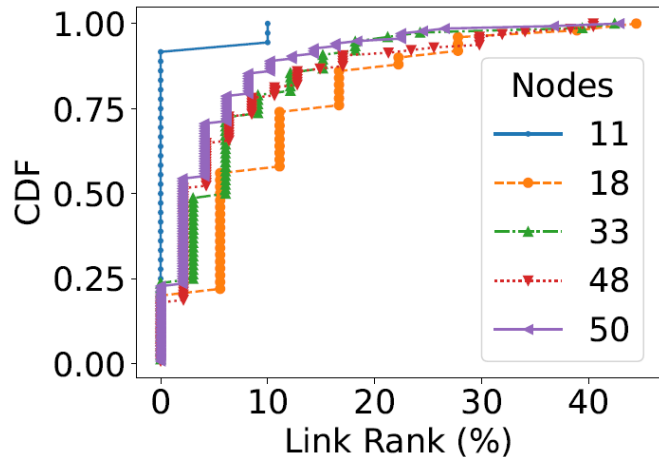
ONSET: Topology Pruning (step 1)

Apply a technique similar to that from Doppler, i.e., a limited, deliberately chosen, set of candidate links.

Selection Technique:

Link Rank – Measures the number of attacks that induce congestion loss on a given link.

For example, when 100 attacks are considered on a network, and a 178 given link experiences congestion loss in 12 of those cases, the link rank for that link is 12%.



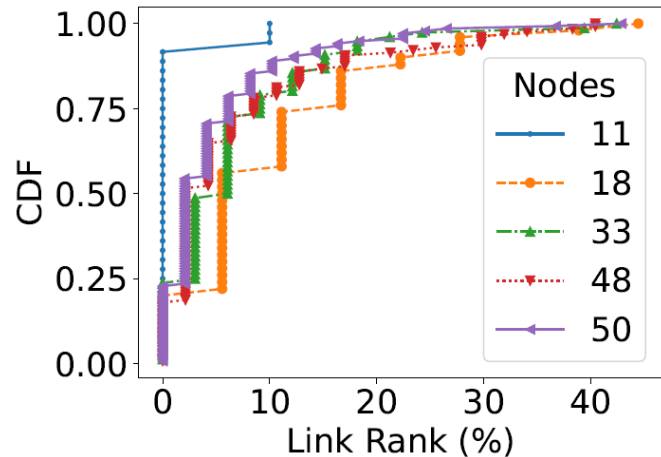
ONSET: Topology Pruning (step 1)

Apply a technique similar to that from Doppler, i.e., a limited, deliberately chosen, set of candidate links.

Selection Technique:

Link Rank – Measures the number of attacks that induce congestion loss on a given link.

For example, when 100 attacks are considered on a network, and a 178 given link experiences congestion loss in 12 of those cases, the link rank for that link is 12%.



Insight: Most links have a low rank (< 10%) while a few links in all networks have high rank, between 30 and 40%. Therefore, prioritize reconfiguration around these links.

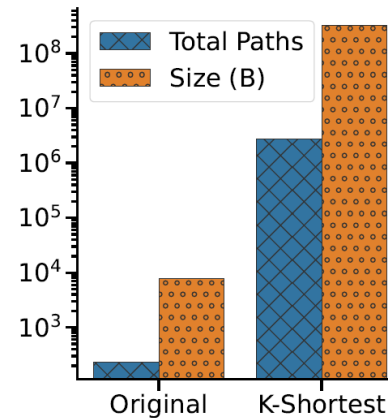
ONSET: Topology Pruning (step 2)

We must enumerate all the forwarding paths for potential topologies to optimize routing on them.

ONSET: Topology Pruning (step 2)

We must enumerate all the forwarding paths for potential topologies to optimize routing on them.

This can be 3 to 4 orders of magnitude larger than the set of all-pairs shortest paths in the original graph!

