

Autonomous Traffic Engineering with Self-Configuring Link Weights

Srikanth Sundaresan[†], Cristian Lumezanu[†],
Pierre Francois[‡], Nick Feamster[†]

[†]Georgia Institute of Technology

[‡]Université catholique de Louvain

Motivation

- Intradomain TE balances load across a network
- Load balancing improves
 - Latency
 - Resilience to traffic surges
 - Available capacity of network
- Typical objective function:

Minimize maximum utilization over all links

Outline

- **What makes a good Traffic Engineering Protocol?**
- The key idea behind SculpTE
- Stability and responsiveness
- Evaluating SculpTE
- Implementation issues

What makes a good TE protocol?

- Responsiveness
 - Has to react to fluctuations in a timely manner
- Stability
 - Reaction to fluctuations should not induce instability
- Performance
 - Has to perform under variety of conditions
- No prior assumptions
 - About traffic patterns, failures, network state

State of the Art

Scheme	Responsiveness	Stability	Performance	Config-free
IGP-WO	x	✓	?	x
MATE	✓	✓	✓	x

[Fortz2000, Fortz2002, Fortz2003, Elwalid2001]

Why Configuration-free?

- Flexibility
 - Zero knowledge of traffic patterns, or network
- Risk reduction
 - No risk of poor configuration

SculpTE

- We propose a completely self-configuring, adaptive TE solution
 - No initial configuration required
 - Dynamic link weight updates in response to traffic patterns
 - Stable, responsive, robust to failures

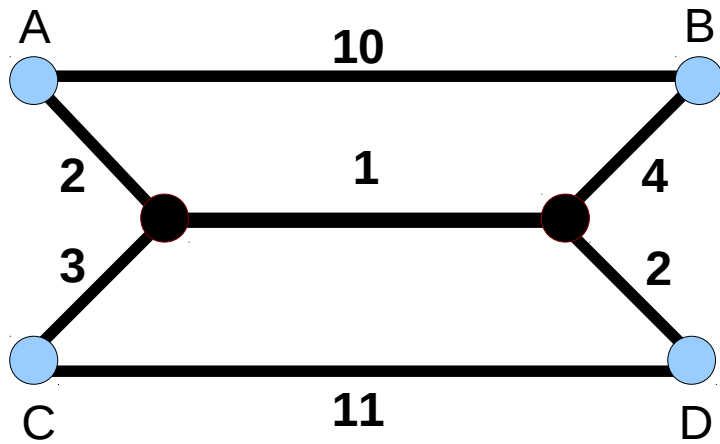
Outline

- What makes a good Traffic Engineering Protocol?
- **The key idea behind SculpTE**
- Stability and responsiveness
- Evaluating SculpTE
- Implementation issues

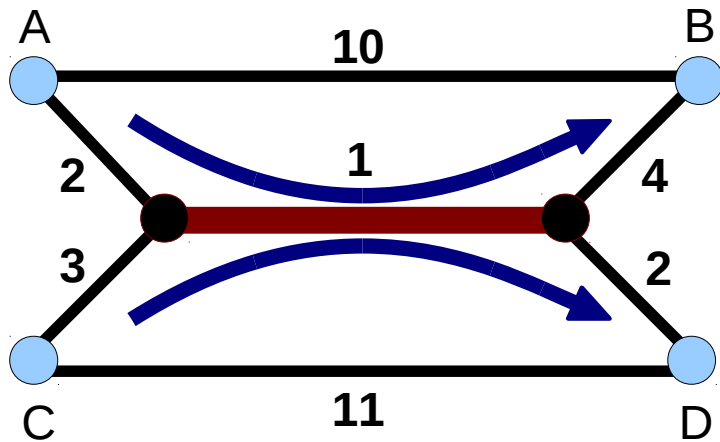
Using link weights for traffic control

- How can traffic be diverted from congested links in a controlled way using link weights?
 - Increasing the link weight of the congested link
- How much to increase it by?
 - Too little, and nothing happens
 - Too much, and a lot of traffic is shifted away
- Use the ***key metric*** of the congested link

