GradientGraph: A Network Optimization Framework for High-Precision Analysis of Bottleneck and Flow Performance

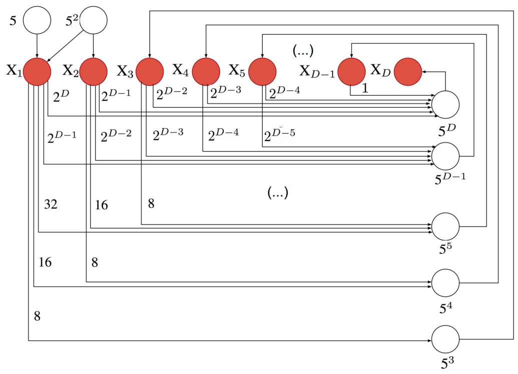
4th SIG-NGN Meeting
January 16, 2019

Reservoir Labs

Jordi Ros-Giralt, Sruthi Yellamraju, Atul Bohara, Harper Langston, Rich Lethin (Reservoir Labs)

Research Collaborators: Yuang Jiang, Leandros Tassiulas (Yale University)

Project partially funded by DOE under contract DE-SC0019523

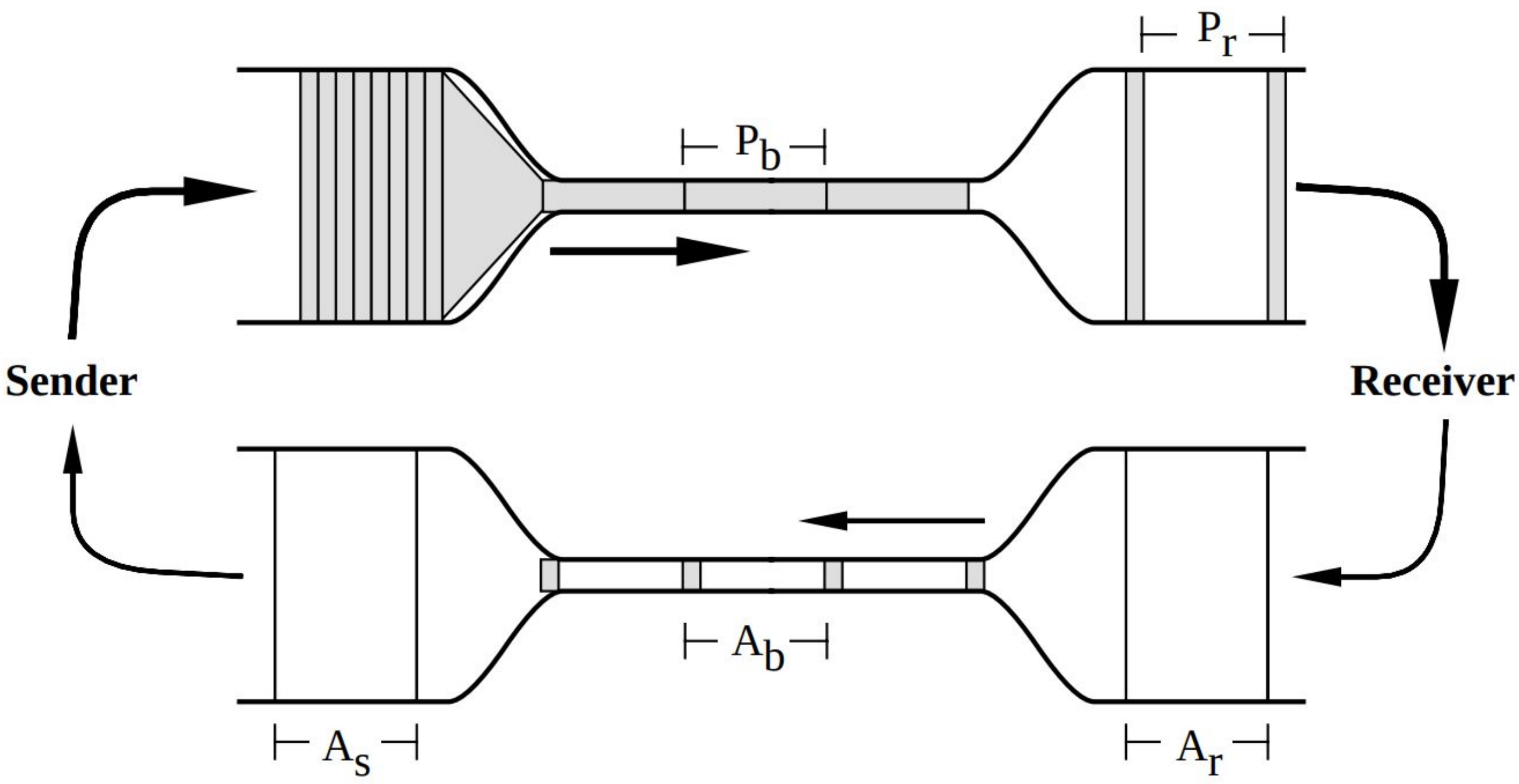


GradientGraph (G2) Analytics

- Objective: Flow performance optimization in very high speed networks. Bring today's network utilization from 30% to 90%.
- G2 provides a new level of understanding of the bottleneck structure of networks and the interactions between bottlenecks and flows.
- Applicable to: R&N Networks (e.g., ESnet), large scale data centers (e.g., Google Jupiter), cloud (e.g., AWS), SDN-WAN (e.g., Google B4), Supercomputers (e.g., NERSC Cori), the Internet itself.
- Some examples of problems G2 can resolve: scheduling of deadline-bound flows, flow admission control, bandwidth tapering and bandwidth steering, flow optimization in multi-domain / heterogeneous networks, network baselining and predictive modeling, multi-resource modeling (link, storage and compute), capacity planning.
- Status:
  - Technology (prototype level) demonstrated live at SC19 / SCinet.
  - Mathematics to be presented at ACM SIGMETRICS, June 2020.

Conventional view

Figure 1: Window Flow Control 'Self-clocking'



[*] Van Jacobson, "Congestion Avoidance and Control," SIGCOMM computer communication review 18, 4 (August 1988), 314–329

Conventional view

Regardless of how many links a connection traverses or what their individual speeds are, from TCP's viewpoint an arbitrarily complex path behaves as a single link with the same RTT (round-trip time) and bottleneck rate. Two

[*] Neal Cardwell, Yuchung Cheng, C. Stephen Gunn, Soheil Hassas Yeganeh, Van Jacobson, "BBR: Congestion-Based Congestion Control," ACM Queue, Dec 2016.

Are all Elephant Flows Heavy Hitters?

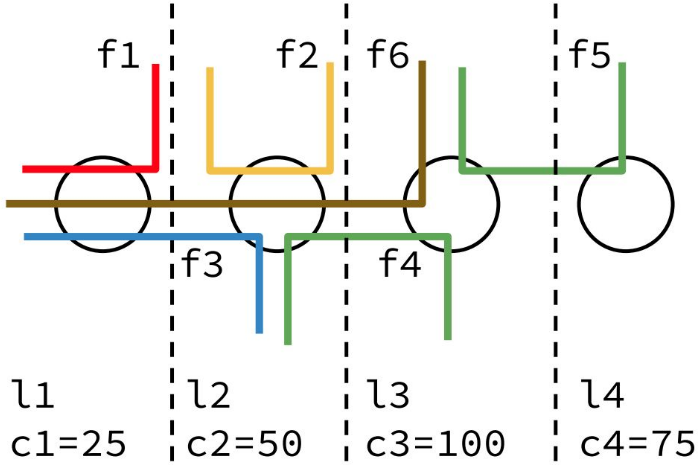
- Suppose N is a network with 6 TCP flows that receive this rate allocation vector: $\mathbf{r} = [8.3, 16.6, 8.3, 16.6, 75, 8.3]$ Mbps
- Which is the largest (elephant) flow?

Are all Elephant Flows Heavy Hitters?

- Suppose N is a network with 6 TCP flows that receive this rate allocation vector: $\mathbf{r} = [\underbrace{8.3}_{f_1}, \underbrace{16.6}_{f_2}, \underbrace{8.3}_{f_3}, \underbrace{16.6}_{f_4}, \underbrace{75}_{f_5}, \underbrace{8.3}_{f_6}]$ Mbps
- Which is the largest (elephant) flow?

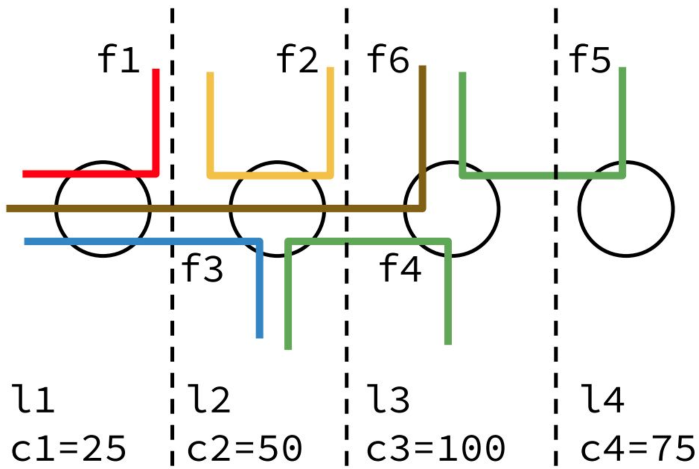
Are all Elephant Flows Heavy Hitters?

- Suppose N is a network with 6 TCP flows that receive this rate allocation vector: $\mathbf{r} = [\underset{f_1}{\boxed{8.3}}, \underset{f_2}{\boxed{16.6}}, \underset{f_3}{\boxed{8.3}}, \underset{f_4}{\boxed{16.6}}, \underset{f_5}{\boxed{75}}, \underset{f_6}{\boxed{8.3}}]$ Mbps
- Which is the largest (elephant) flow?

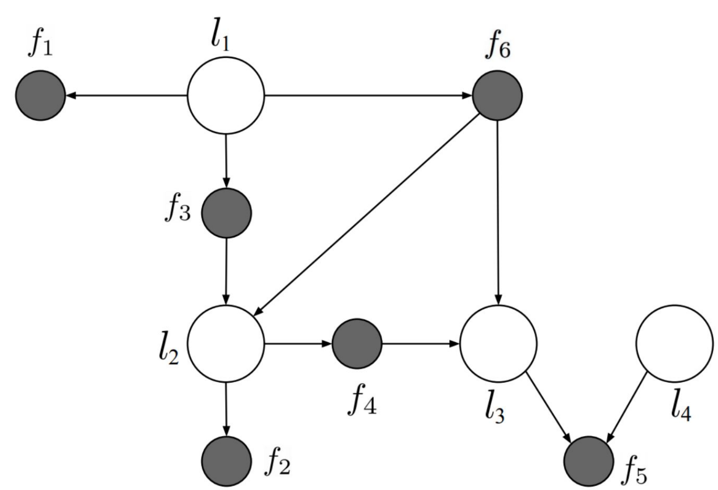


Are all Elephant Flows Heavy Hitters?

- Suppose N is a network with 6 TCP flows that receive this rate allocation vector: $\mathbf{r} = [\underset{f_1}{8.3}, \underset{f_2}{16.6}, \underset{f_3}{8.3}, \underset{f_4}{16.6}, \underset{f_5}{75}, \underset{f_6}{8.3}]$ Mbps
- Which is the largest (elephant) flow?

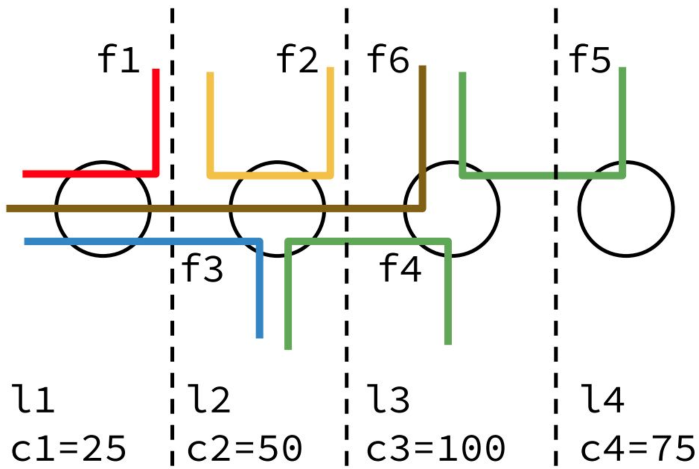


Flow Gradient Graph:

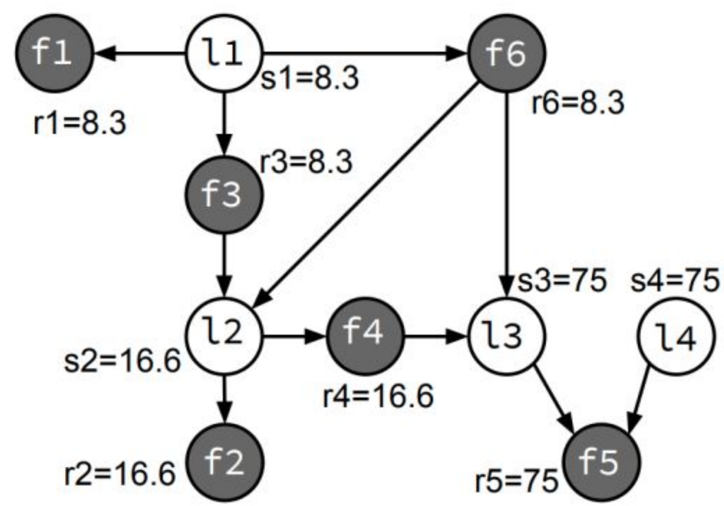


Are all Elephant Flows Heavy Hitters?

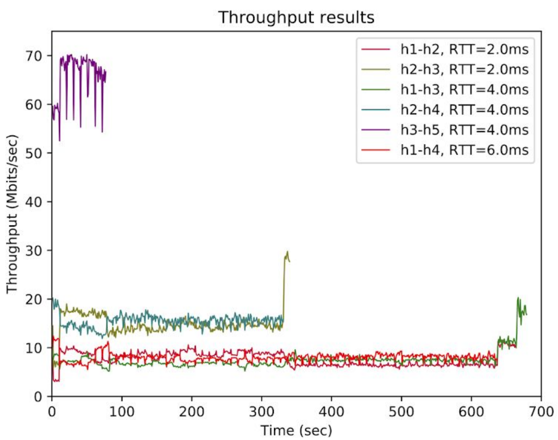
- Suppose N is a network with 6 TCP flows that receive this rate allocation vector: $\mathbf{r} = [\underset{f_1}{8.3}, \underset{f_2}{16.6}, \underset{f_3}{8.3}, \underset{f_4}{16.6}, \underset{f_5}{75}, \underset{f_6}{8.3}]$ Mbps
- Which is the largest (elephant) flow?



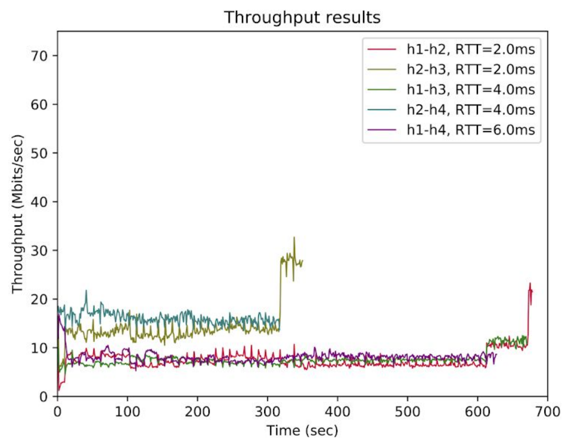
Flow Gradient Graph:



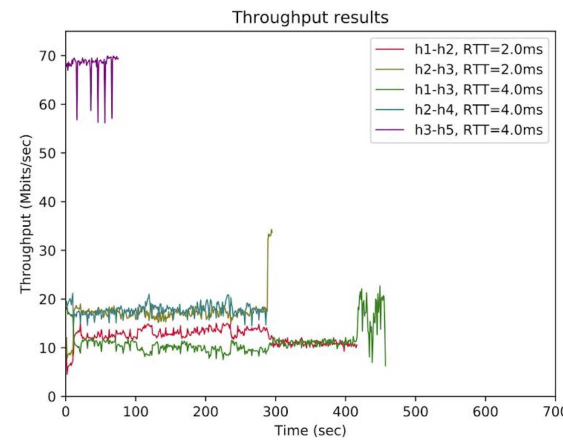
Are all Elephant Flows Heavy Hitters?