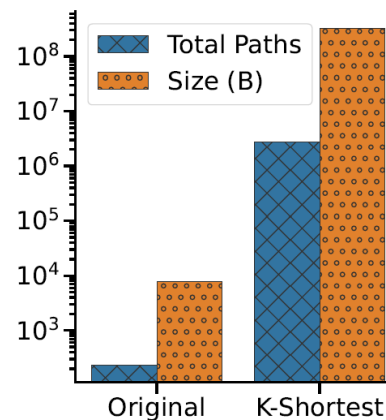


ONSET: Topology Pruning (step 2)

We must enumerate all the forwarding paths for potential topologies to optimize routing on them.

This can be 3 to 4 orders of magnitude larger than the set of all-pairs shortest paths in the original graph!

Using A* to iteratively add paths to this set, instead of Dijkstra's algorithm, we reduce the total paths by 2 to 3 orders of magnitude.

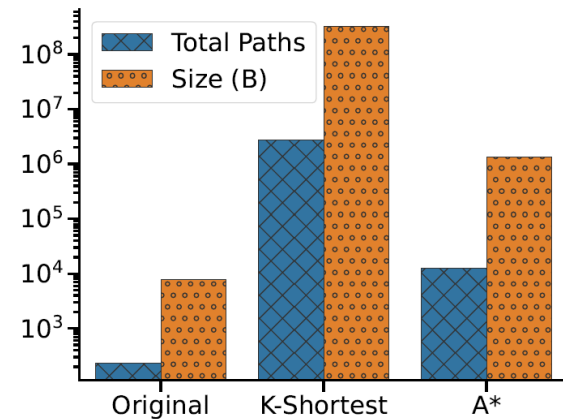


ONSET: Topology Pruning (step 2)

We must enumerate all the forwarding paths for potential topologies to optimize routing on them.

This can be 3 to 4 orders of magnitude larger than the set of all-pairs shortest paths in the original graph!

Using A* to iteratively add paths to this set, instead of Dijkstra's algorithm, we reduce the total paths by 2 to 3 orders of magnitude.



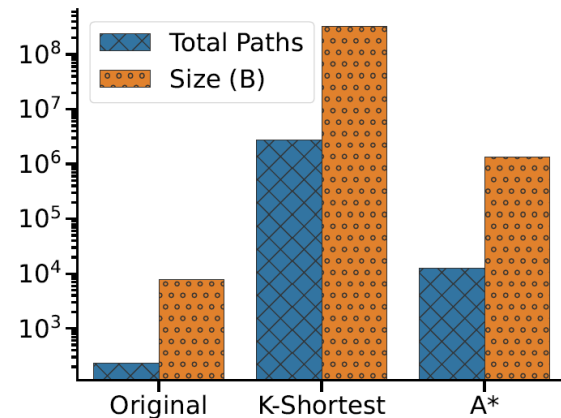
ONSET: Topology Pruning (step 2)

We must enumerate all the forwarding paths for potential topologies to optimize routing on them.

This can be 3 to 4 orders of magnitude larger than the set of all-pairs shortest paths in the original graph!

Using A* to iteratively add paths to this set, instead of Dijkstra's algorithm, we reduce the total paths by 2 to 3 orders of magnitude.

This is key to enabling fast solutions on large graphs.



ONSET Evaluation

We evaluate ONSET on 5 networks with the following parameters.

- Routing/Defense:
 - Ripple, ECMP, Ripple+ONSET, ECMP+ONSET
- Fallow transponders
 - 1 fallow transponder at each node.
- Candidate Link/Path Selection
 - Conservative candidate link selection with A* path selection

Investigating:

1. How does ONSET handle LFAs at different scales?
2. How does its performance compare to Ripple or ECMP?

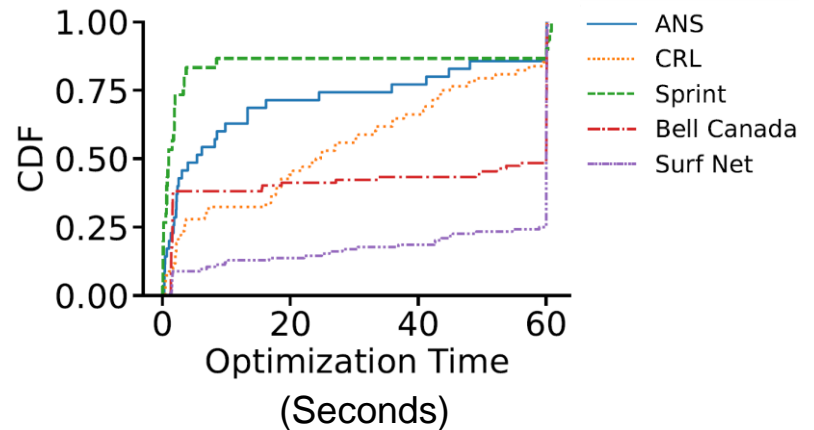
Network	Nodes	Links
Sprint	11	18
ANS	18	25
CRL	33	38
Bell Canada	48	65
SurfNet	50	68

ONSET Optimization Time

ONSET finds an optimal solution within 60 seconds for Sprint, ANS, and CRL in nearly all experiments.

Bell Canada has an optimal solution in 38% of experiments.

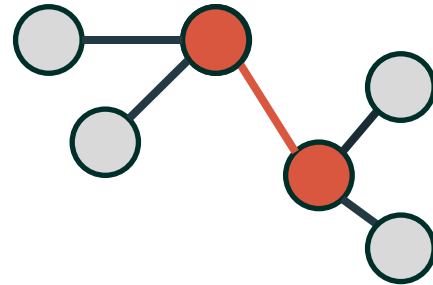
SurfNet has an optimal solution in 25% of experiments.



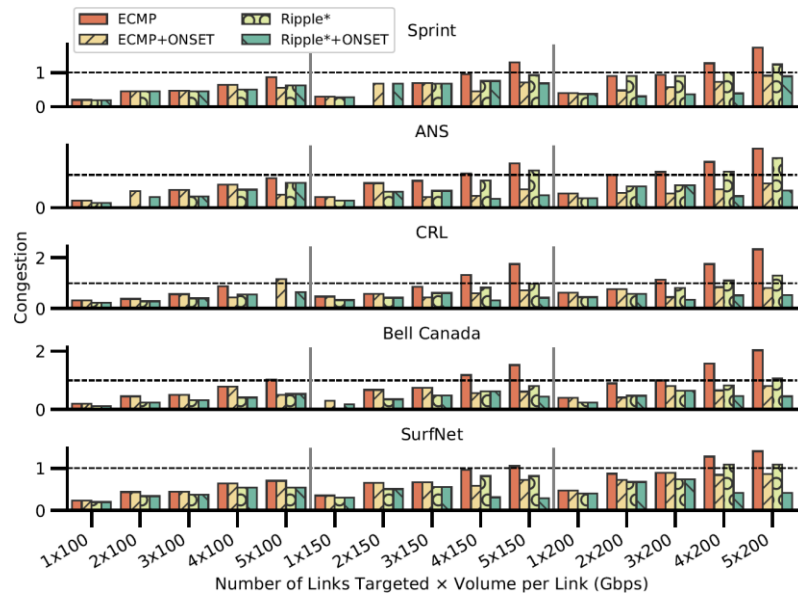
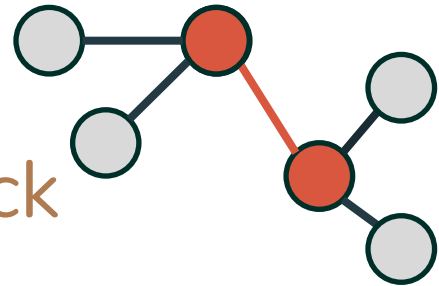
ONSET Evaluation: Attacks