How the key metric works

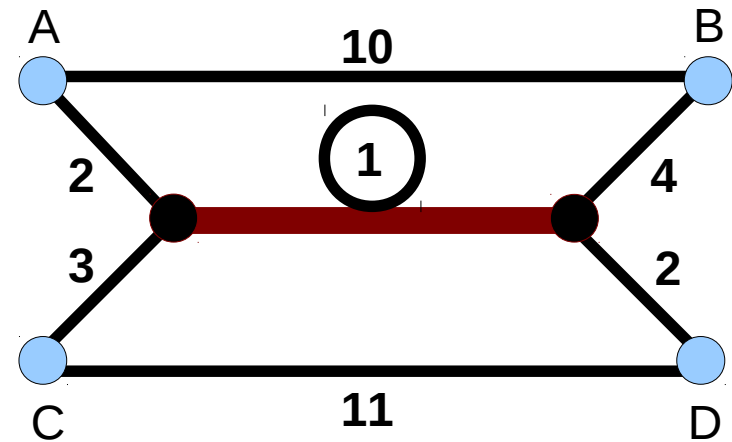
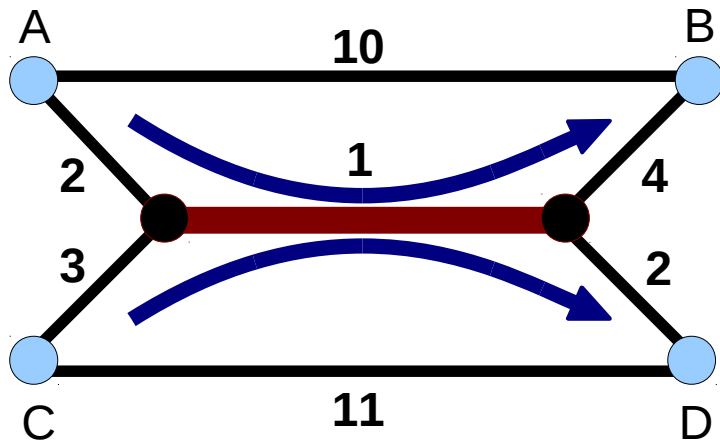


How the key metric works



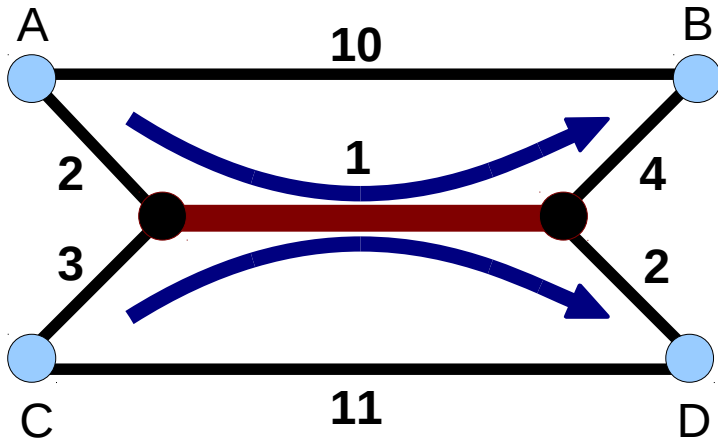
Both AB and CD send traffic through the red link.

How the key metric works

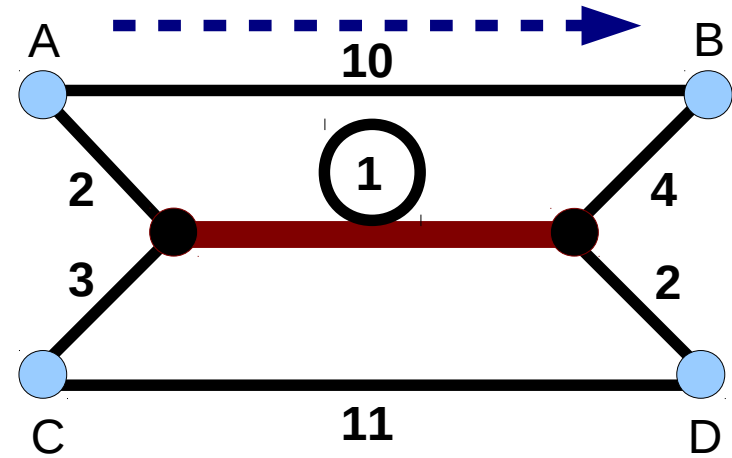


Both AB and CD send traffic through the red link.

How the key metric works

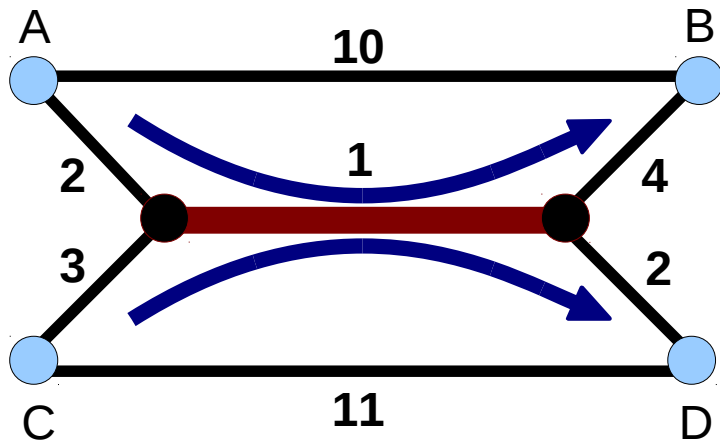


Both AB and CD send traffic through the red link.