(a) Without removing any flow.



(b) Removing the heavy-hitter flow f_5.



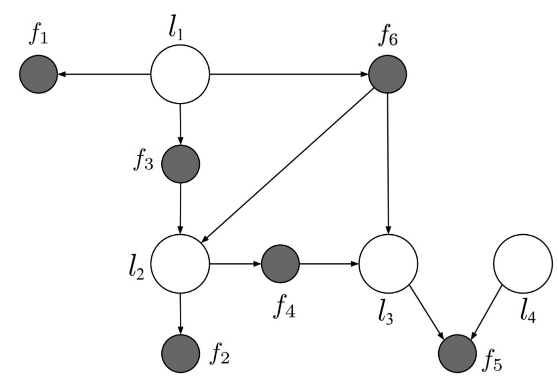
(c) Removing a low-hitter flow f_6.

Table 3: As predicted by the theory of bottleneck ordering, flow f_6 is a significantly higher impact flow than flow f_5.

Comp. time (secs)	f_1	f_2	f_3	f_4	f_5	f_6	Slowest
With all flows	664	340	679	331	77	636	679
Without flow f_5	678	350	671	317	—	611	678
Without flow f_6	416	295	457	288	75	—	457

Avg rate (Mbps)	f_1	f_2	f_3	f_4	f_5	f_6	Total
With all flows	7.7	15.1	7.5	15.4	65.8	8.1	119.6
Without flow f_5	7.5	14.5	7.6	16.1	—	8.3	54
Without flow f_6	12.2	17.2	11.1	17.7	68.1	—	126.3

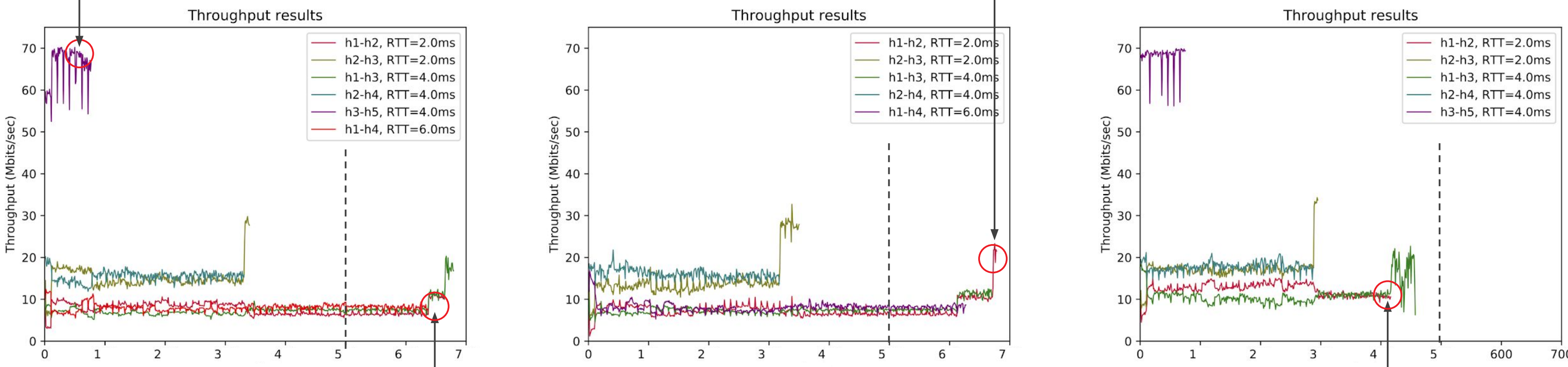
Flow Gradient Graph:



Operational Use Case: Scheduling of Deadline-Bound Data Transfers

(2) Traditional approach: look at heavy hitters

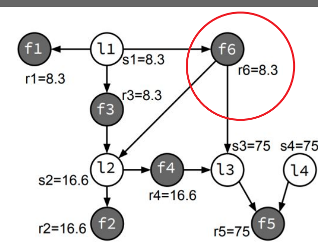
(3) Traditional approach is unable to help



(a) Without removing any flow. **(b) Removing the heavy-hitter flow f_5.** **(c) Removing a low-hitter flow f_6.**

(1) Goal: deliver red flow (h1-h2) by 5 am, two hours ahead

(4) GradientGraph reveals the solution to meet the deadline-bound constraint



Problem Scope and Relevance

[Slide taken from Bill Johnston's talk at ASCAC19: "ESnet: Advanced Networking for Data-Intensive Science"]

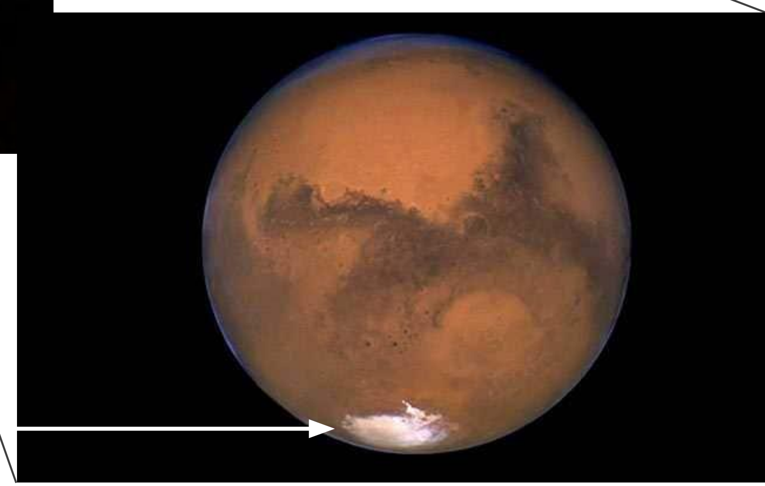
LHCONE: Not all big data traffic is suitable for the general Internet

- As the LHC ramped up to first production operation, ESnet monitoring detected several transatlantic network paths serving the R&E community were being congested
- Finding the cause was not trivial because it turned out to be LHC data analysis groups moving data with GridFTP using dozens of parallel data transfers, so no one end system stood out in the monitoring
- ESnet engaged CERN on how to deal with this, and CERN set up a study group to characterize the problem
- CERN, ESnet, and Internet2 to set up a working group to make recommendations on how to address this issue
  - ESnet engineers proposed a **network overlay approach where the paths used by the overlay were explicitly under control of network operators**
    - In other words, the paths could be easily configured by network engineers not to interfere with general R&E traffic in their domain
    - Access to the overlay was limited to high energy physics projects, which also provided a modicum of security
- The result is called **LHCONE** and carries most of the LHC data worldwide
  - [13] See http://lhcone.web.cern.ch

ESnet

Naked eye view is nice…
But insufficient to understand bottlenecks and flows

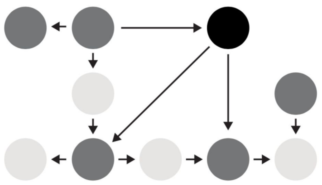
Towards an intimate understanding of bottlenecks and flows



Water at 1 mile below Mars' surface

Networks have Bottleneck Structures...
and they can be computed in polynomial time

GradientGraph Analytics



Networks have Bottleneck Structures...
and they can be computed in polynomial time

Algorithm 1 BPG

$\mathcal{L}^0 = \mathcal{L}; \mathcal{C}^0 = \{\emptyset\};$
$\mathcal{D}_l^0 = \mathcal{I}_l^0 = \mathcal{R}_l^0 = \{\emptyset\}, \forall l \in \mathcal{L};$
$k = 0;$
while $\mathcal{C}^k \neq \mathcal{F}$ **do**
  $s_l^k = (c_l - \sum_{\forall f \in \mathcal{C}^k \cap \mathcal{F}_l} r_f)/|\mathcal{F}_l \setminus \mathcal{C}^k|, \forall l \in \mathcal{L}^k;$
  $u_l^k = min\{s_{l'}^k \mid \mathcal{F}_{l'} \cap \mathcal{F}_l \neq \{\emptyset\}, \forall l' \in \mathcal{L}^k\}, \forall l \in \mathcal{L}^k;$
  for $l \in \mathcal{L}^k, s_l^k = u_l^k$ **do**
    $r_f = s_l^k, \forall f \in \mathcal{F}_l;$
    $\mathcal{L}^k = \mathcal{L}^k \setminus \{l\};$
    $\mathcal{C}^k = \mathcal{C}^k \cup \{f, \forall f \in \mathcal{F}_l\};$
    for $l' \in \mathcal{L}^k, \mathcal{F}_{l'} \cap \mathcal{F}_l \neq \{\emptyset\}$ **do**
      $\mathcal{D}_{l'}^k = \mathcal{D}_{l'}^k \cup l;$
    end for
    for $l', l_r \in \mathcal{L}^k, \mathcal{F}_{l'} \cap \mathcal{F}_{l_r} \neq \{\emptyset\}, s_{l_r}^k < s_{l'}^k$ **do**
      $\mathcal{R}_{l'}^k = \mathcal{R}_{l'}^k \cup \{l_r\};$
    end for
    for $l' \in \mathcal{D}_{l_r}^k \setminus \mathcal{D}_l^k, l_r \in \mathcal{R}_l^k \setminus \mathcal{D}_l^k$ **do**
      $\mathcal{I}_l^k = \mathcal{I}_l^k \cup \{l'\};$
    end for
  end for
  $\mathcal{L}^{k+1} = \mathcal{L}^k; \mathcal{C}^{k+1} = \mathcal{C}^k;$
  $\mathcal{D}_l^{k+1} = \mathcal{D}_l^k; \mathcal{I}_l^{k+1} = \mathcal{I}_l^k; \mathcal{R}_l^{k+1} = \mathcal{R}_l^k;$
  $k = k + 1;$
end while
$\mathcal{B} = \mathcal{L} \setminus \mathcal{L}^k;$
$\mathcal{P} = \{\mathcal{D}_l^k, \forall l \in \mathcal{B}\} \cup \{\mathcal{I}_l^k, \forall l \in \mathcal{B}\};$
return $\langle \mathcal{B}, \mathcal{P} \rangle;$

GradientGraph Analytics

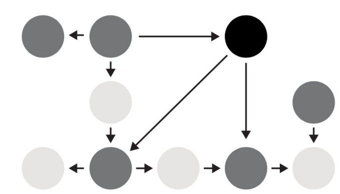


TABLE I: *Notations used in the BPG algorithm [5].*

Variable	Definition
$\mathcal{L}$	Set of links in the input network
$\mathcal{F}$	Set of flows in the input network
$\mathcal{F}_l$	Set of flows going through link l
c_l	Capacity of link l
s_l^k	Fair share of link l at iteration k
u_l^k	Upstream fair share of link l at iteration k
$\mathcal{L}^k$	Set of unresolved links at iteration k
$\mathcal{C}^k$	Set of converged flows at iteration k
$\mathcal{D}_l^k$	Set of direct precedents of link l at iteration k
$\mathcal{I}_l^k$	Set of indirect precedents of link l at iteration k
$\mathcal{R}_l^k$	Set of relays of link l at iteration k
$\mathcal{B}$	Set of bottleneck links
r_f	Rate of flow f
$\setminus$	Set minus operator

Bottleneck Structure of Google's SDN WAN B4 Network

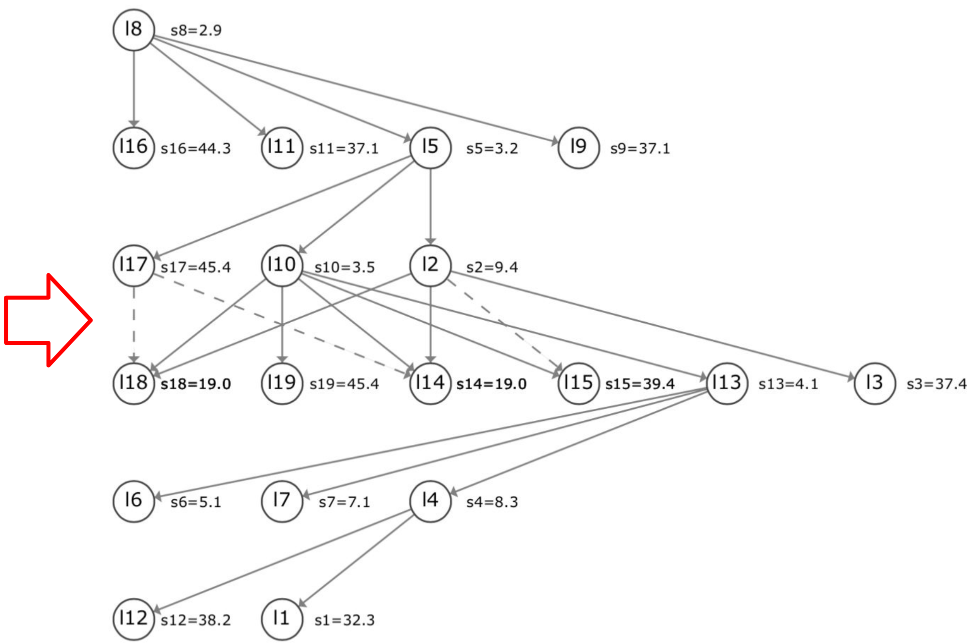
- Google's B4 Network:
  (from ACM SIGCOMM paper)

Bottleneck Structure of Google's SDN WAN B4 Network

- Google's B4 Network:
  (from ACM SIGCOMM paper)

- Bottleneck Structure of B4

  (shortest path full mesh configuration):



Theory of Bottleneck Ordering

LEMMA 2.4. *Bottleneck influence. A bottleneck l can influence the performance of another bottleneck l', i.e., $\partial s_{l'}/\partial c_l \neq 0$, if and only if there exists a set of bottlenecks $\{l_1, l_2, ..., l_n\}$ such that l_i is a direct precedent of l_{i+1}, for $1 \leq i \leq n-1$, $l_1 = l$ and $l_n = l'$.*

LEMMA 2.12. *Flow influence. A flow f can influence the performance of another flow f', i.e., $\partial r_{f'}/\partial r_f \neq 0$ if and only if there exists a set of bottlenecks $\{l_1, l_2, ..., l_n\}$ such that (1) l_i is a direct precedent of l_{i+1}, for $1 \leq i \leq n-1$, (2) flow f' is bottlenecked at link l_n and (3) flow f goes through l_1.*

Full details on the mathematics will be presented at ACM SIGMETRICS 2020

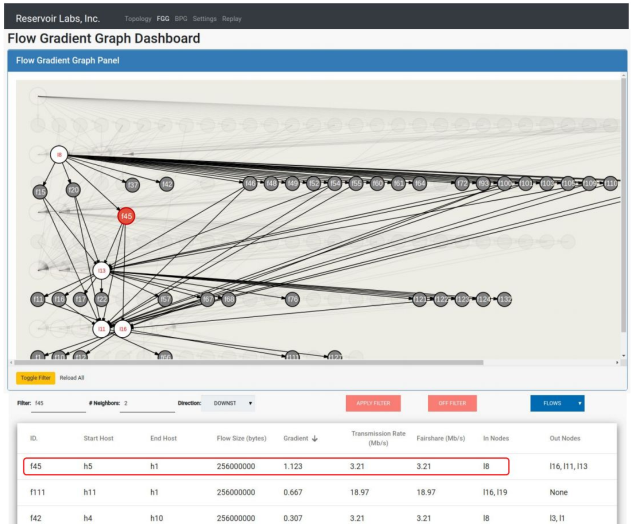
LEMMA 2.9. *Minimum convergence time of a distributed congestion control algorithm. Let $\tau(l_i, l_j)$ be a weight assigned to each edge (l_i, l_j) of the BPG graph as follows: (1) If l_i is a direct precedent of l_j, then $\tau(l_i, l_j)$ is the time that it takes for a message to be sent from l_i to l_j; (2) If l_i is an indirect precedent of l_j, then $\tau(l_i, l_j) = max\{\tau(l_i, l_r) + \tau(l_r, l_j) \mid$ for any relay link l_r between l_i and $l_j\}$. Let $l_1 - l_2 - ... - l_n$ be a longest path terminating at link l_n according to these weights. Then the minimum convergence time for link l_n is $\sum_{1 \leq i \leq n-1} \tau(l_i, l_{i+1})$.*

On the Bottleneck Structure of Congestion-Controlled Networks

Jordi Ros-Giralt[1], Sruthi Yellamraju[1], Atul Bohara[1], Harper Langston[1], Richard Lethin[1], Yuang Jiang[2], Leandros Tassiulas[2], Josie Li[3], Malathi Veeraraghavan[3]
[1] Reservoir Labs, 632 Broadway, Suite 803 New York, New York 10012
[2] Yale Institute of Network Science
[3] University of Virginia
{giralt,yellamraju,bohara,langston,lethin}@reservoir.com
{yuang.jiang,leandros.tassiulas}@yale.edu
{jl9gf,mv5g}@virginia.edu

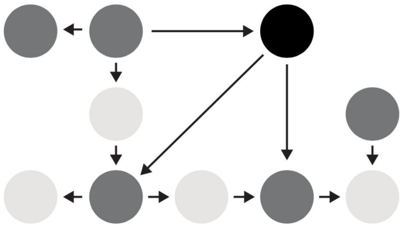
GradientGraph Analytics: Features and Functions

- Interactive analytical dashboards
- Computation of bottleneck structures
- Real-time traffic engineering recommendations
- Flow / resource allocation and scheduling
- Offline capacity planning suggestions
- Network performance baselining
- Locating routing misconfigurations
- Replay bottleneck structures



Experiments

GradientGraph Analytics



- Do networks behave according to their bottleneck structure?
- Can we use GradientGraph to Optimize Flow Performance?
- Can we use GradientGraph to Perform Capacity Planning?
- Can we use GradientGraph for Network Baselining?
- Does GradientGraph work under partial information? (multi-domain networks or lack of full network visibility)

Do Networks Act According to the Bottleneck Structure?

