Achieving Near-Optimal Traffic Engineering in Current OSPF/ISIS Networks

Ashwin Sridharan, Roch Guérin and Christophe Diot

University of Pennsylvania

Sprint Labs

NANOG 27

Phoenix, Arizona

Initial Motivation

- Traffic engineering promises greater network efficiency (lower cost!)
  - Better performance/lower capacity through knowledge of traffic matrix
- What can we "really" get?
  - What is achievable through "tweaking" existing techniques ?
  - How much configuration overhead is involved?
  - Do we need new technology ?

Why Bother?

- Leverage large installed OSPF/ISIS base.
  - Has been in use for a reasonably long period.
  - A lot of experience in managing such networks.
- Smoother transition process.
  - Known technology base.
  - Incremental versus fork-lift upgrade.
- Potential for lower cost solution.
  - No need for new equipment?

- But...... What about performance ?
  - Routing only along shortest paths.
  - Must split traffic equally over equal cost paths.

What Do We Know?

- Local search heuristic of Fortz and Thorup [1].
  - Knowledge of traffic and picking the “right” link weights can improve performance.
  - Traffic distribution can still be far from optimal for some topologies and traffic matrices.
  - Equal splitting across equal cost paths is a problem.

- Can match optimal - Wang, Wang & Zhang [2].
  - Forwarding along “shortest paths” not a critical limitation.
  - Achievable by formulation of a linear multi-commodity flow problem that yields shortest paths.

What is missing ?

The Linear Program formulation requires *unequal* splitting over equal cost shortest paths

- This violates the forwarding paradigm of current OSPF/ISIS networks
  - Doable but not straightforward
  - Requires data path changes

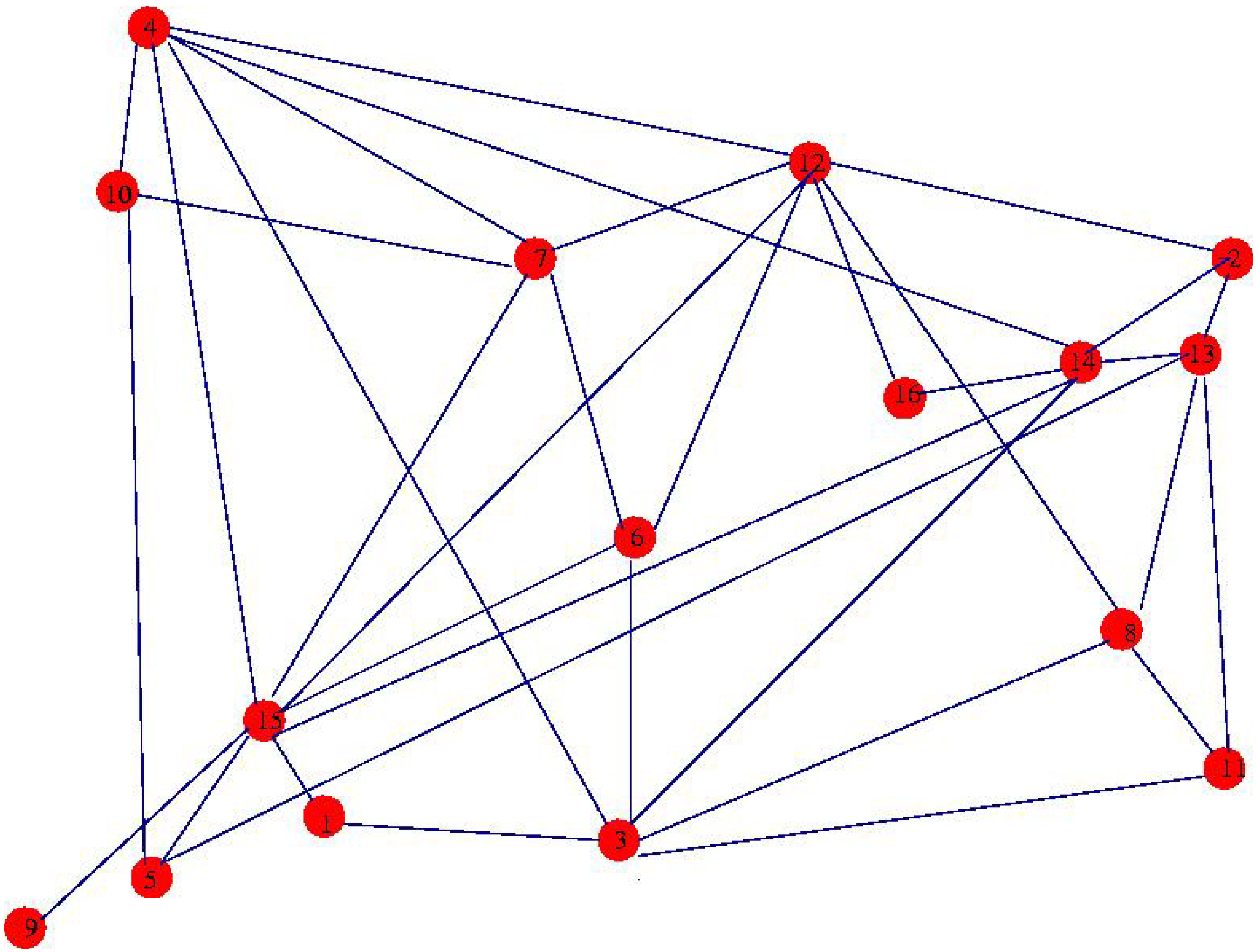
Our Approach to Overcome The Splitting Problem