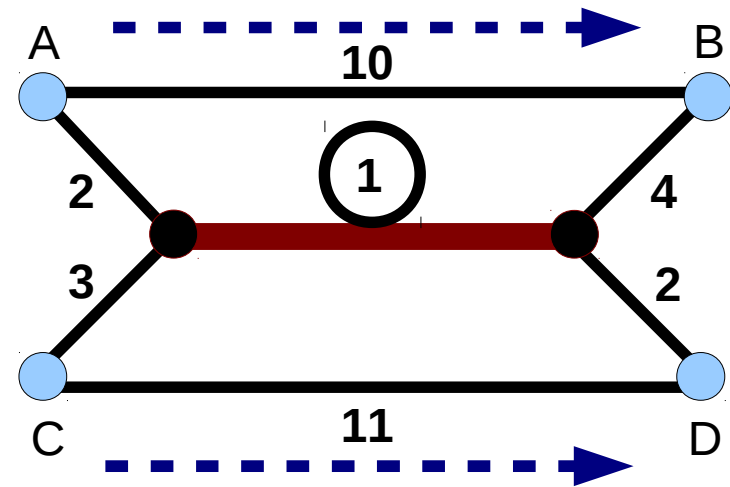


$AB's\ key\ metric = 10 - 7 = 3$

How the key metric works



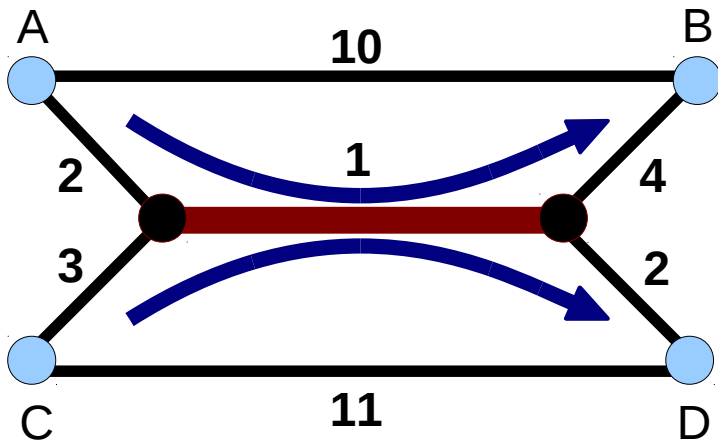
Both AB and CD send traffic through the red link.



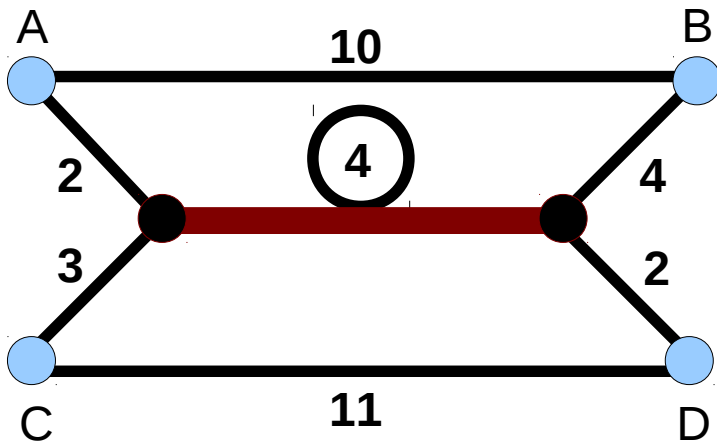
$$\text{AB's key metric} = 10 - 7 = 3$$

$$\text{CD's key metric} = 11 - 6 = 5$$

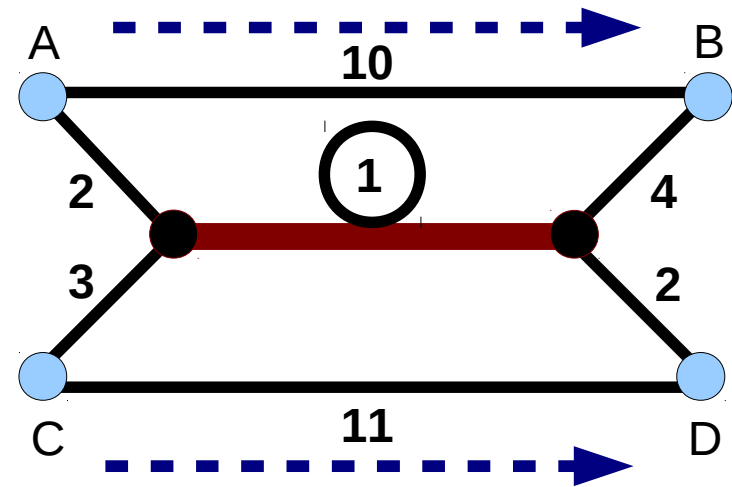
How the key metric works



Both AB and CD send traffic through the red link.