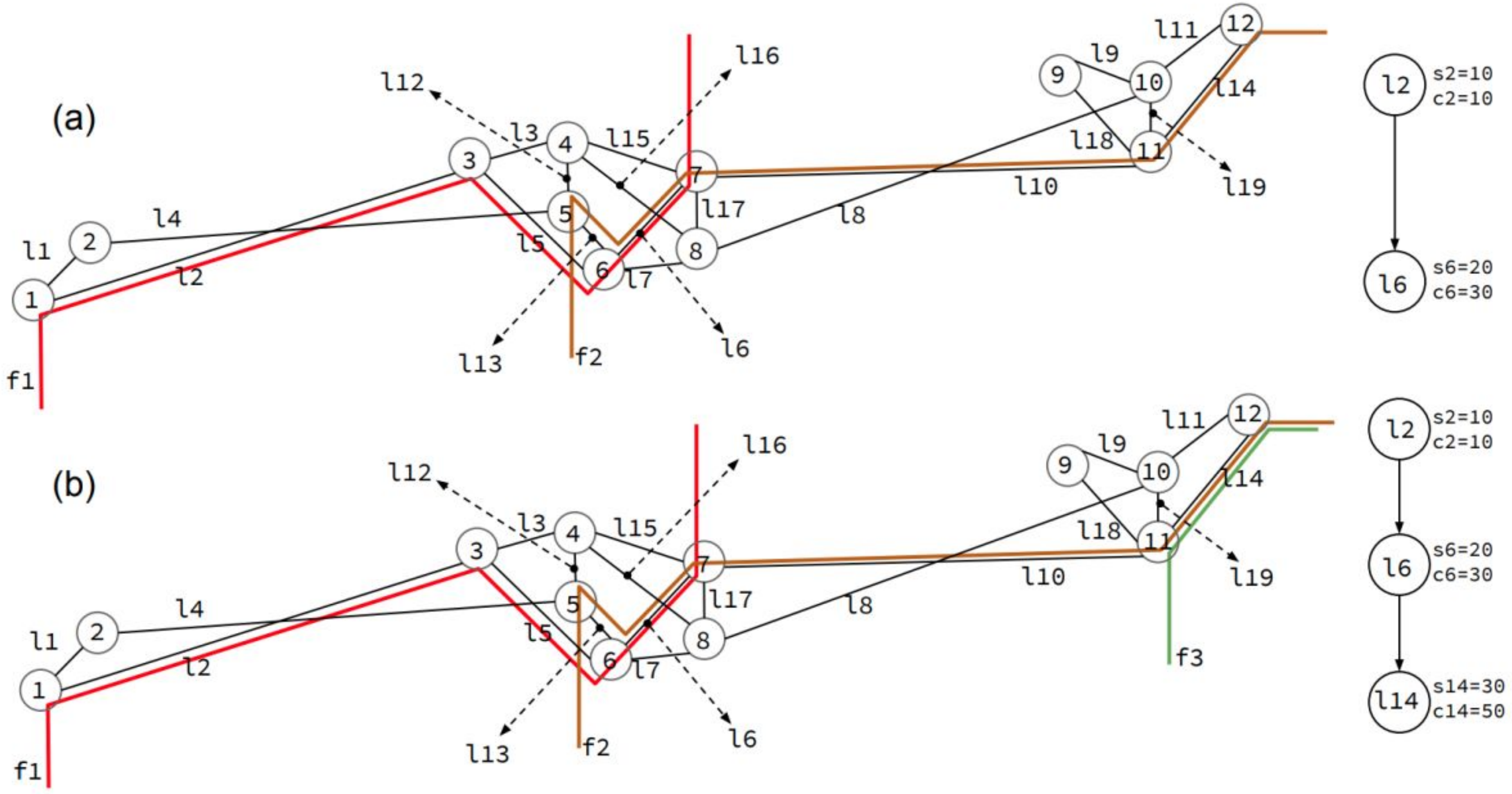
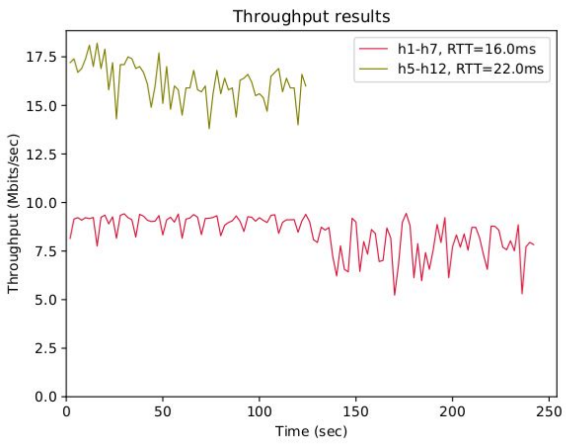
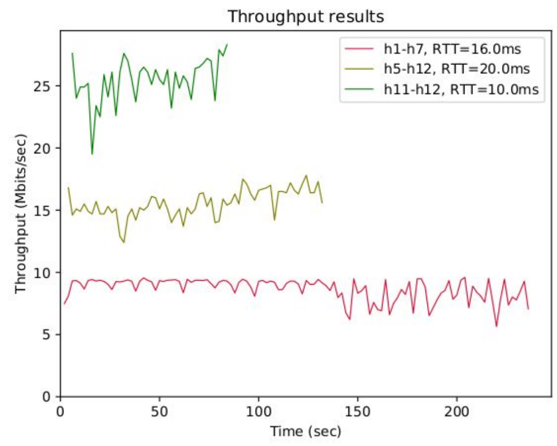


Fig. 10: Network configurations to benchmark (a) 2-level and (b) 3-level bottleneck structures.

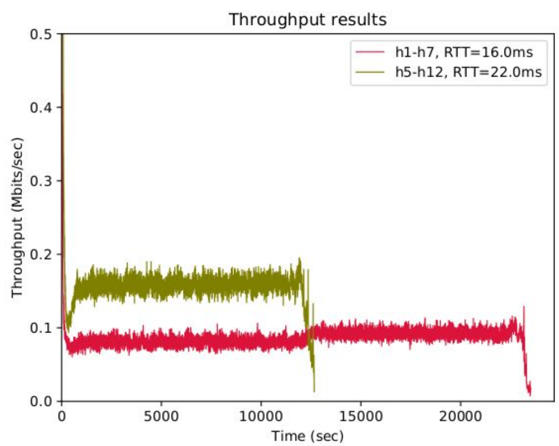
Do Networks Act According to the Bottleneck Structure?



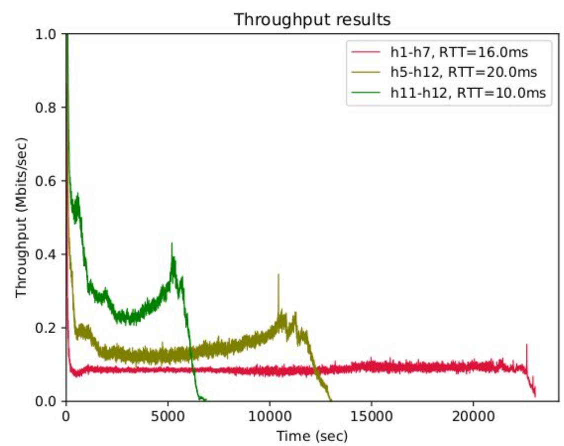
(a) 2-level / 2 BBR flows.



(b) 3-level / 3 BBR flows.

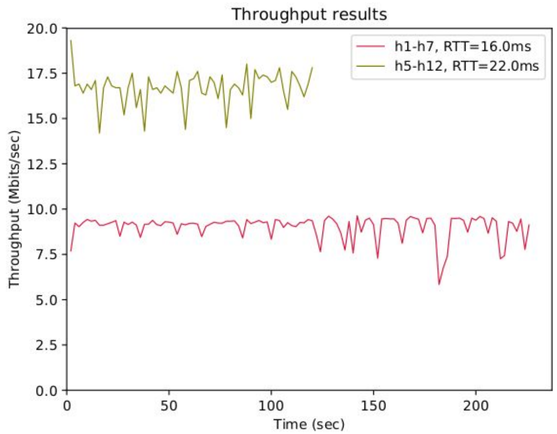


(c) 2-level / 200 BBR flows.

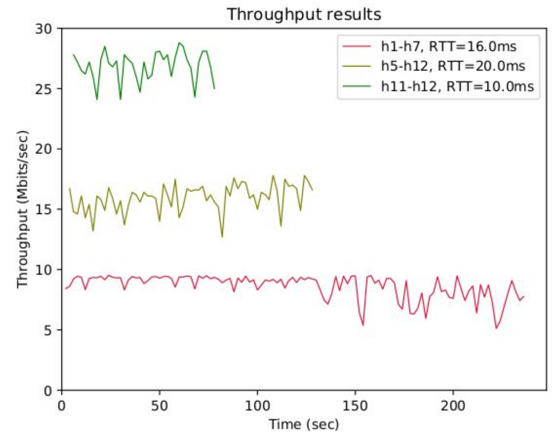


(d) 3-level / 300 BBR flows.

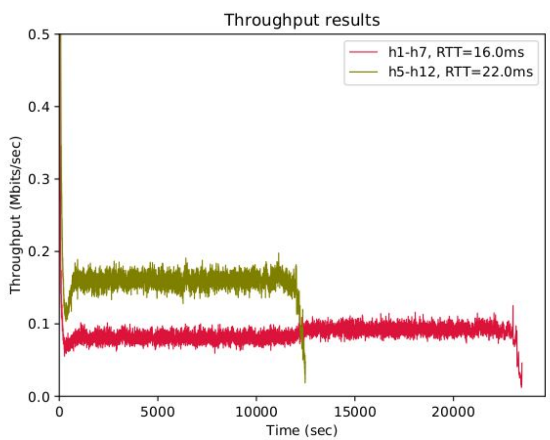
Do Networks Act According to the Bottleneck Structure?



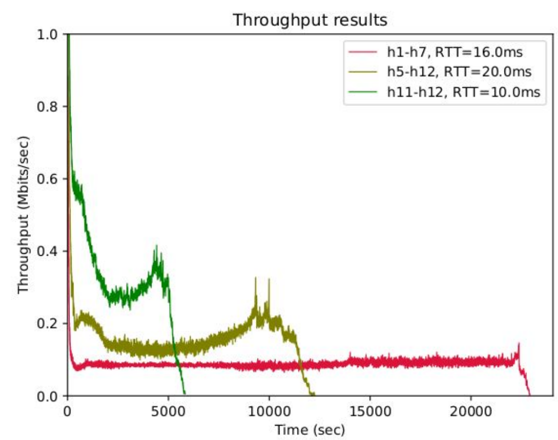
(e) 2-level / 2 Cubic flows.



(f) 3-level / 3 Cubic flows.

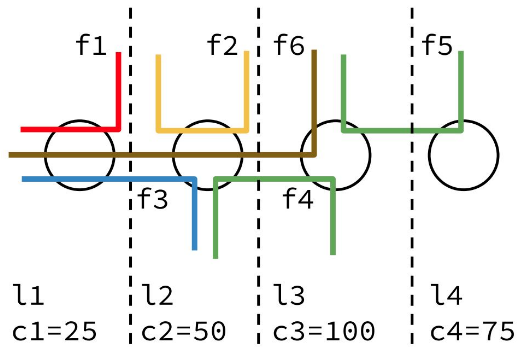


(g) 2-level / 200 Cubic flows.

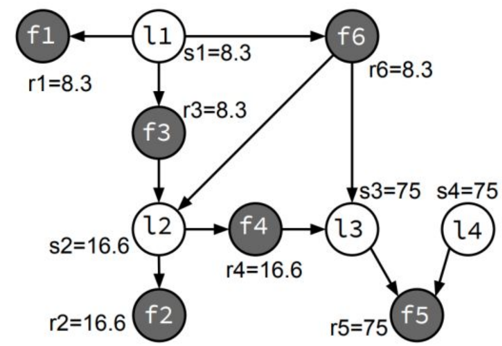


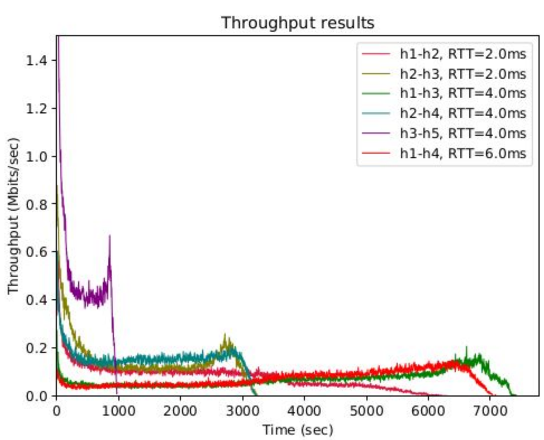
(h) 3-level / 300 Cubic flows.