Coremelt Attack

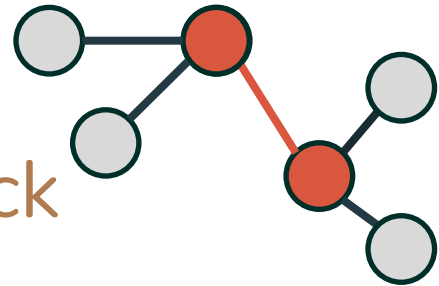
- Concentrates on attacking one or more links.
- We look at crossfire attacks targeting 1 to 5 links simultaneously with an attack volume of 100, 150, or 200 Gbps per link.



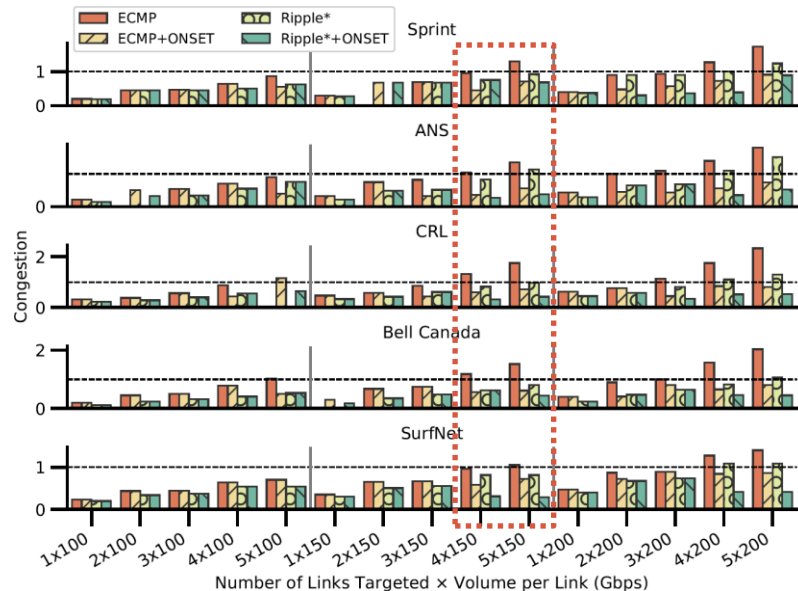
ONSET Evaluation: Coremelt Attack



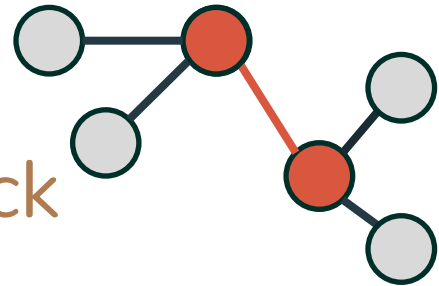
ONSET Evaluation: Coremelt Attack



In the 4 link by 150 Gbps per link attacks, ECMP starts to show congestion loss on each network.