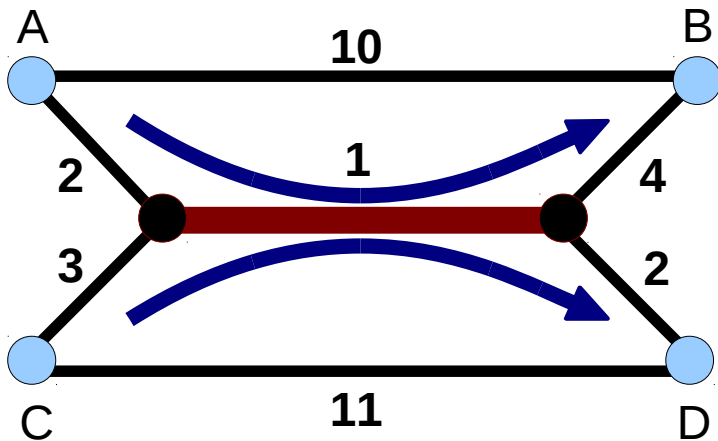
Chosen key metric = 3



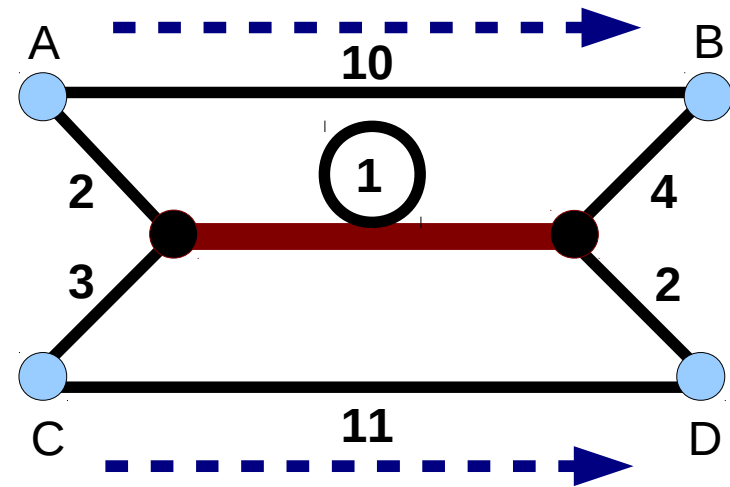
$$\text{AB's key metric} = 10 - 7 = 3$$

$$\text{CD's key metric} = 11 - 6 = 5$$

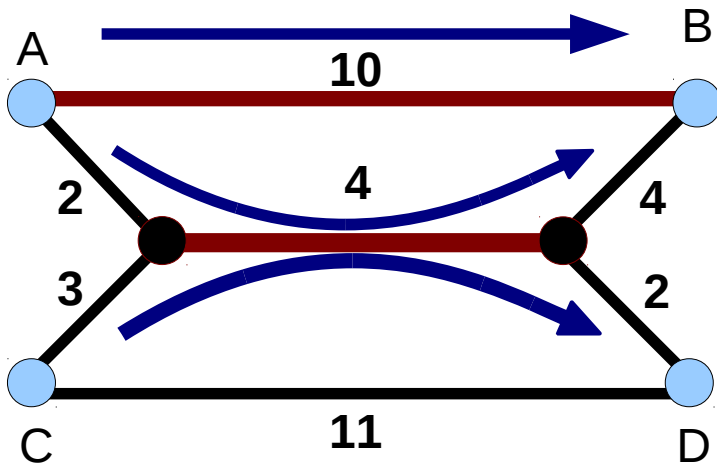
How the key metric works



Both AB and CD send traffic through the red link.