Using GradientGraph to Optimize Flow Performance

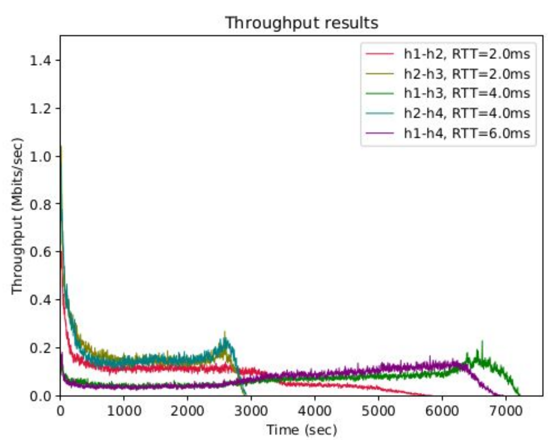


Flow Gradient Graph:

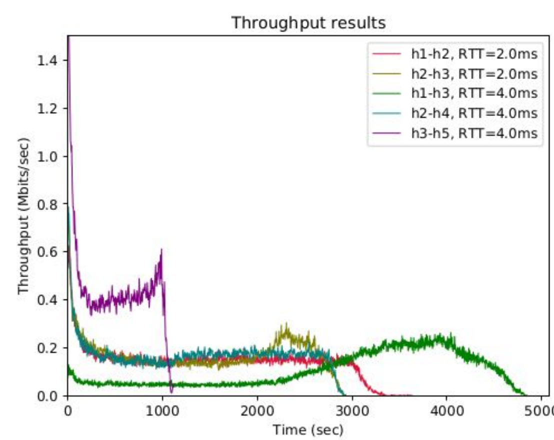




(d) Cubic without removing any flow.
(600 flows)

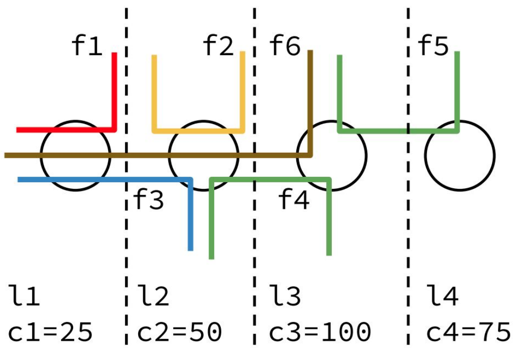


(e) Cubic removing flow f_5 and replicas.
(500 flows)

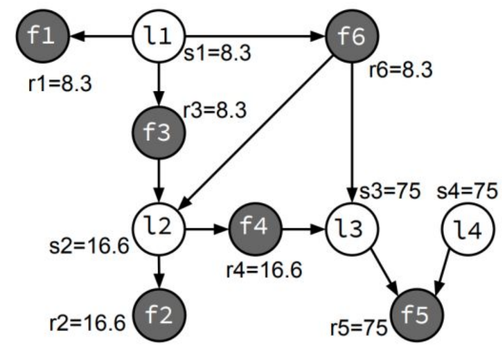


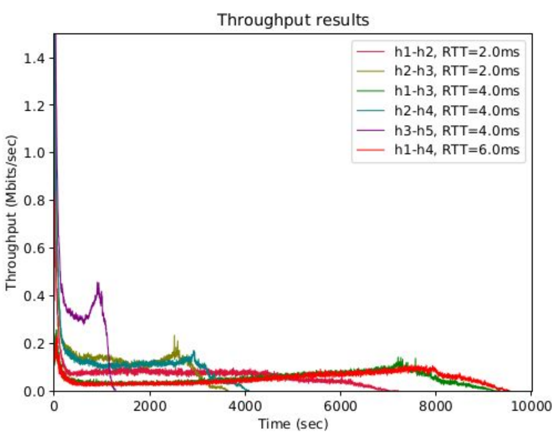
(f) Cubic removing flow f_6 and replicas.
(500 flows)

Using GradientGraph to Optimize Flow Performance

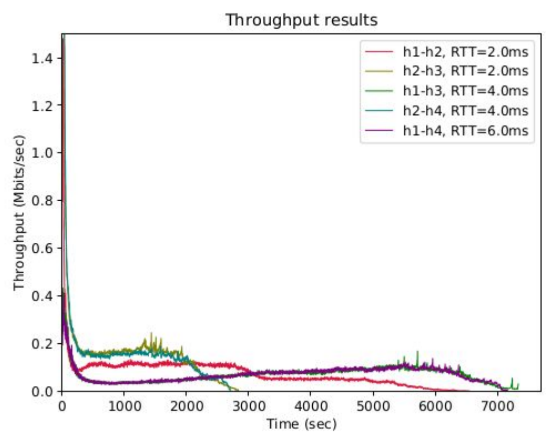


Flow Gradient Graph:

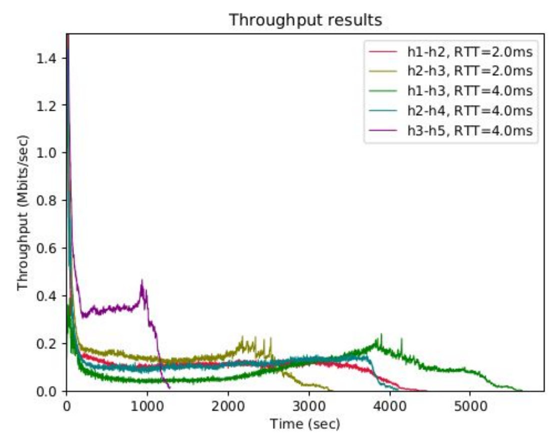




(a) BBR without removing any flow.

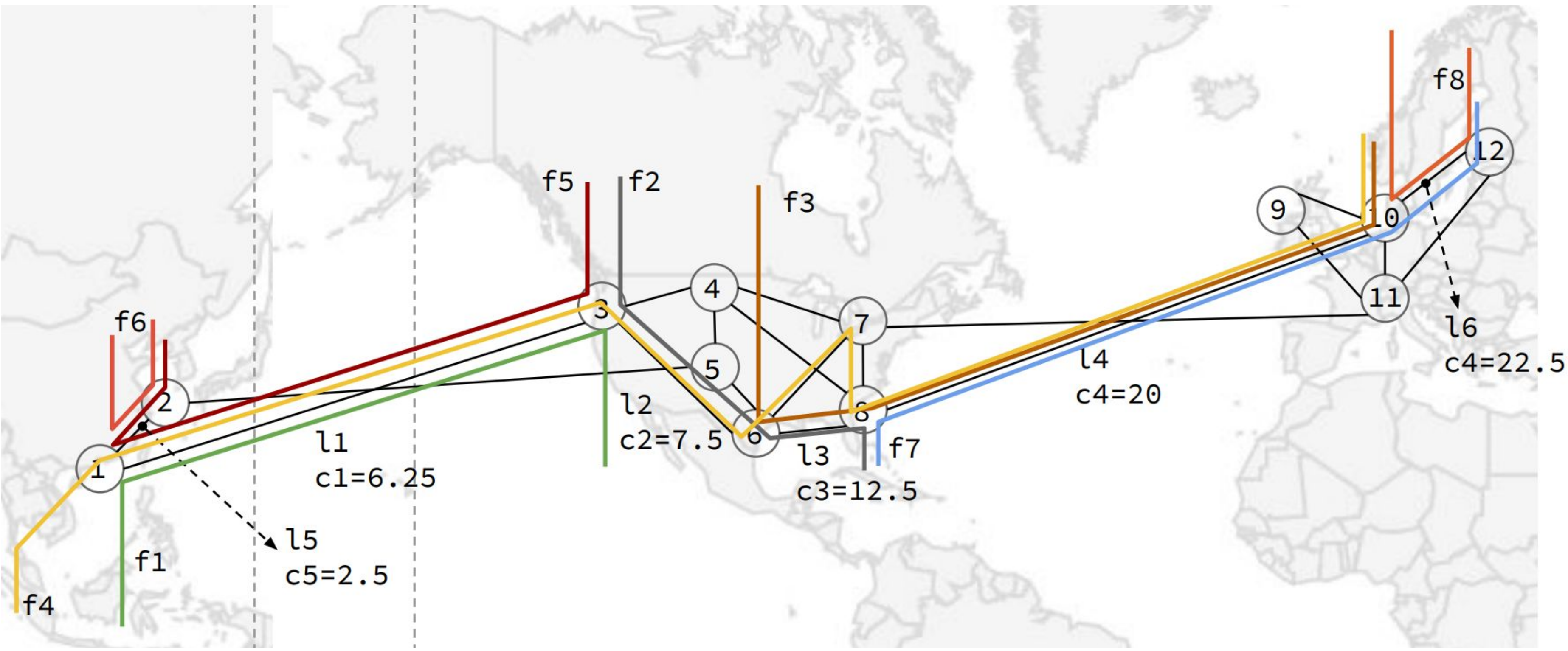


(b) BBR removing flow f_5 and replicas.

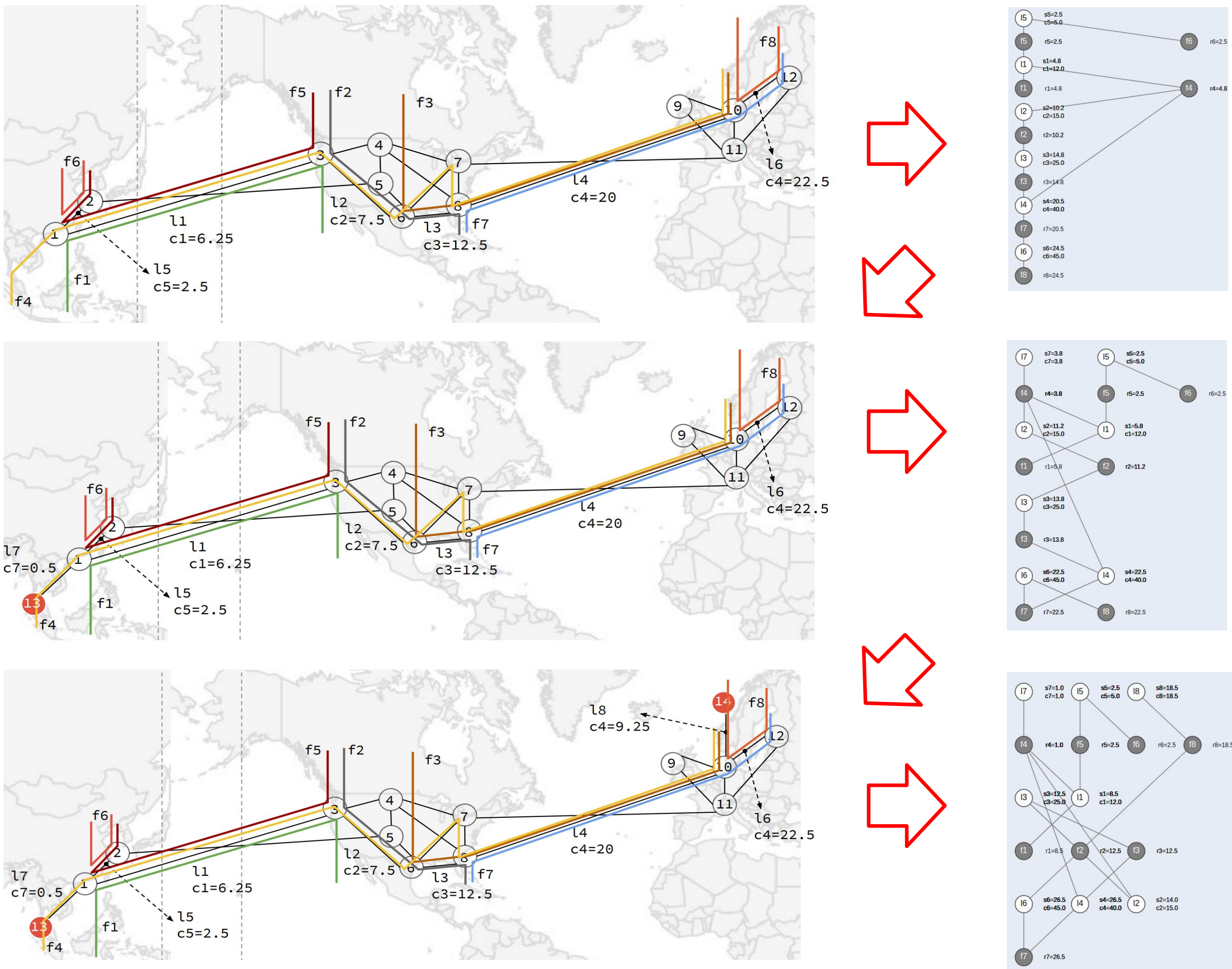


(c) BBR removing flow f_6 and replicas.

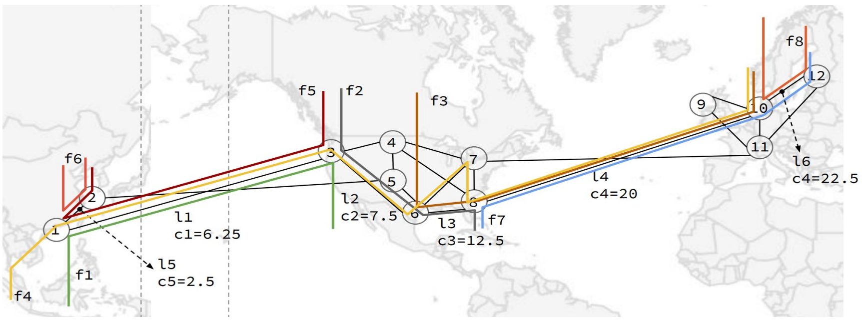
Using GradientGraph to Optimize Flow Performance

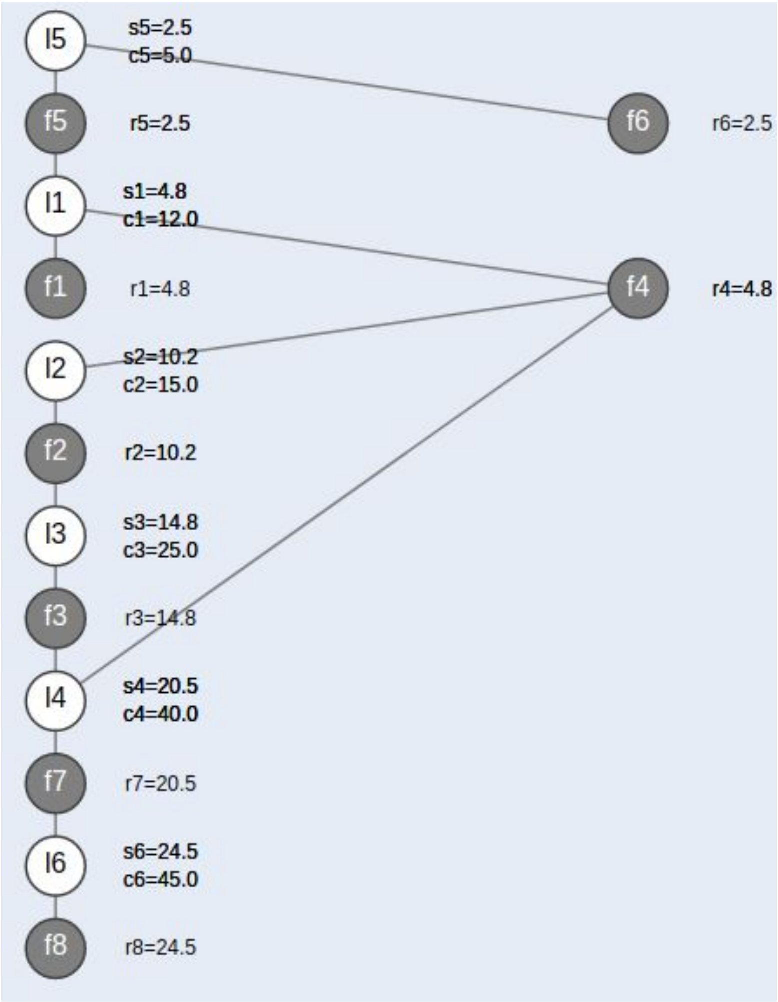


Using GradientGraph to Optimize Flow Performance

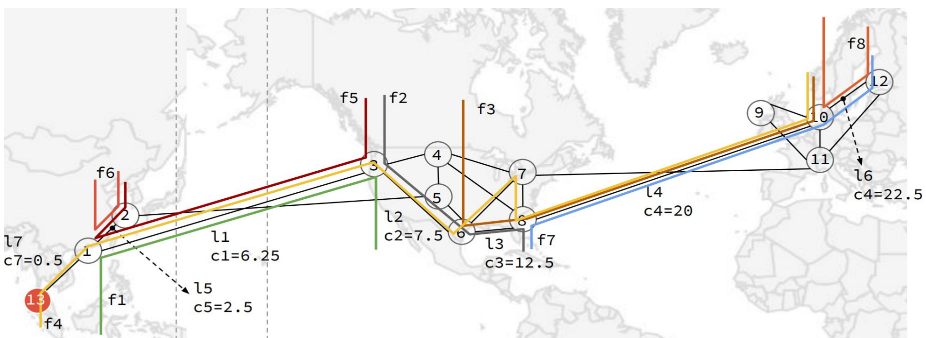


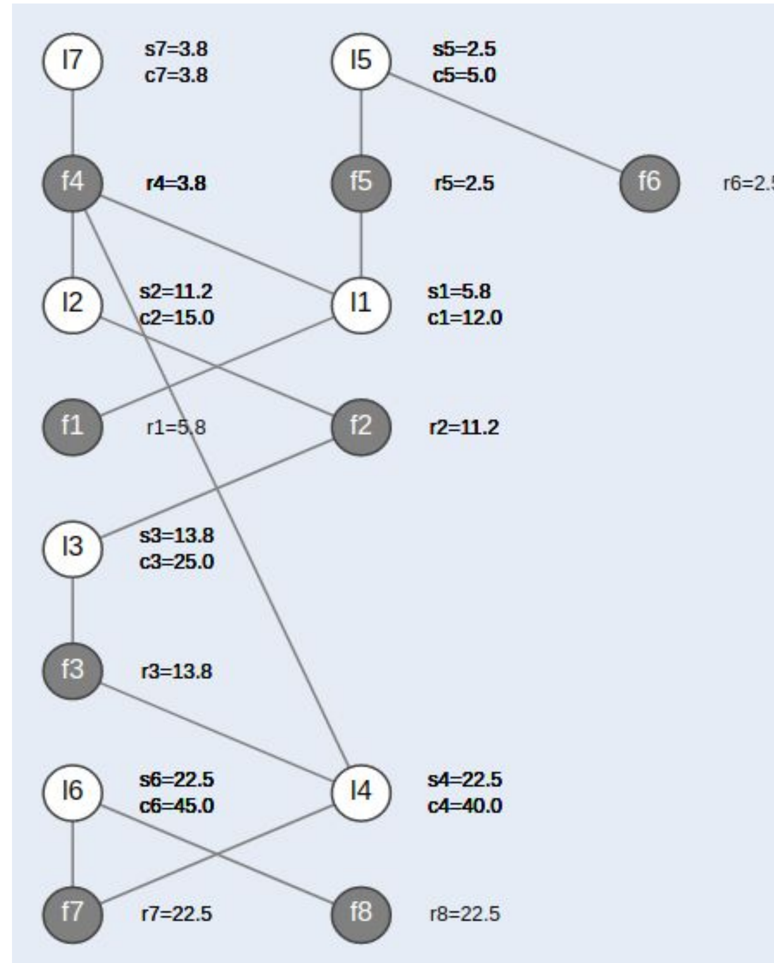
Using GradientGraph to Optimize Flow Performance