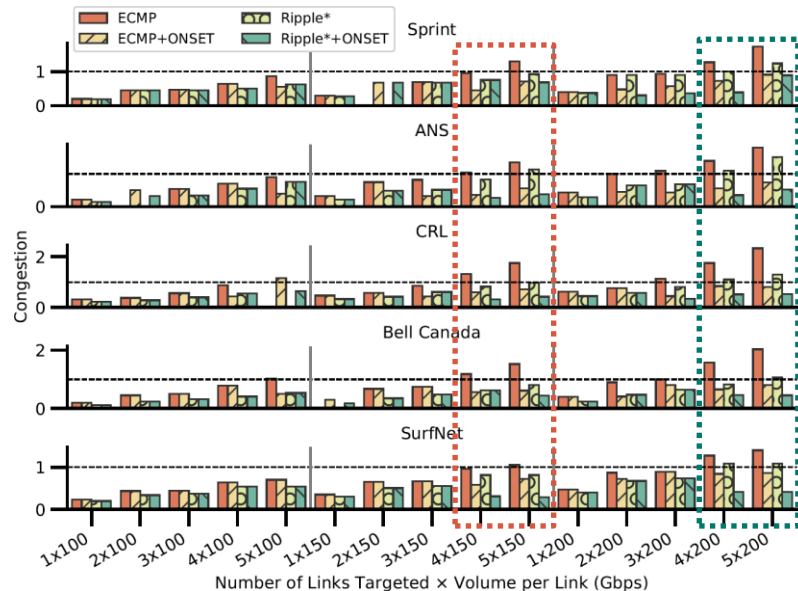


ONSET Evaluation: Coremelt Attack

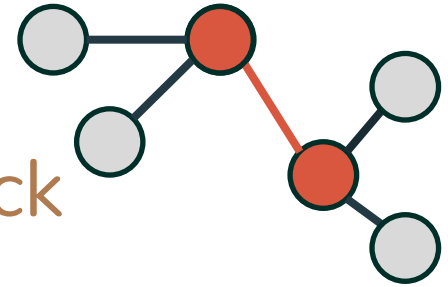


In the 4 link by 150 Gbps per link attacks, ECMP starts to show congestion loss on each network.

In the 4 link by 200 Gbps per link attacks, Ripple* starts to show congestion loss on each network.



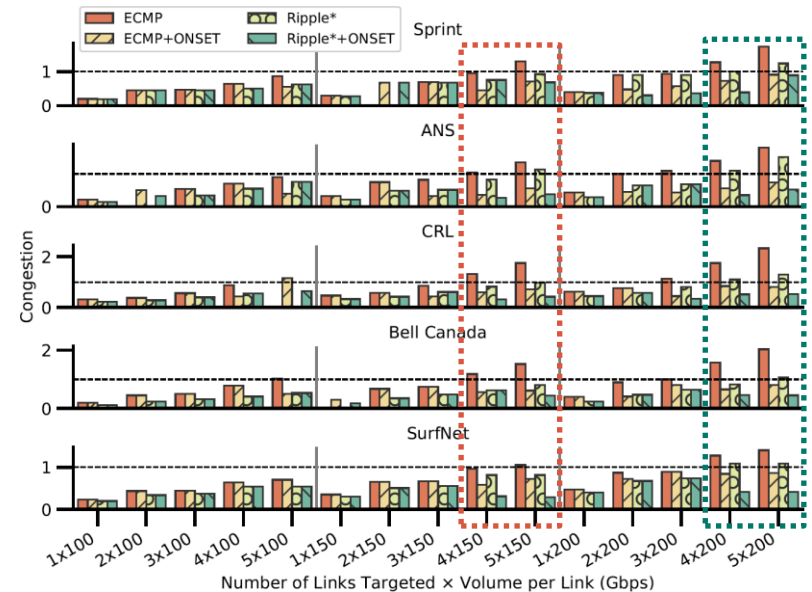
ONSET Evaluation: Coremelt Attack



In the 4 link by 150 Gbps per link attacks, ECMP starts to show congestion loss on each network.

In the 4 link by 200 Gbps per link attacks, Ripple* starts to show congestion loss on each network.

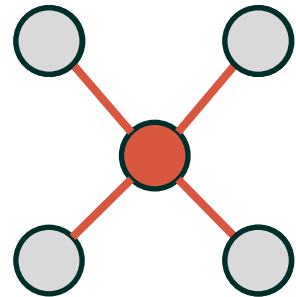
In each of these instances, ECMP+ONSET and Ripple+ONSET completely mitigate congestion loss from the attack.



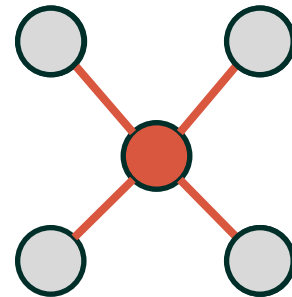
ONSET Evaluation: Attacks

Crossfire Attack

- Concentrates on attacking links adjacent to a specific node.
- We look at crossfire attacks targeting **Each node in the network** individually with an attack volume of **100 or 200 Gbps** per link.