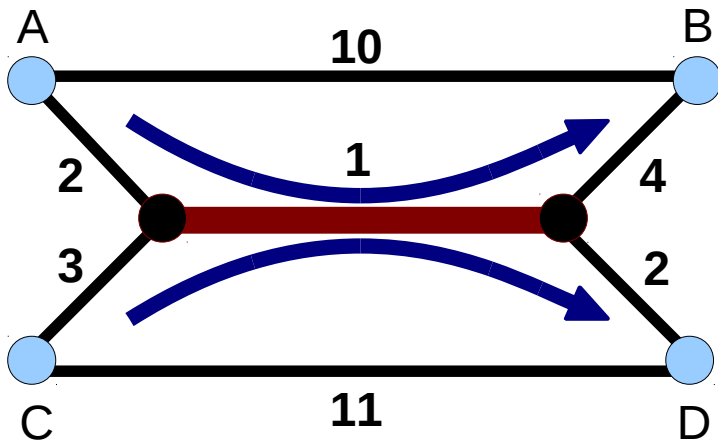


AB's key metric = $10 - 7 = 3$
 CD's key metric = $11 - 6 = 5$

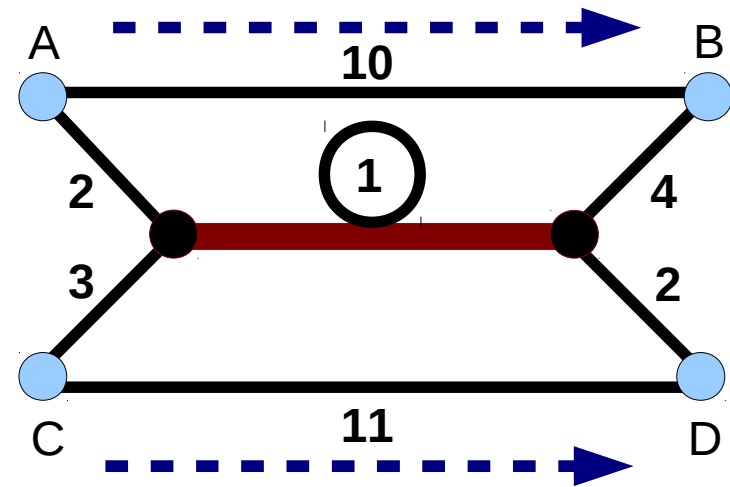


If chosen key metric = 3
 AB's traffic gets split.
 CD's traffic *does not*.

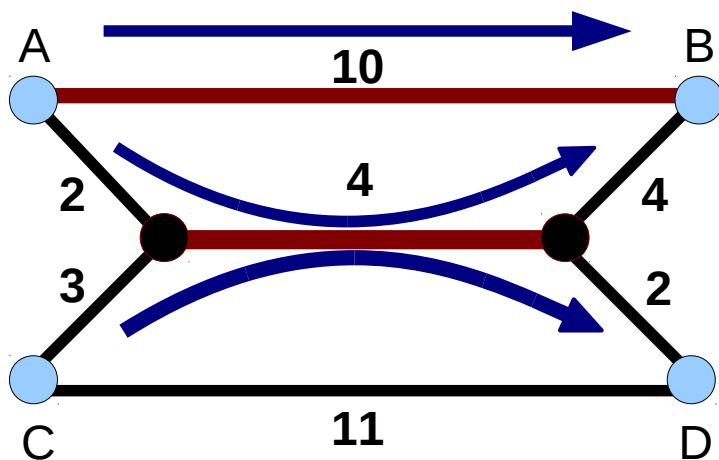
How the key metric works



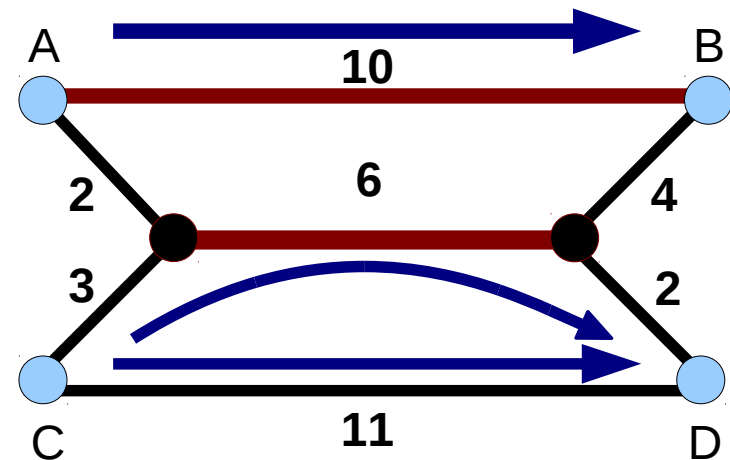
Both AB and CD send traffic through the red link.



AB's key metric = $10 - 7 = 3$
 CD's key metric = $11 - 6 = 5$



If chosen key metric = 3
 AB's traffic gets split.
 CD's traffic *does not*.