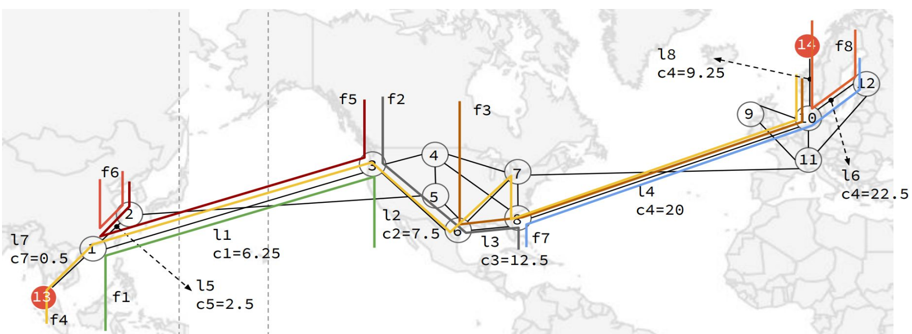


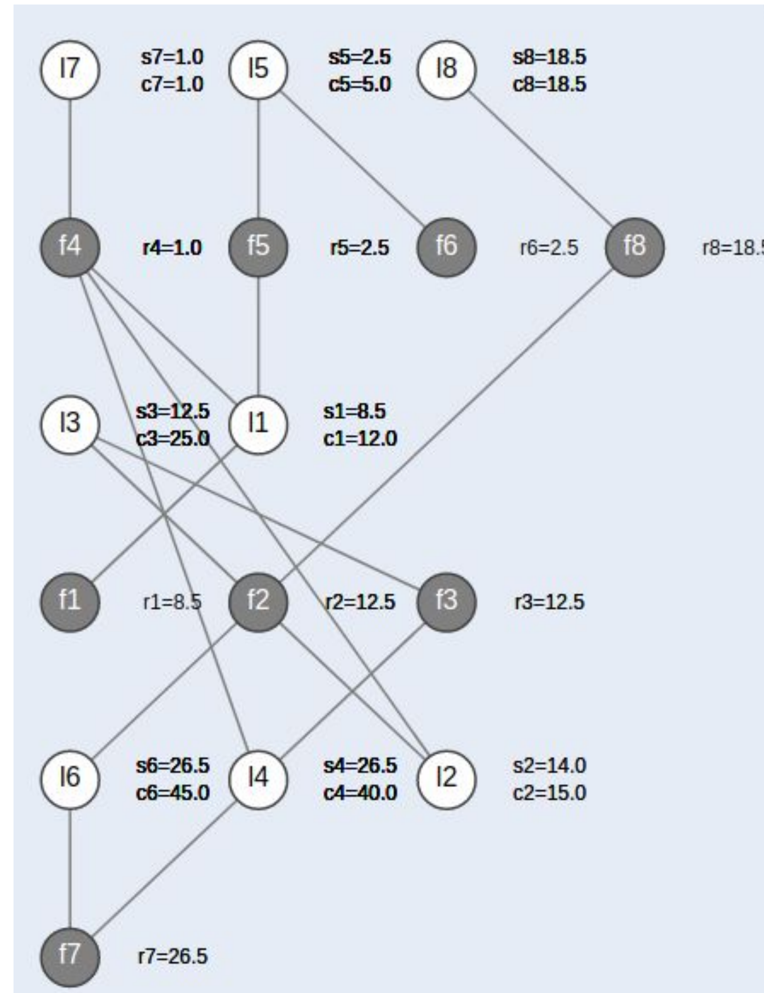
Using GradientGraph to Optimize Flow Performance



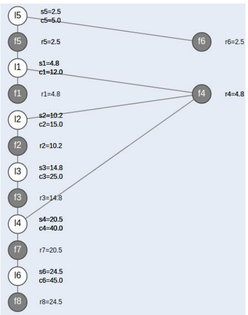


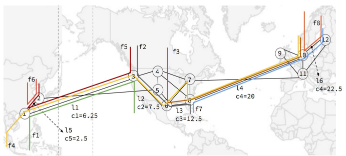
Using GradientGraph to Optimize Flow Performance

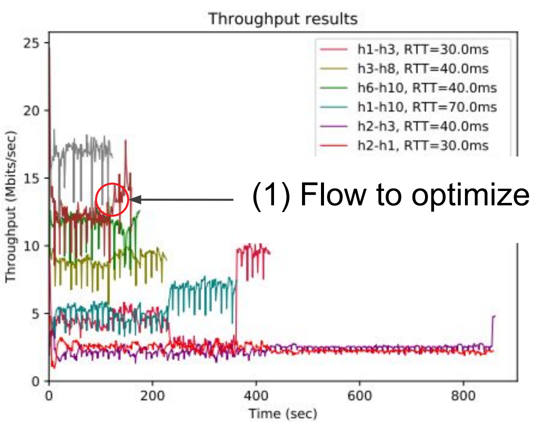




Using GradientGraph to Optimize Flow Performance

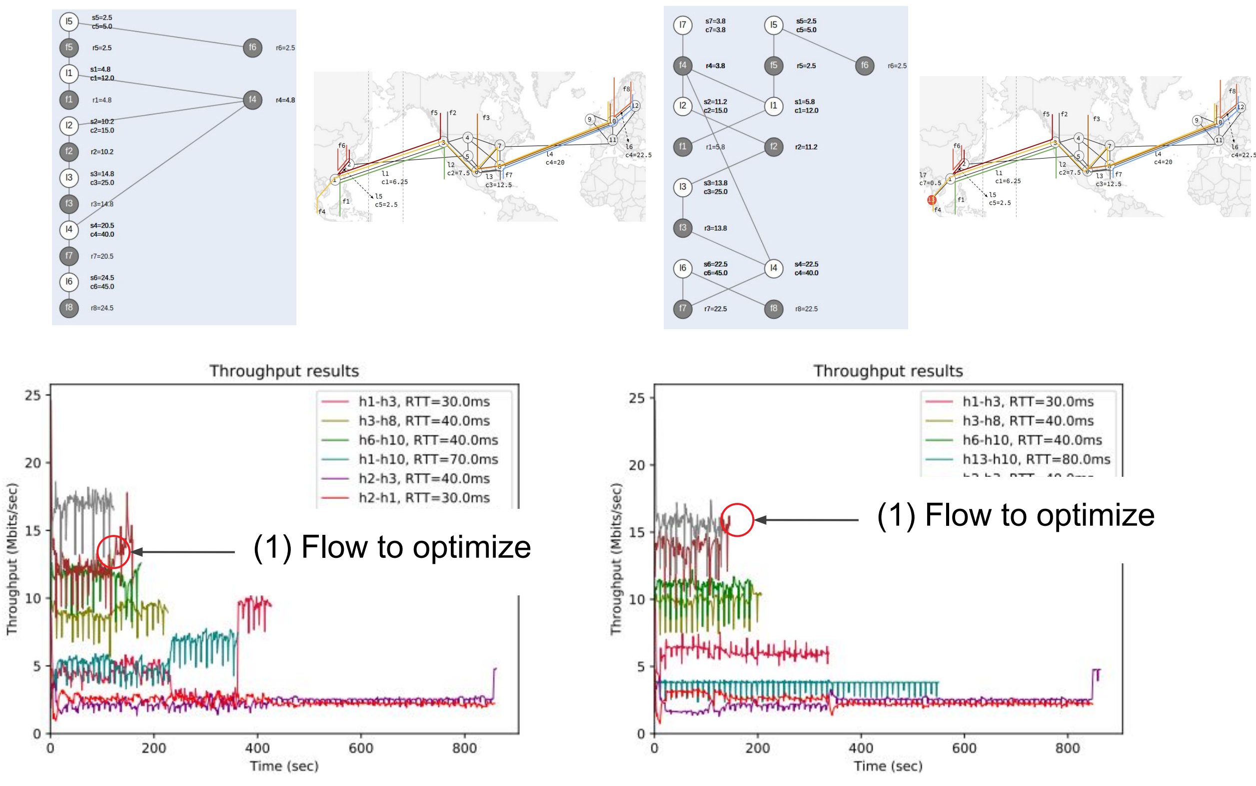






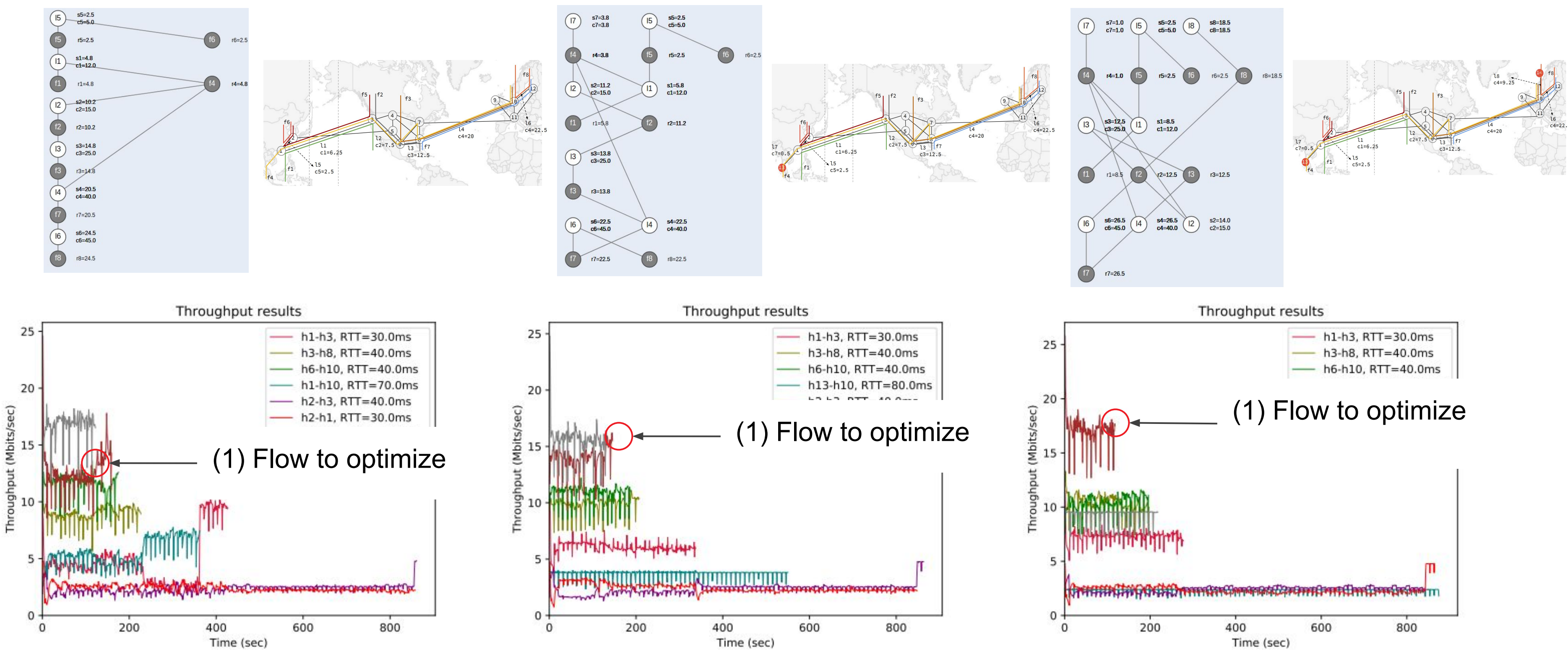
(a) No acceleration.

Using GradientGraph to Optimize Flow Performance



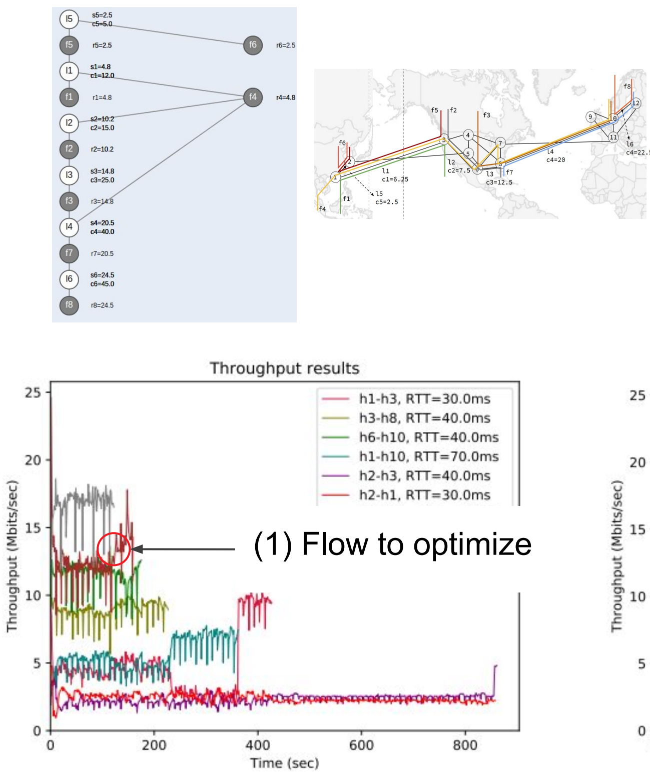
(a) No acceleration. **(b) Optimal solution shaping 1 flow.**

Using GradientGraph to Optimize Flow Performance

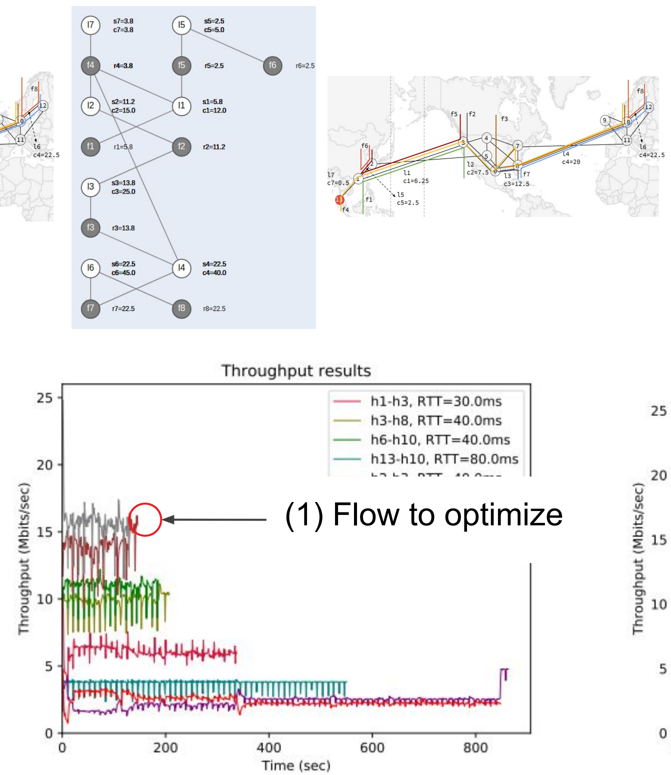


(a) No acceleration. **(b) Optimal solution shaping 1 flow.** **(c) Optimal solution shaping 2 flows.**

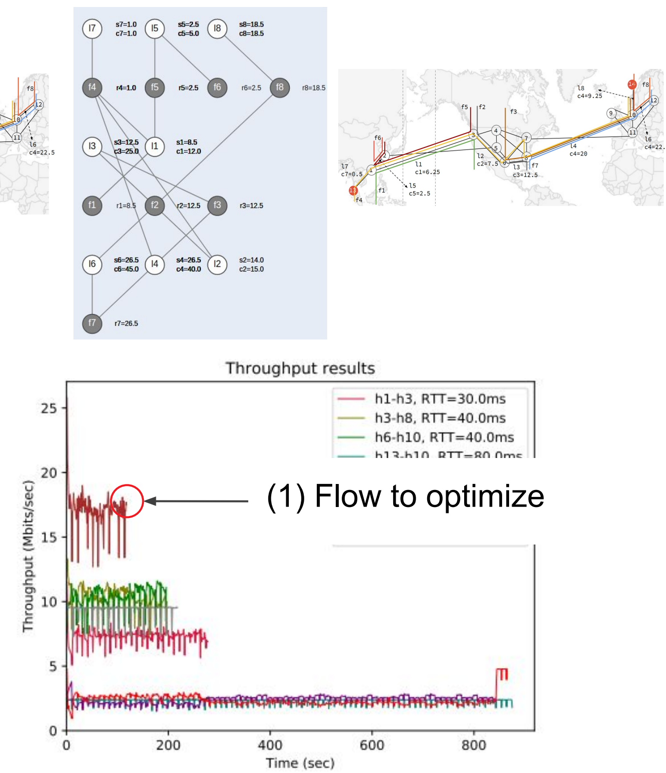
Using GradientGraph to Optimize Flow Performance



(a) No acceleration.