- Current routing tables have thousands of prefixes
- Instead of routing each prefix on all equal cost paths, selectively assign next hops to (each) prefix
  - In other words, *remove* some equal cost next hops assigned to prefixes
- Goal is to approximate the optimal link load

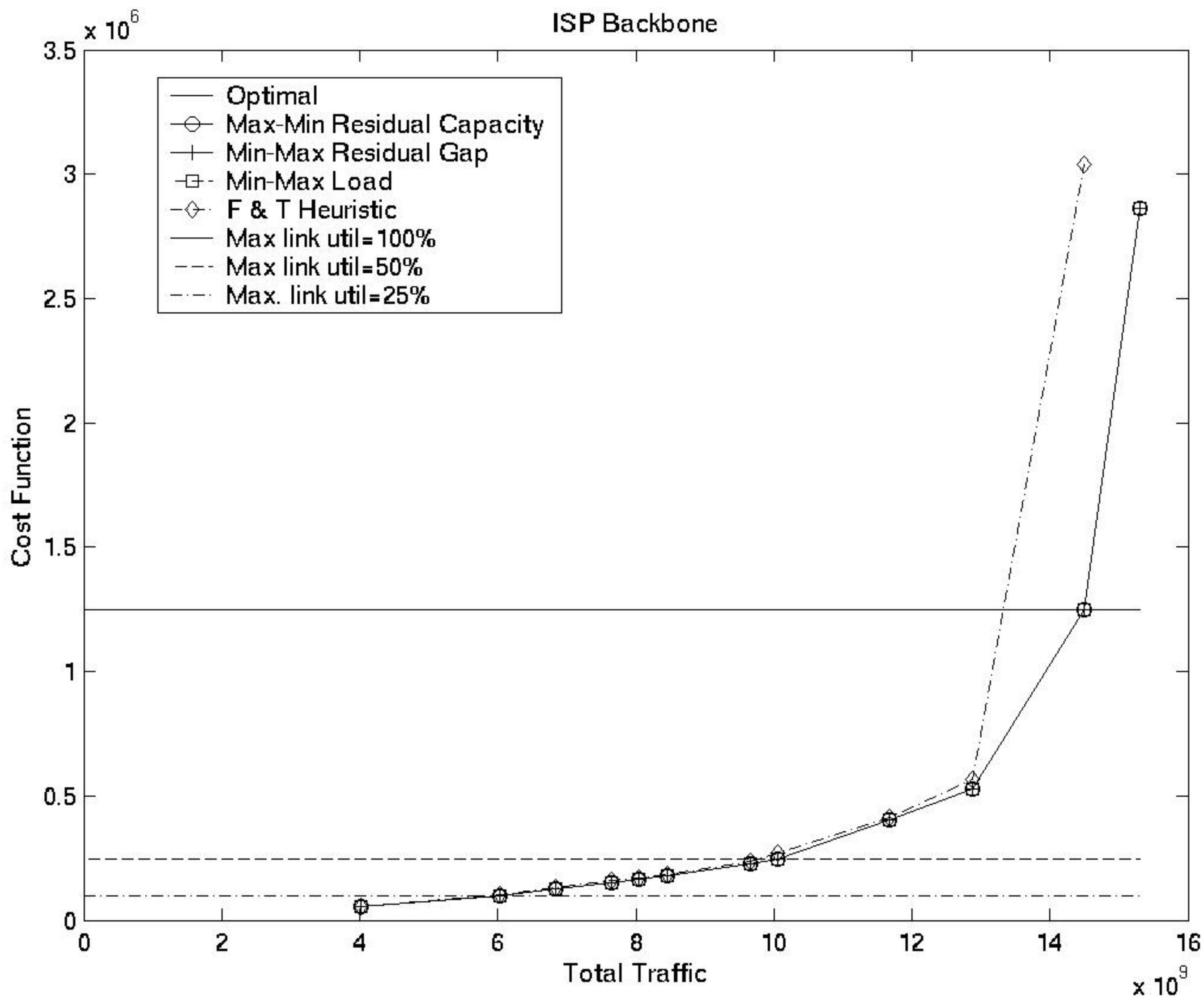
Solution & Benefits

- Problem is NP-Hard
  - We have 3 good heuristics, one with a performance gap bounded logarithmically from the optimum
- There is a large degree of flexibility in next hop selection to match optimal allocation
  - Current day routers have typically tens of thousands of routes in their routing tables
- The approach requires no change to existing data path and routing protocols
- And it works pretty well!