If chosen key metric = 5
 AB's traffic gets diverted fully.
 CD's traffic gets split

Outline

- What makes a good Traffic Engineering Protocol?
- The key idea behind SculpTE
- **Stability and responsiveness**
- Evaluating SculpTE
- Implementation issues

Choosing the key metric for stability

- A congested link has potentially many IE pair paths passing through it
 - Each path has its own key metric value k
- Applying k will deflect traffic between all IE pairs that have key metric $\leq k$
- SculpTE chooses the lowest key metric
 - Minimum quantum of traffic shifted away from congested link

SculpTE avoids instability scenarios

- What if traffic is diverted from the most congested link to second most congested link?
 - Ripe situation for oscillation
- Do not divert traffic towards links with low *residual capacity*
 - Forces search for low-utilization paths with higher residual capacities
 - Accounts for skewed link capacities

Improving Stability

- Multiple layer-3 topologies over a single physical topology
 - Each topology has own set of link weights
 - Incoming flows hashed uniformly to one topology
 - IE flows divided
 - Only a single topology updated at a time
 - Reduces amount of traffic diverted

SculpTE responds to network state

- Link weight updates applied periodically
- Responds to traffic changes at granularity of periodicity of updates
- Guaranteed to shift traffic away from the most congested link
 - As long as there is an alternate path that avoids that link

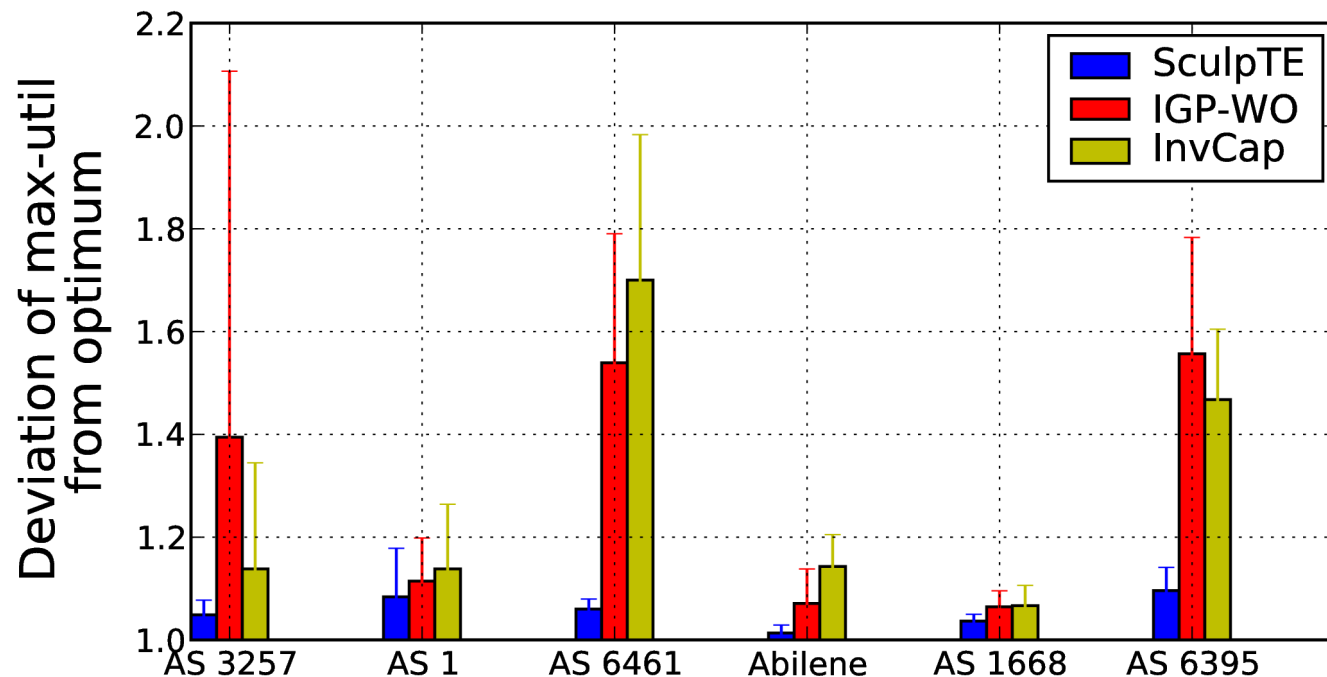
Outline

- What makes a good Traffic Engineering Protocol?
- The key idea behind SculpTE
- Stability and responsiveness
- **Evaluating SculpTE**
- Implementation issues