(b) Optimal solution shaping 1 flow.

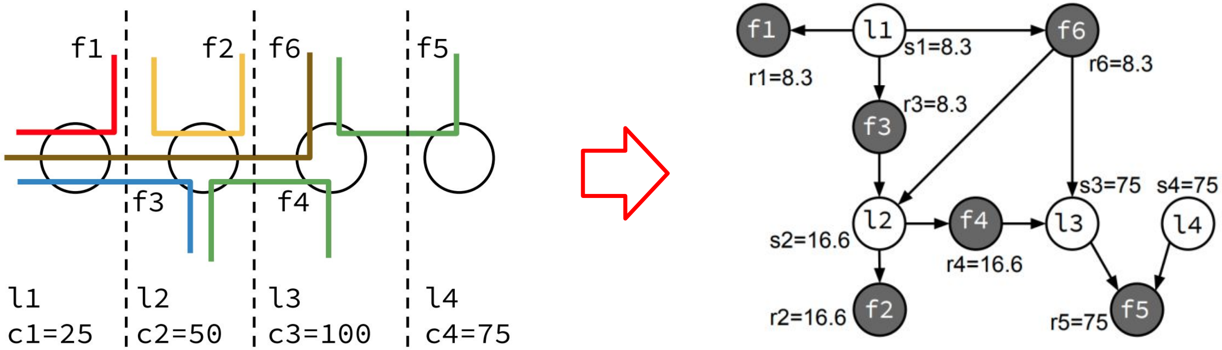


(c) Optimal solution shaping 2 flows.

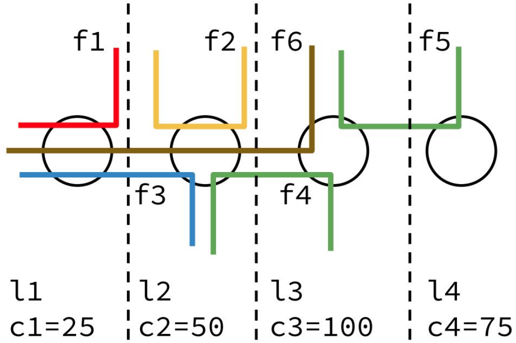
27% flow completion time reduction: The flow completing at 3pm will now complete before noon time.

Can we use GradientGraph to Perform Capacity Planning?

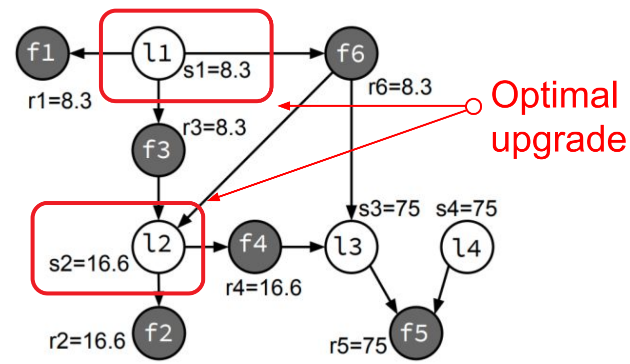
Flow Gradient Graph:



Can we use GradientGraph to Perform Capacity Planning?



Flow Gradient Graph:



Mathematical results:	$\nabla_{l_1}(N) = 1/3;\ \nabla_{l_2}(N) = 1/2;\ \nabla_{l_3}(N) = 0;\ \nabla_{l_4}(N) = 0;$

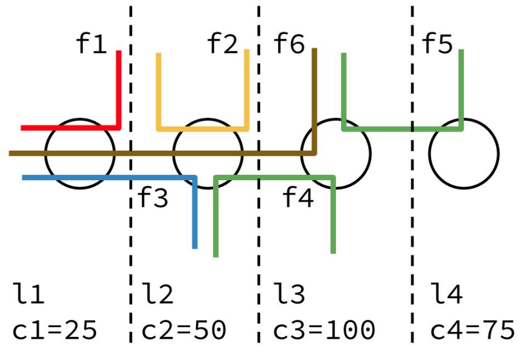
Experimental results:

TABLE VI: *Total throughput (Mbps) obtained when upgrading a link with 10 additional units of capacity.*

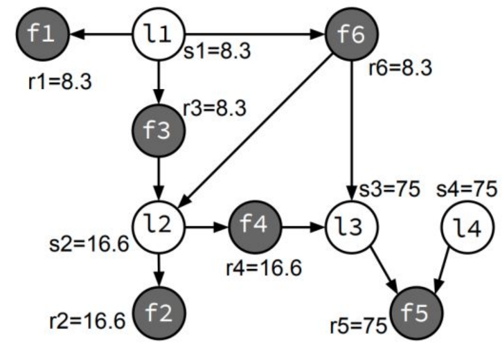
	BBR	Cubic
No links upgrade	119.9	119.6
Upgrade link l_1	128.46	137.88
Upgrade link l_2	126.17	178.59
Upgrade link l_3	122.88	123.54
Upgrade link l_4	120.67	119.10

Positive gradients

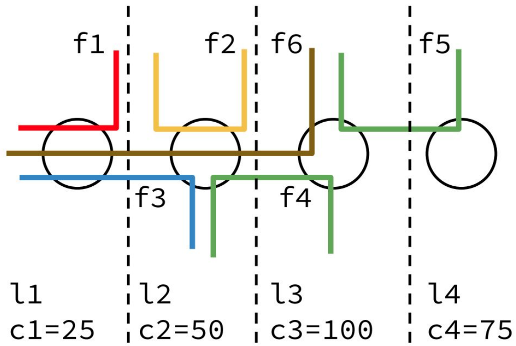
Can we use GradientGraph for Network Baselining?



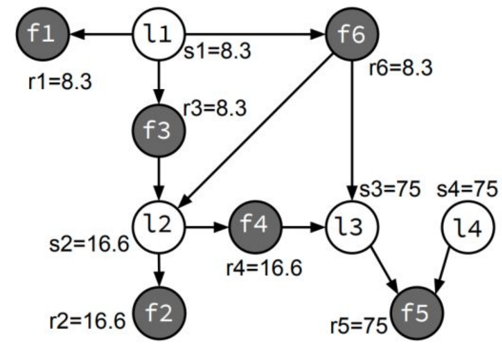
Flow Gradient Graph:



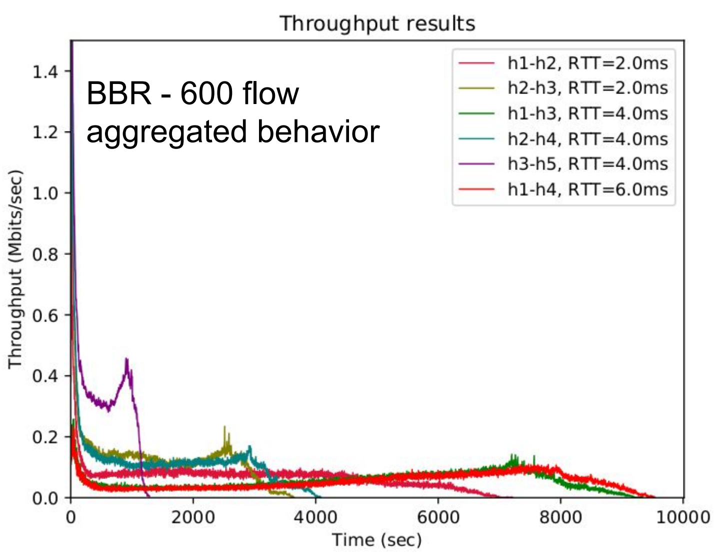
Can we use GradientGraph for Network Baselining?

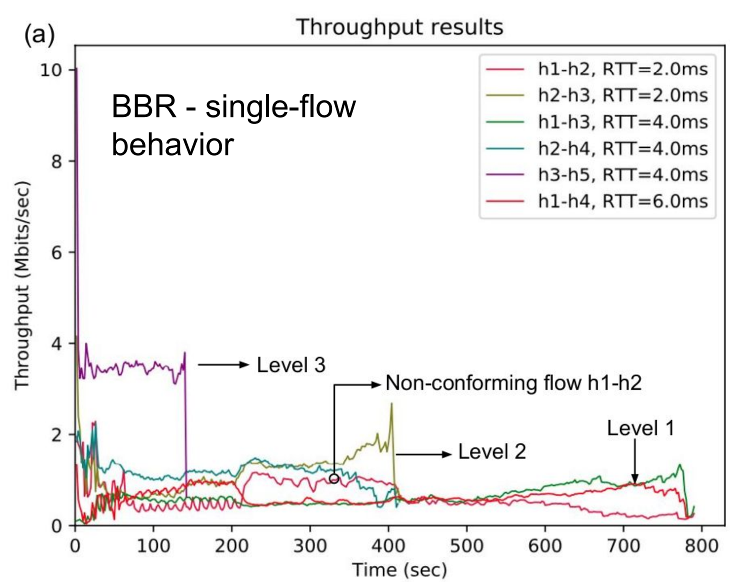


Flow Gradient Graph:

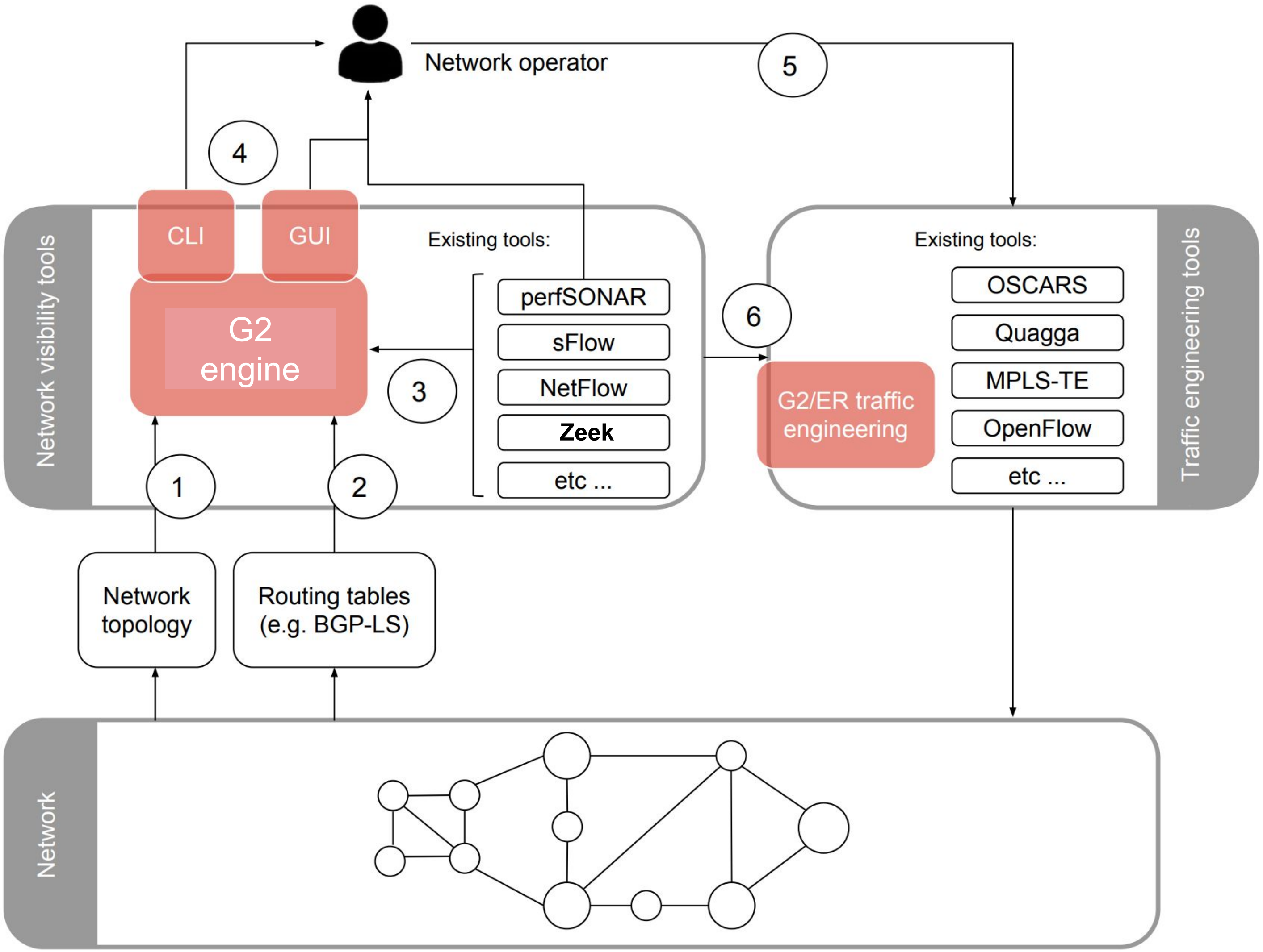


$r_{th}^* = 0.83$ Mbps (theoretical); $r_{peak} = 1.41$ Mbps (real peak); $r_{avg} = 1.17$ Mbps (real average);