ONSET Evaluation: Crossfire Attack



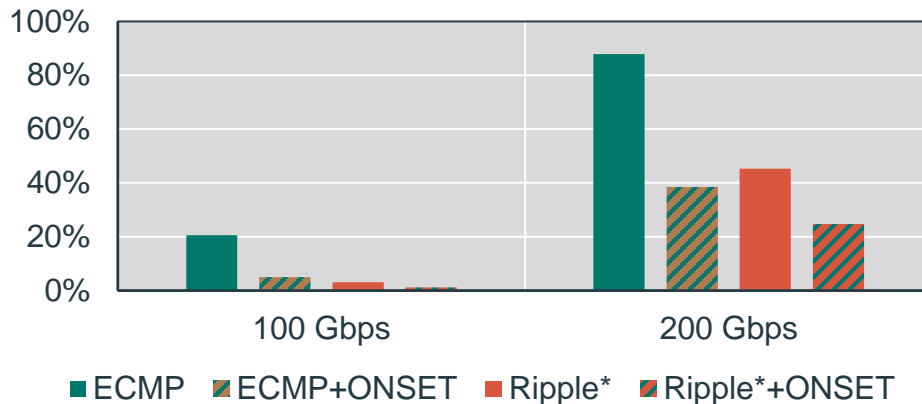
- 20% of the 100 Gbps attacks result in congestion loss with ECMP.

ONSET reduces the number to 5%.

- 45% of the 200 Gbps attacks result in congestion loss with Ripple*.

ONSET reduced the number to 25%.

Attacks Resulting in Congestion Loss



ONSET Conclusion

ONSET has the power to augment defenses for Link Flood Attacks.

It can be applied as a complement to existing SoTA defenses like Ripple.

It can also be applied to legacy networks with ECMP-based routing schemes.



Wrapping Up



Summary

- We developed a framework for optical topology programming, with foundational aspects that include an efficient method for optimizing topology.
- We conducted a measurement study of long-haul optical fiber hardware to benchmark optical topology programming link reconfiguration times.
- We developed a simulator to apply this framework with applications to challenging areas of ongoing study.
- Our framework can:
 - Scale traffic engineering systems to provide protection against link failures and flash-crowds.
 - Provide a means to quickly and proactively defend against network reconnaissance.
 - Improve capabilities of systems for defense against link flood attacks.

Future

- Optical Topology Programming as still an area with lots of potential for growth.
- A live network testbed to test the performance of real applications on top of an OTP backbone would help uncover some of the more subtle risks associated with OTP, e.g., performance impact on transport and application layer protocols.
- Systems challenges for the design of a stable optical/IP controller have only begun to be addressed.
 - E.g., a transactional framework for adding/removing links and all the intricacies associated for amplifier gain control, power adjustment, consistent state representation, etc.
- Network traffic and topology simulation.
 - A “digital twin” of the optical network.
 - Dynamically optimizing network topology for large AI/ML training workloads.

Achievements

- Published 9 peer-reviewed papers
- Erwin & Gertrude Juilfs Scholarship 2019
- Ripple Cyber-security fellow 2019
- Bell Labs Summer Research Award for Distinguished Innovation 2020
- University of Oregon Doctoral Research fellow 2022
- Worked on 700 km optical fiber span and reduced optical link addition time 20x
- Internship work at Bell Labs now running in their production network

Thank you!



References

[1] <https://www.dihuni.com/2020/04/10/every-day-big-data-statistics-2-5-quintillion-bytes-of-data-created-daily/>

[2] <https://www.itu.int/itu-d/reports/statistics/2023/10/10/ff23-internet-traffic/>

[3] <https://blogs.gartner.com/andrew-lerner/2014/07/16/the-cost-of-downtime/>