Evaluation method

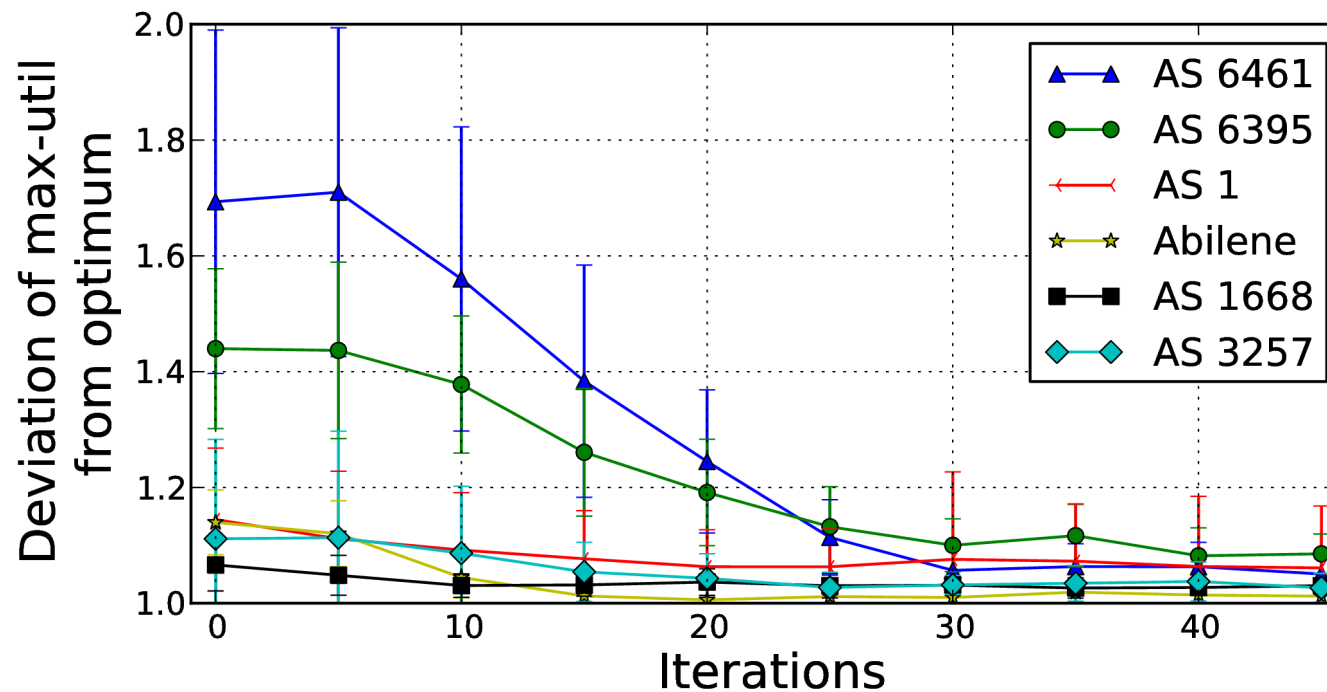
- Flow level simulator used
- Evaluation over multiple AS topologies
- Variety of traffic demands, network failure events
- Comparison to IGP-WO (various flavors), InvCap
- Baseline – optimum solution obtained by solving LP formulation

SculpTE outperforms offline TE



- Comparison to InvCap, and IGP-WO
- Performs well over a variety of ASes. (within 15% of optimum)