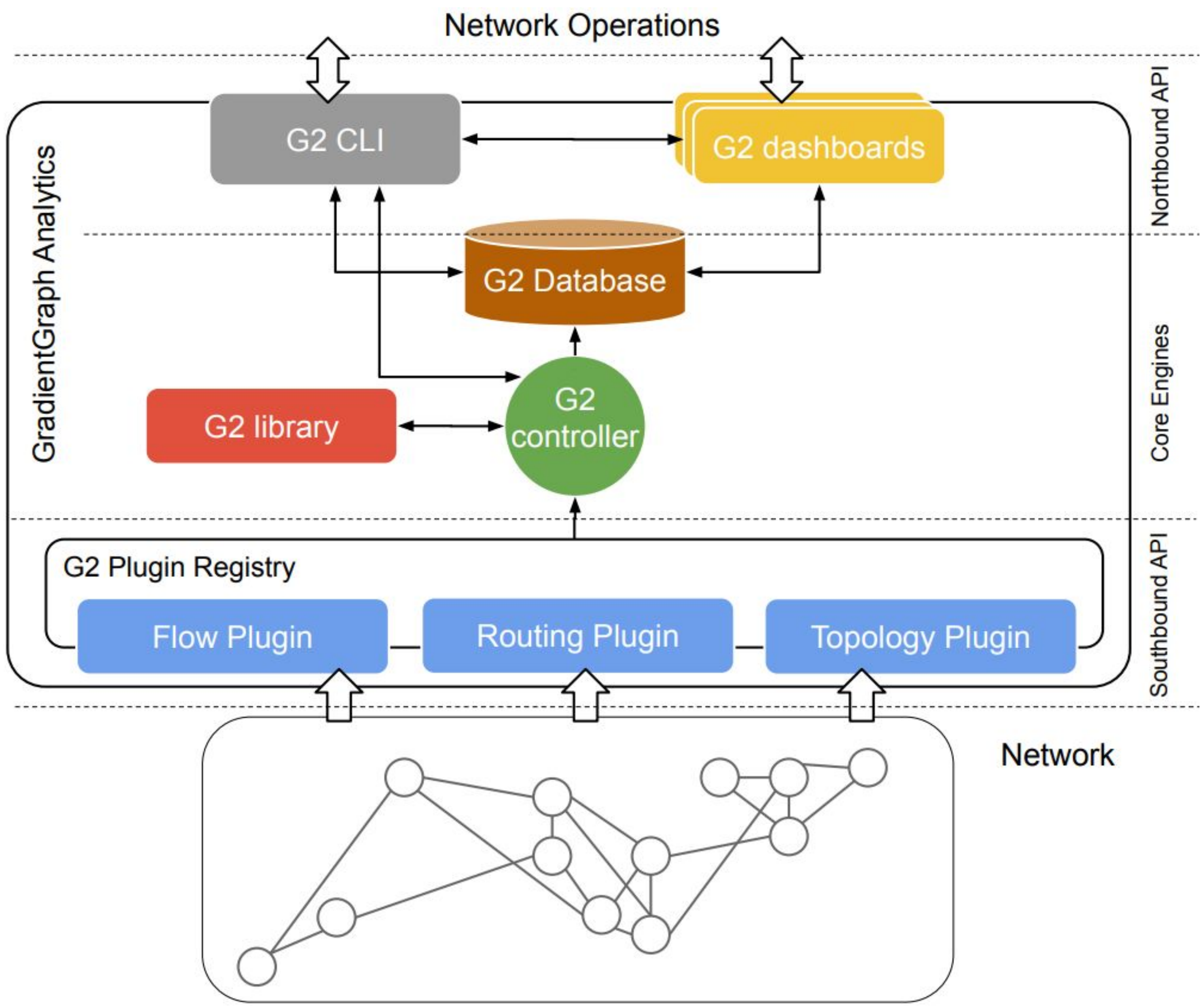




GradientGraph Analytics: Operational Workflow



GradientGraph Analytics: Architecture



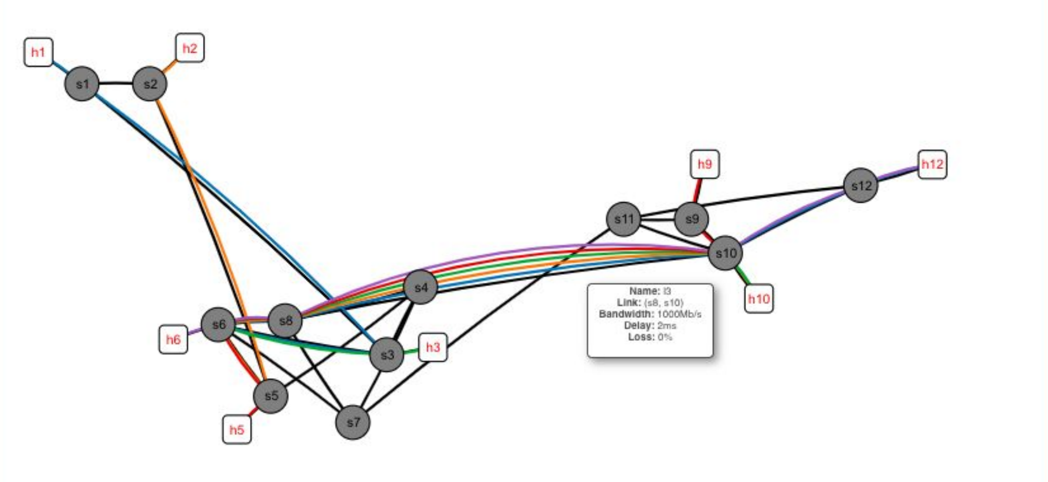
GradientGraph Analytics Platform

Reservoir Labs, Inc.

Network Topology Dashboard

Dashboard ▾

Topology Panel



Toggle Filter | Toggle Flows | Reload Flows | Reload All

Switches

Switches | Hosts | Links | Flows

Type in element to search

Show 10 entries

ID	Links Attached
s1	(s1 , s3) (s1 , s2)
s10	(s9 , s10) (s8 , s10) (s10 , s12) (s10 , s11)
s11	(s9 , s11) (s10 , s11) (s7 , s11) (s11 , s12)
s12	(s10 , s12) (s11 , s12)
s2	(s2 , s5) (s1 , s2)
s3	(s3 , s6) (s3 , s4) (s1 , s3)
s4	(s3 , s4) (s4 , s8) (s4 , s7) (s4 , s5)
s5	(s2 , s5) (s5 , s6) (s4 , s5)
s6	(s3 , s6) (s6 , s8) (s6 , s7) (s5 , s6)
s7	(s7 , s11) (s7 , s8) (s6 , s7) (s4 , s7)

Showing 1 to 10 of 12 entries

Previous | 1 | 2 | Next

Dashboard Configuration

Automated Refresh

Off

Refresh Rate (s)

10

Time to refresh (positive integer)

Overlap Flows

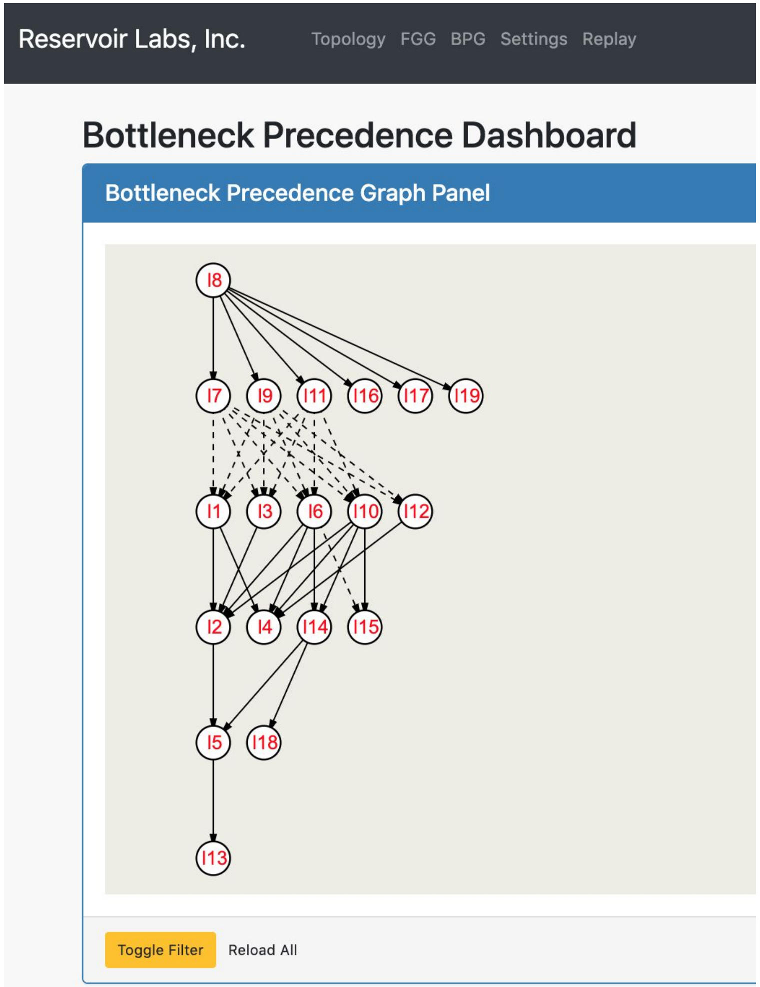
No

☐ **Export Configuration**

Apply

Automated Refresh Stops

GradientGraph Analytics Platform



Filter: l1 **# Neighbors:** 2 APPLY FILTER OFF FILTER

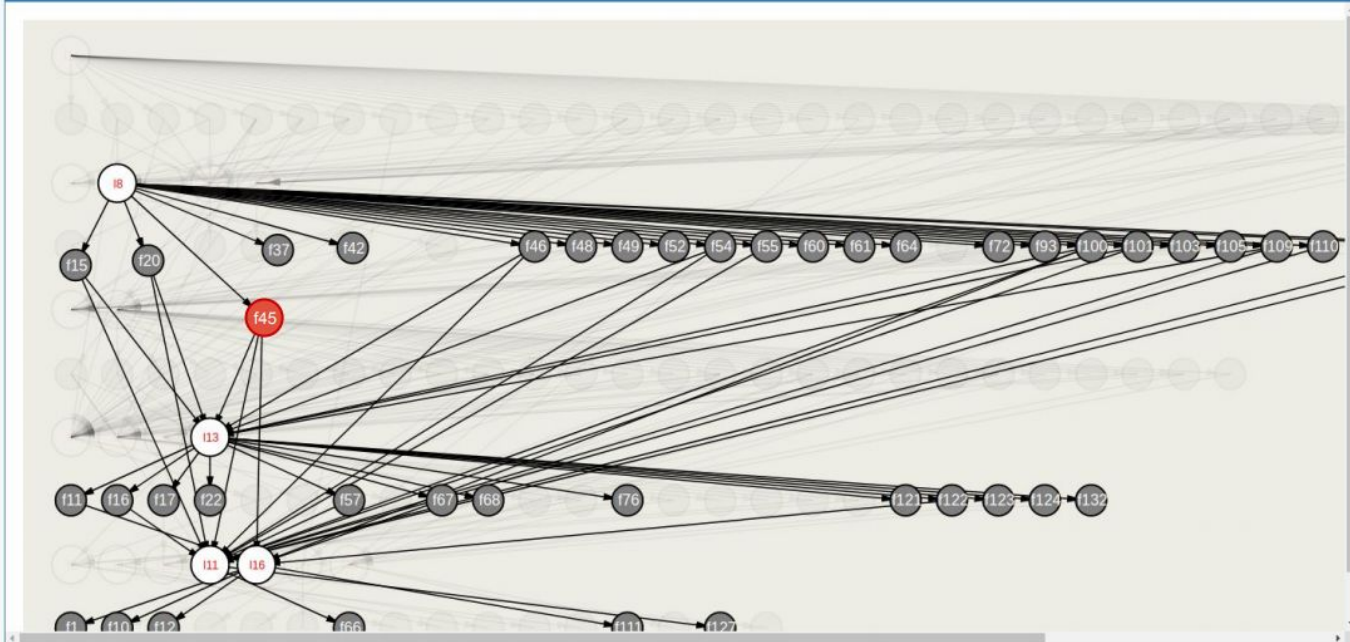
ID.	Nodes Attached	Bandwidth (Mb/s)	Delay (ms)	Loss (%)	Fairshare (Mb/s)	In Nodes	Out Nodes
l1	(s1 , s3)		2ms	0	31.54	l7(i), l9(i), l11(i)	l4, l2
l2	(s1 , s2)		2ms	0	9.57	l1, l3, l6, l10	l5
l3	(s6 , s8)		10ms	1	37.08	l7(i), l9(i), l11(i)	l2
l4	(s4 , s8)		2ms	0	8.89	l1, l6, l10, l12	None
l5	(s2 , s5)		2ms	0	3.35	l2, l14	l13
l6	(s4 , s5)		2ms	0	7.33	l7(i), l9(i), l11(i)	l14, l4, l2, l15(i)
l7	(s5 , s6)		2ms	0	4.50	l8	l6(i), l10(i), l3(i), l12(i), l1(i)
l8	(s3 , s6)		2ms	0	2.63	None	l19, l16, l17, l11, l7, l9

GradientGraph Analytics Platform

Reservoir Labs, Inc. Topology **FGG** BPG Settings Replay

Flow Gradient Graph Dashboard

Flow Gradient Graph Panel



Toggle Filter Reload All

Filter: f45 **# Neighbors:** 2 **Direction:** DOWNST ▾ APPLY FILTER OFF FILTER FLOWS ▾

ID.	Start Host	End Host	Flow Size (bytes)	Gradient ↓	Transmission Rate (Mb/s)	Fairshare (Mb/s)	In Nodes	Out Nodes
f45	h5	h1	256000000	1.123	3.21	3.21	l8	l16, l11, l13
f111	h11	h1	256000000	0.667	18.97	18.97	l16, l19	None
f42	h4	h10	256000000	0.307	3.21	3.21	l8	l3, l1

Use Cases: Towards Intelligent Networks

3.3 Intelligent Networks

In many senses, the CERN network has been "software defined" for many years given the extent to which its management and operation would be impossible without the extensive suite of tools developed to manage and control the hundreds of routers and thousands of switches deployed across the site. That being said, Software Defined Networking and Network Function Virtualisation technologies being discussed in the industry could be paired with new routing technologies and network status information to implement so-called Intelligent Networks, i.e. networks that are able to adapt themselves in real time according to their status and utilization. An interesting use case—and one that has already been successfully demonstrated for data transfers between CERN and Nikhef—is to adjust network routing so that backup paths can temporarily be used to increase the available bandwidth for high-volume data transfers.

[*] "CERN External Network evolution for LHC Run3 and Run4"",: Edoardo Martelli, Tony Cass, CERN IT-CS CERN, 28th of February 2019

Use Cases: Optimized Routing in NOTED

NOTED activity

Exploring options to select outgoing network path from a site to load balance traffic across links to

- smooth peaks
- increase usable bandwidth

Principles

Shared knowledge:

- **Data transfers repository**: centralized repository of upcoming and ongoing major(*) data transfers
- **Network status repository**: centralized repository with information of congested interconnecting links

Act local:

- Network Providers can use such info to more efficiently use their own networks

[*] "NOTED activity", LHCONE meeting, Edoardo Martelli, 31st of October 2018

Use Cases: Heterogeneous Networks

DUNEONE proposal

It is proposed to build a VPN similar to LHCONE to connect protoDUNE and DUNE sites that are already connected to LHCONE, to allow those sites to prototype and test technical solution to correctly separate the traffic between the two VPNs. A two-phase project is proposed:

Phase 1: Migration of ongoing data transfers of the pre-processed data generated by the CERN-based protoDUNE detector(s) to FNAL (DUNE T0) archive facilities. This data movement is currently being carried over the LHCOPN.

Phase 2: Selective migration of Rucio-based data movement for DUNE's emerging distributed data storage facilities. This would be implemented on a site-by-site basis, as individual DUNE sites elect to participate in the project. Phase 2 would commence after satisfactory demonstration of proof-of-concept in the Phase 1 testing & evaluation, and consultation with the DUNE collaboration.

The tests will be structured in a manner to not disrupt production traffic. It also should be emphasized that **this project is targeted at proof-of-concept, not establishing a DUNE-wide service**.

CERN | IT Information Technology Department 10

[*] MultiONE presentation at LHCOPN/LHCONE Workshop Jan 2020

Use Cases: Heterogeneous Networks

LHCONE, DUNEONE, SKAONE, ALLINONE

Threats and Opportunities from many (scientific) communities competing for networks (say SKA, LSST, DUNE, Belle-2, 3rd Generation G-Waves, CTA, …)

Threats: competing for bandwidth (really a funding issue) and increasing complexity (security, “QualityOfService”, ..). We have very good experience with LHCONE, but how does it extend to those other communities? Invitation to the network community to define the problem scope and look for solutions. Notice this is not a point-2-point problem but a global problem and the solution needs to be simple enough

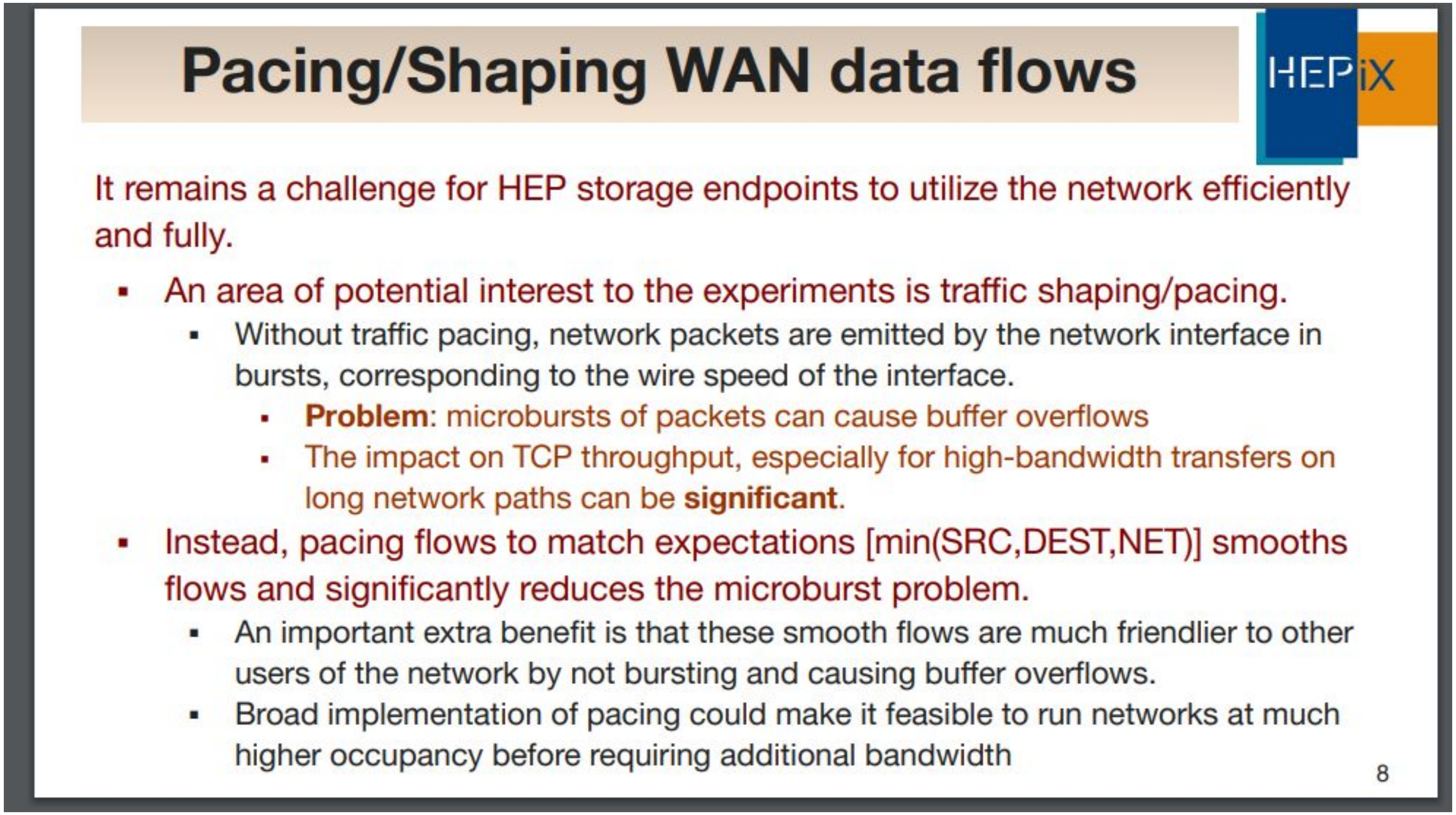
Opportunities: increase the global worldwide connectivity particularly regions with a complicated connectivity (Asia is an example – synergy with ATCF). Expand the LHCOPN/LHCONE experience and ecosystem to new scientific communities (operations, policies, …). Global scientific network operations ?