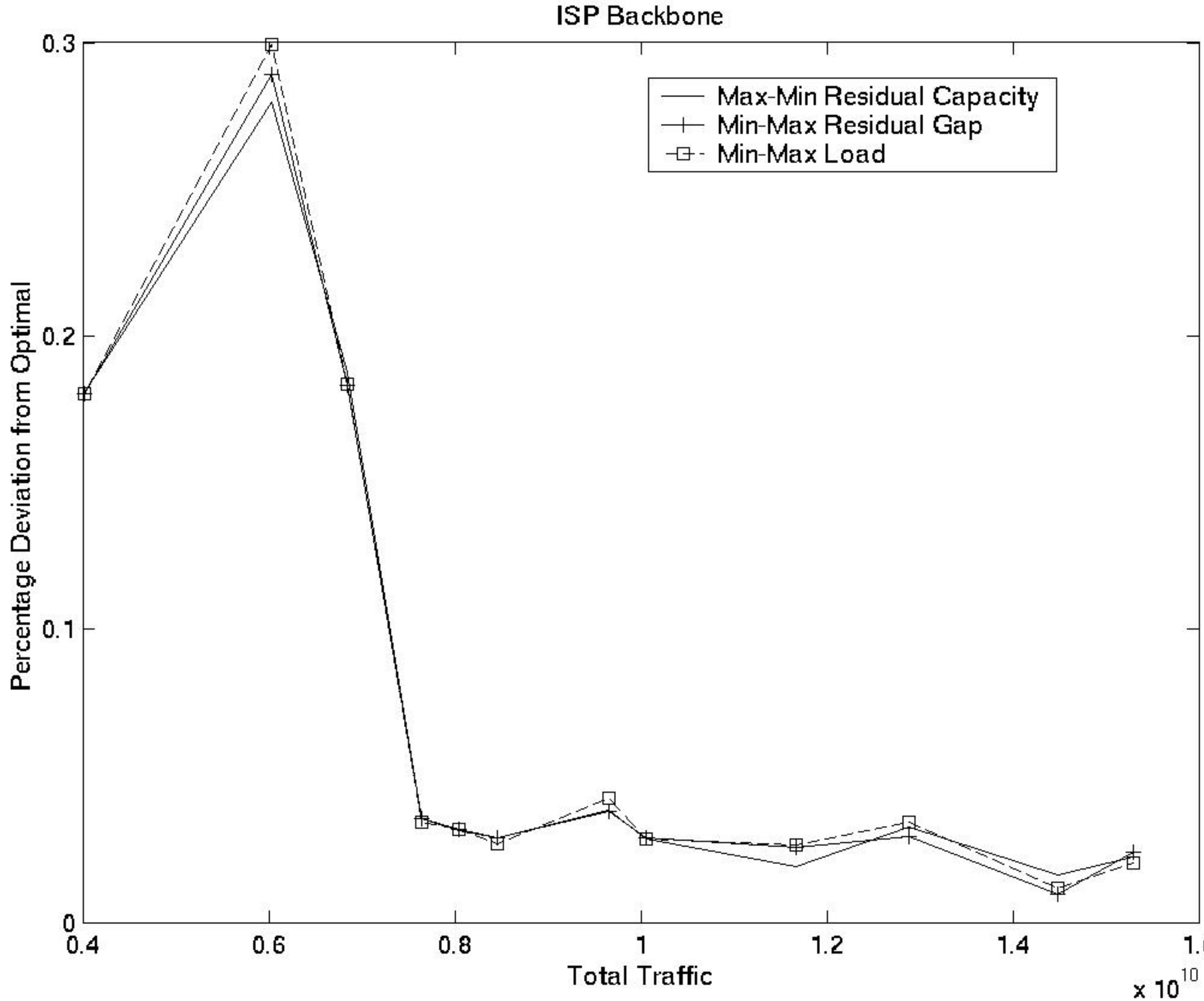
Sample Topology



Performance



Average Deviation



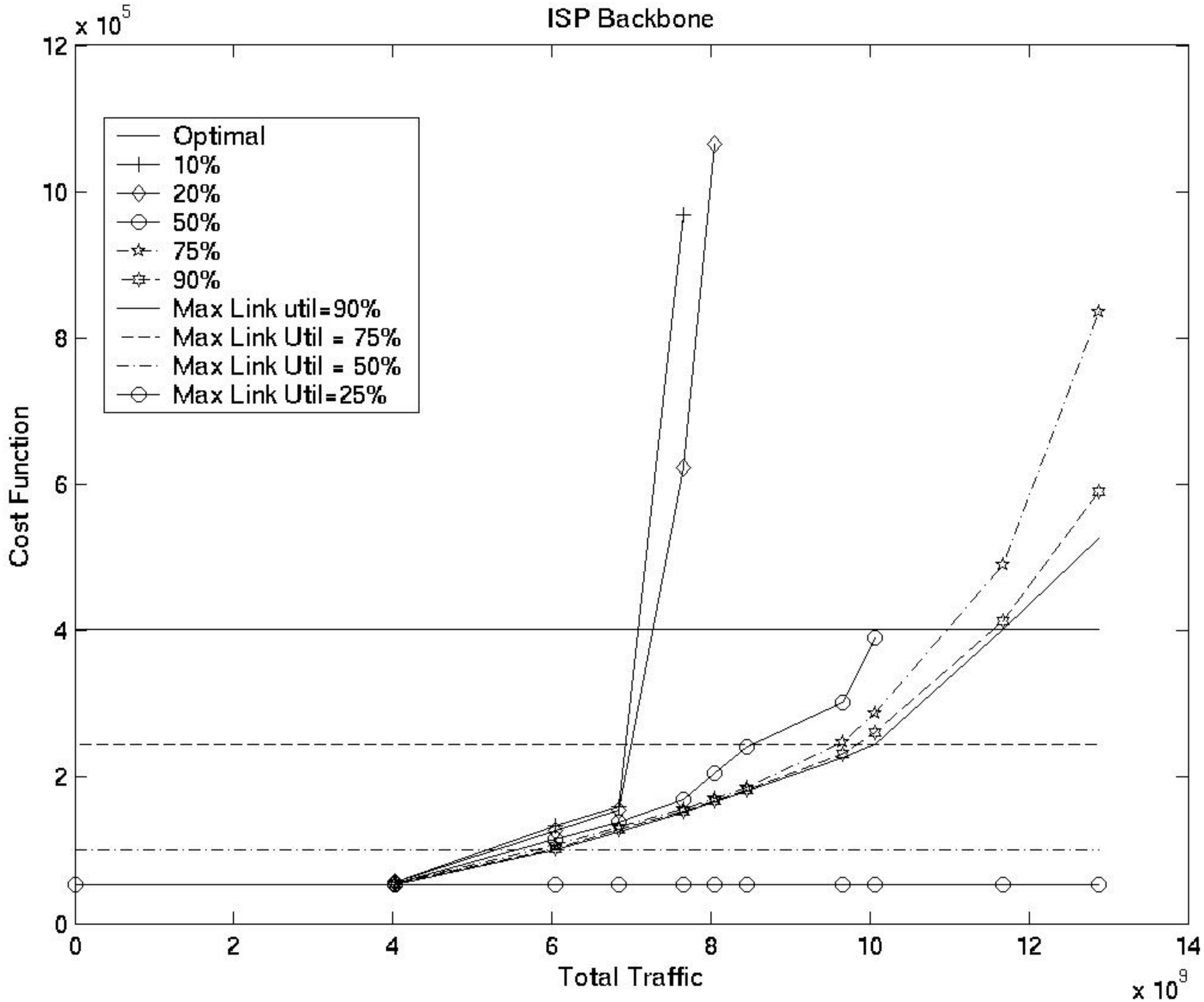
Trying to Make It Practical

- Configuring next hops for each routing table entry involves considerable overhead
  - It's simply not feasible in practice
- Do we really need to do it for all routes?
  - Traffic measurements indicate that a large portion of traffic is typically concentrated over a few route entries
- What is the trade-off?
  - What performance as a function of the number of routes for which we configure next hops

Top Routing Entries in Carried Traffic (30,700 Per node)