SculpTE converges quickly

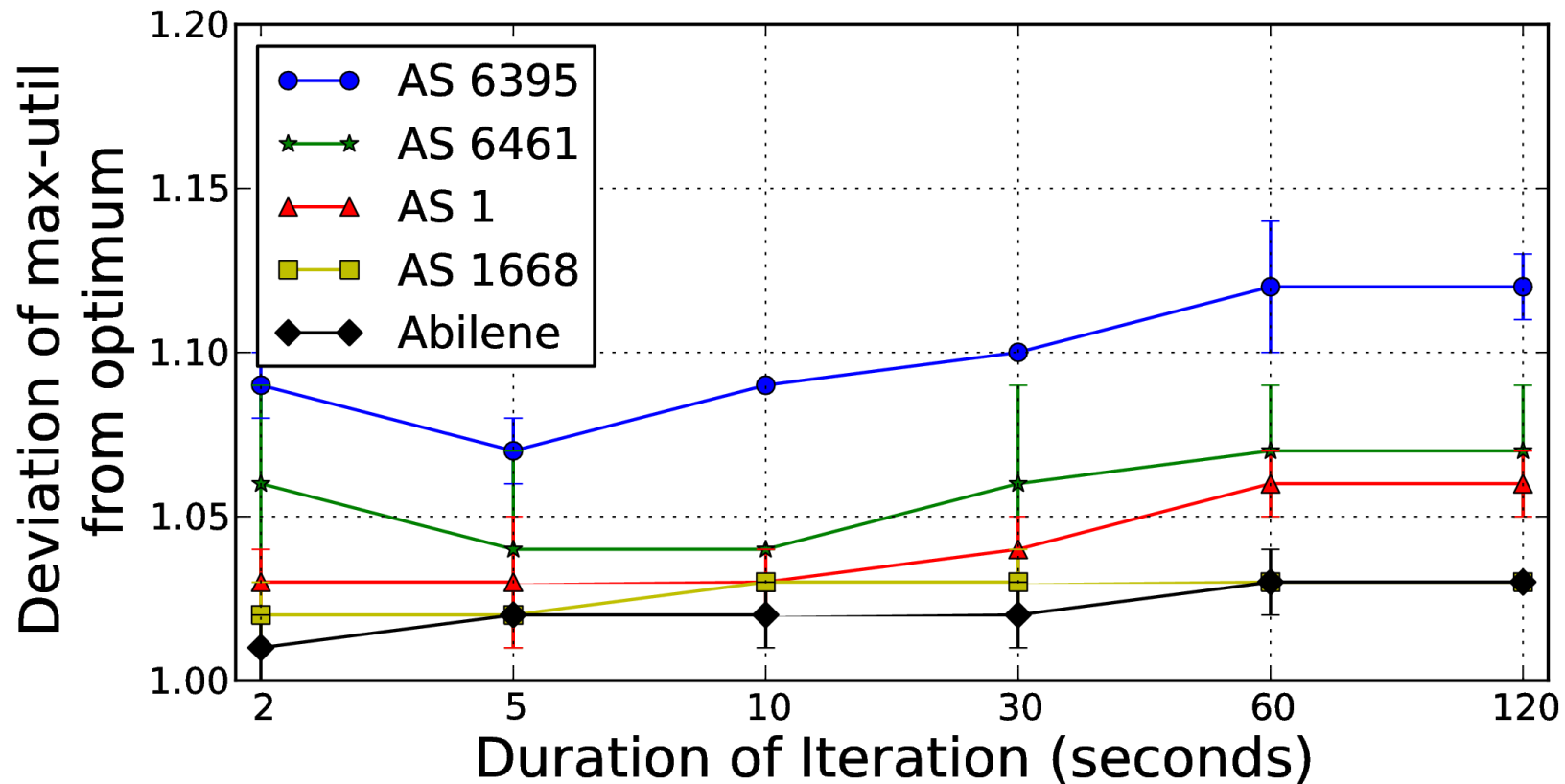


- Convergence within ~25 iterations
 - Even when initial setting is sub-optimal

SculpTE handles stress conditions

- Traffic Surges:
 - Random spikes and troughs induces in demand
 - SculpTE adapts to traffic demand
 - IGP-WO performance sub-optimal
 - Also hard to predict nature of surges
- Link failures
 - Upto 3 link failures tested
 - SculpTE performs well under multiple link failures
 - IGP-WO (optimized for specific link failures) performs poorly under no-failure (normal) case.

SculpTE is Config-free



- Agnostic to initial settings
 - Stable across wide variety of settings
- Achieves goal of *self-configuring*

How does SculpTE compare to the state of the art?

Scheme	Responsiveness	Stability	Performance	Config-free
IGP-WO	x	✓	?	x
MATE	✓	✓	✓	x
SculpTE	✓	✓	✓	✓

Outline

- What makes a good Traffic Engineering Protocol?
- The key idea behind SculpTE
- Stability and responsiveness
- Evaluating SculpTE
- **Implementation issues**

Implementing SculpTE

- Distributed implementation with existing routing protocols
 - Routers already support multipath
 - Bits in IP header to choose topology
- Centralized implementation
 - Centralized implementation with OpenFlow switches and NOX controllers
 - Successful implementation of small 4 node network

Implementation challenges

- IGP convergence and router synchronization
 - Performs well with iteration duration $> 100s$
 - Can be configured to be threshold-triggered
- Flow rerouting
 - Pin flows to routes
- Router overhead
 - Needs to support multiple FIBs, frequent link weight updates
- Fast computation of key metric

Deployment Scenarios

- Datacenters
 - Short lived bursts and long term Map-reduce-style traffic surges
- Threshold -triggered updates
 - Activated only if max-utilization exceeds threshold
 - Minimizes overhead during normal operation

Conclusion

- Current TE proposals require specialized configuration – easy to go wrong!
- SculpTE is the first TE stable proposal that is essentially configuration-free
 - SculpTE modifies link weights online
- Excellent performance across wide variety of ASes, traffic scenarios, network states

Contact: srikanth@gatech.edu

References

- [Fortz2000] B. Fortz and M. Thorup. Internet traffic engineering by optimizing OSPF weights. In Proc. IEEE INFOCOM, March 2000
- [Fortz2002] B. Fortz and M. Thorup. Optimizing OSPF/IS-IS weights in a changing world. IEEE JSAC, May 2002
- [Fortz2003] B. Fortz and M. Thorup. Robust optimization of ospf/is-is weights. In Proc. International Network Optimization Conference, 2003.
- [Kandula2005] S. Kandula, D. Katabi, B. Davie, and A. Charny. Walking the tightrope: Responsive yet stable traffic engineering. In Proc. ACM SIGCOMM, 2005.