Carefully craft a solution simple for everyone: experiments, NRENs, sites

CERN Simone.Campana@cern.ch - LHCOPN/LHCONE meeting 14/01/2020 14

[*] The DOMA project, Simona Campana, LHCOPN/LHCONE Workshop Jan 2020

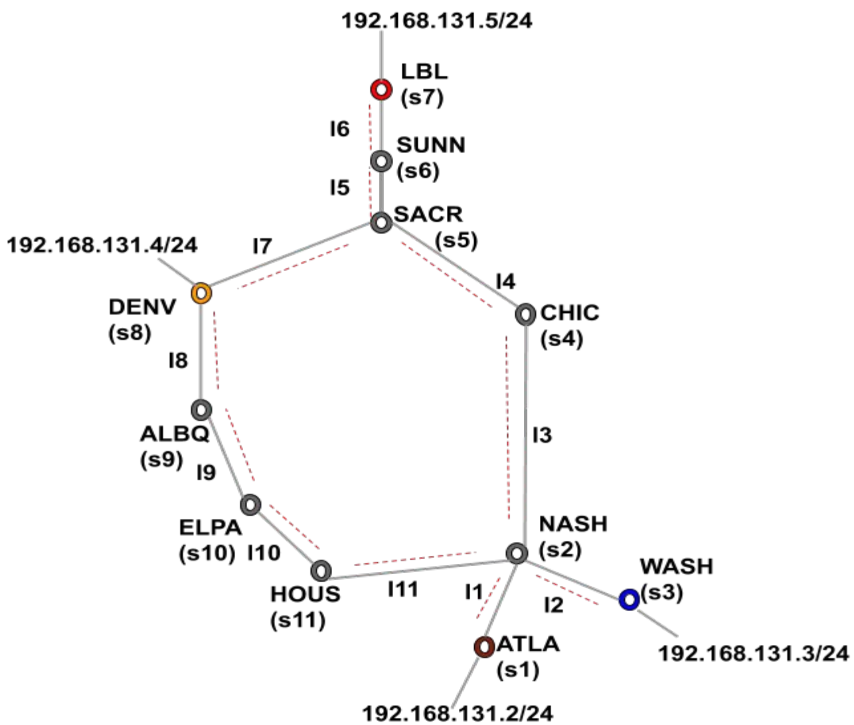
Use Cases: Optimal Traffic Shaping



[*] HEPiX NFV, Shawn McKee, LHCOPN/LHCONE Workshop Jan 2020

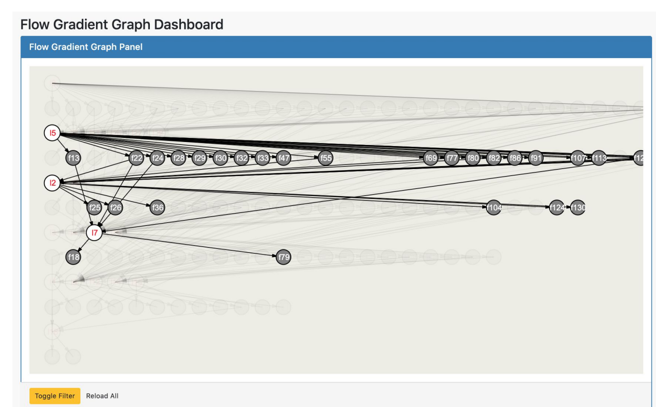
Thank you!

Reach out to us for a demo of GradientGraph Analytics
giralt@reservoir.com, info@reservoir.com



Thank you to DOE/ESnet 100Gbps Testbed Network

ESnet
100Gbps Testbed Network

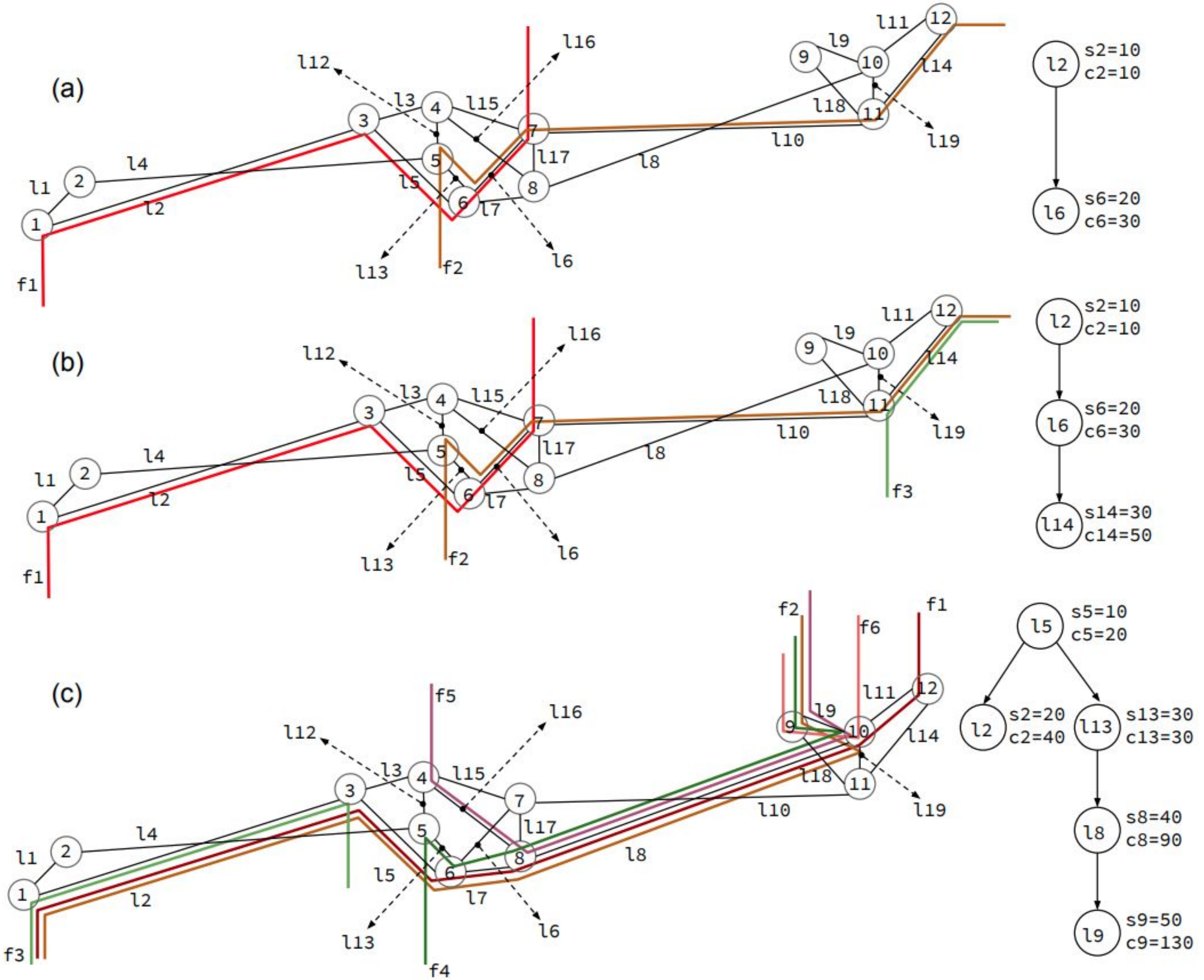


Backup slides

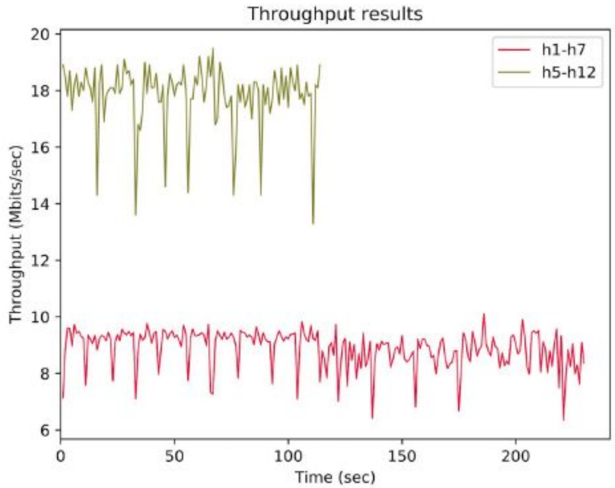
Towards Full Network Visibility and Understanding

- Assume a network N consisting of a set of flows F and a set of links L.
- Assume flows control their transmission rate using TCP.
- We'd like to answer:
  - What are the bottleneck links? bottlenecks
  - What is each flow's bottleneck link? bottlenecks
  - What is the transmission rate of each flow? flows
  - If a flow's minimum/maximum rate constraint is increased or decreased by an amount δ: flows
    - How is the rate of the rest of the flows affected? flows
    - How is the bottleneck structure of the network affected? bottlenecks
  - If a link's capacity is decreased or increased by an amount δ: bottlenecks
    - What is the answer to the previous two questions? flows

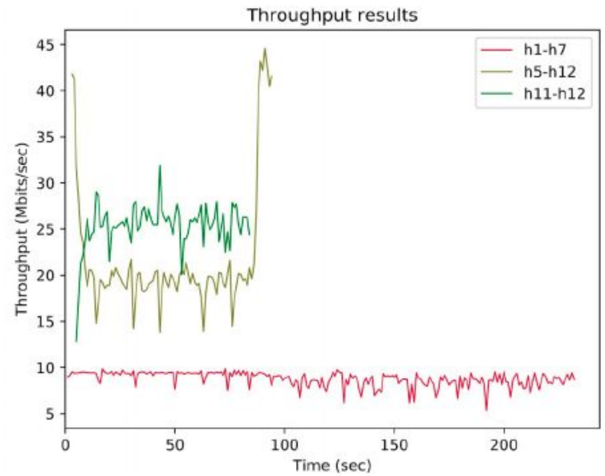
Empirical Evidence of Bottleneck Structures in TCP Networks



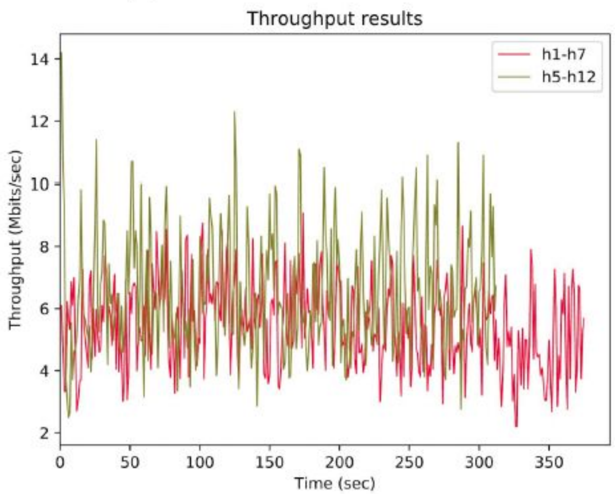
Empirical Evidence of Bottleneck Structures in TCP Networks



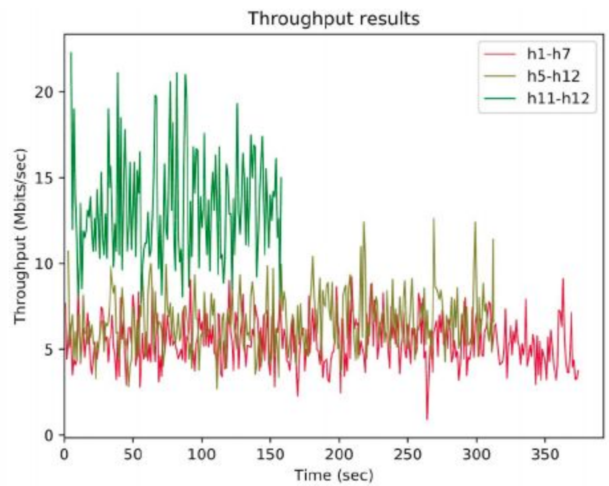
(a) 2-level BPG with BBR.



(b) 3-level BPG with BBR.



(d) 2-level BPG with Cubic.



(e) 3-level BPG with Cubic.

Empirical Evidence on the Depth of the Bottleneck Structure and TCP Convergence Time

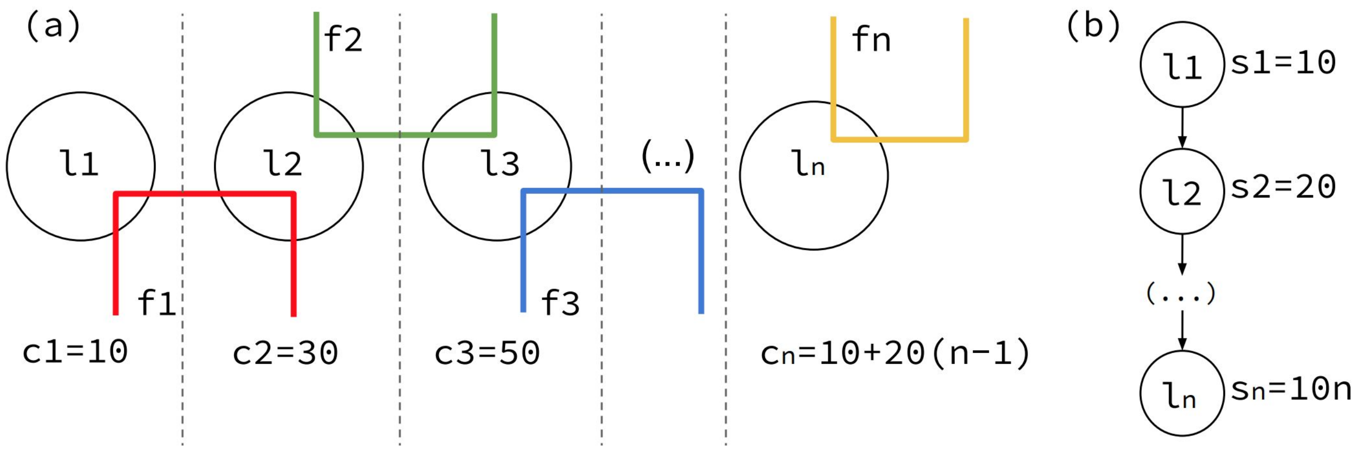


Table 4: Converge time increases with the number of levels and the number of level-competing flows.

	1-Level	2-Level	3-Level	4-Level
num. flows x 1	2	2	2	2
num. flows x 2	2	2	12	26
num. flows x 3	4	16	14	54
num. flows x 4	14	26	34	72

GradientGraph Operational Modes

- Three **operational levels** based on **time granularity**:
  - **Real time** feedback loop traffic engineering (millisecs, secs)
  - **Operator-in-the-middle** traffic engineering (hours, days)
  - Network **design, planning and upgrades** (weeks, years)