No of routes configured	% Routes Configured	%Accounted Traffic
160	0.52 %	10
210	0.65 %	20
620	2.0 %	50
1750	6.5 %	75
4158	13.5 %	90

Partial Configuration



Conclusion

- Our heuristics can give close to optimal performance (at least in the experiments we conducted).
- They can be implemented without any change to the data path.
- They require (small) control path changes.
  - Configuration overhead can be lowered significantly by choosing entries carefully.

- Note: Traffic measurement at the granularity of routing prefixes is needed.

Looking Further

There are some unresolved issues

- Response to link failures
  - Initial work [4] shows that link weights can be adapted to link failures
  - But there are limitations and computational cost can be high
- Response to traffic fluctuations
  - How to adapt to fluctuations in traffic intensity for a given route entry?
- Ensuring integer weights
  - May not be as much of an issue with TE extensions

References

1. Fortz & Thorup, "Internet traffic engineering by optimizing OSPF weights", Infocom 2000.
2. Z.Wang, Y.Wang, L.Zhang, "Internet traffic engineering without full mesh overlaying", Infocom 2001.
3. R.K. Ahuja, T.L Magnanti, J.B. Orlin, Network flows, Prentice Hall, 1993.

References

4. Fortz and Thorup, "Optimizing OSPF/IS-IS weights in a changing world", JSAC 2001.