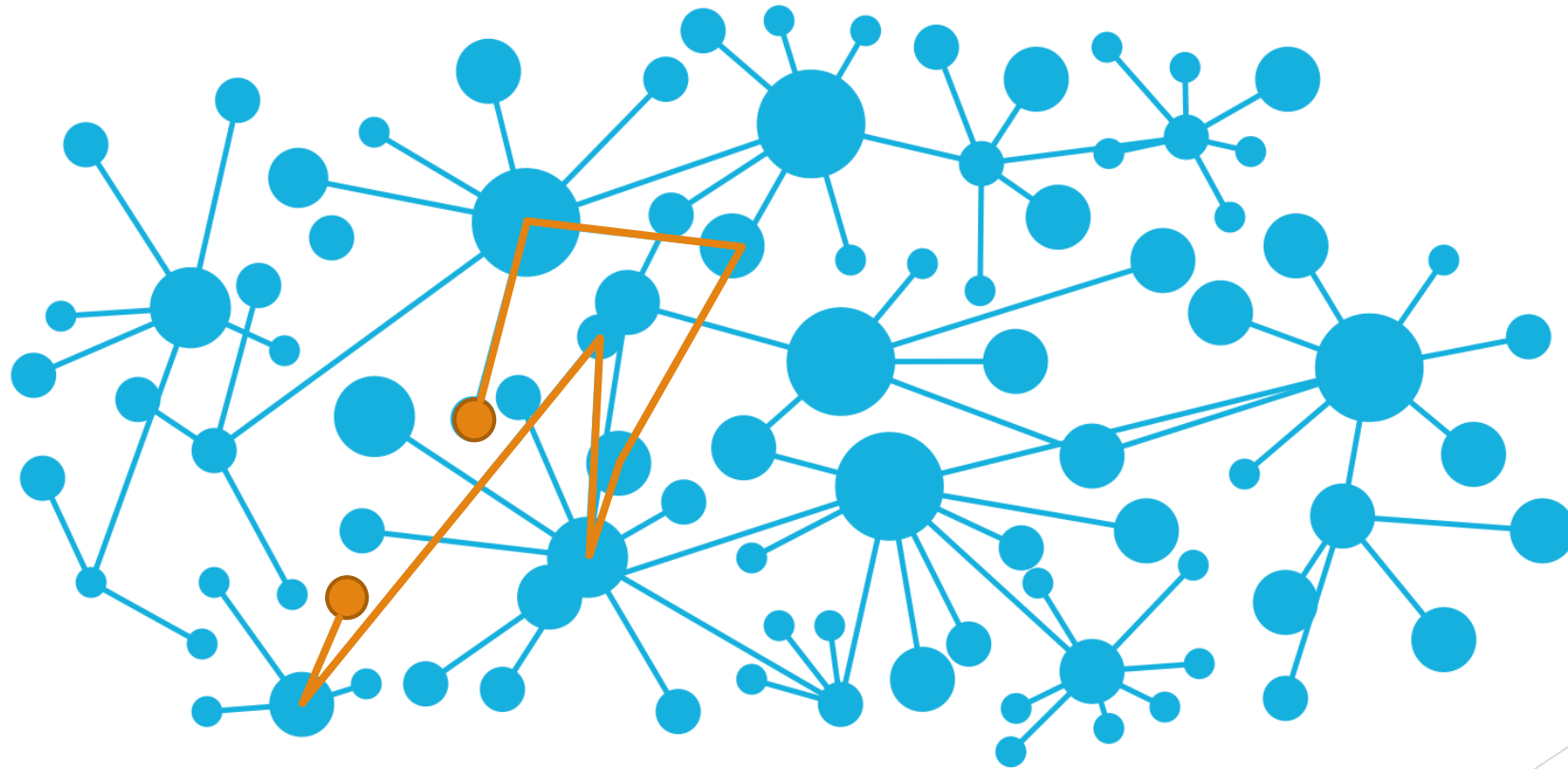


Comparing Oblivious and Robust Routing Approaches

Thibaut Cuvelier, Éric Gourdin

What is network routing?



What is network routing?

- ▶ **What path should each packet take throughout the network?**
- ▶ **Most basic solution: shortest path**
 - ▶ Fewest number of routers crossed
 - ▶ No prior knowledge of the flows that will traverse the network
- ▶ **Traffic engineering**
 - ▶ Optimise some metric: network congestion, end-to-end delay, etc.

Why robust/oblivious routing?

- ▶ In traffic engineering, a lot of **uncertainty!**
 - ▶ What is the traffic between a source and a destination?
 - ▶ How does it evolve over time?
- ▶ Robust/oblivious routing deals with this uncertainty!
- ▶ Ancillary question: should you invest in higher-bandwidth links?
 - ▶ Better routing leads to better hardware utilisation





How to perform traffic engineering

Using mathematical optimisation

Fundamental tool: multicommodity flow (MCF)

- ▶ Knowing the traffic the network will face, how to route it?
- ▶ For now, no uncertainty!

- ▶ How does the model work?
 - ▶ Objective function: minimise the maximum link utilisation
 - ▶ Usual formulation: path-based (column generation)
 - ▶ Variables: fraction of each demand in each path

Fundamental tool: multicommodity flow (MCF)

$\min \mu$

$$\sum_{p \in \mathcal{P}_d} routing_{d,p} = 1 \quad \forall d \in \mathcal{D}$$

$$\sum_{d \in \mathcal{D}} \sum_{\substack{p \in \mathcal{P}_d: \\ e \in p}} demand_d \times routing_{d,p} \leq \mu \times capacity_e \quad \forall e \in E$$

► Intuition?

- If μ is 1, at least one edge needs its full capacity, and has a 100% utilisation
- If μ is 0.5, no edge needs more than half its capacity: the worst utilisation is 50%

Oblivious routing



Uncertainty? Where?



- ▶ Problem of the MCF formulation?
 - ▶ Only **one** traffic matrix is considered!
- ▶ In practice: the traffic is probably far from this matrix
 - ▶ Even from the average matrix
 - ▶ Congestion can be arbitrarily far from that of the optimal routing of the average matrix
- ▶ **Oblivious routing**: consider **several** traffic matrices
 - ▶ Only those that correspond to the **worst** traffic conditions
 - ▶ Only **one** routing for the network that works reasonably well for any traffic scenario

The *new* MCF

- ▶ Main difference? Multiple traffic matrices, taken in Δ (details later)

$$\min \mu$$

$$\sum_{p \in \mathcal{P}_d} \text{routing}_{d,p} = 1 \quad \forall d \in \mathcal{D}, \boxed{\forall \mathcal{D} \in \Delta}$$

$$\sum_{d \in \mathcal{D}} \sum_{\substack{p \in \mathcal{P}_d: \\ e \in p}} \text{demand}_d \times \text{routing}_{d,p} \leq \mu \times \text{capacity}_e \times \boxed{\text{OPT}(D)} \quad \forall e \in E, \boxed{\forall \mathcal{D} \in \Delta}$$

- ▶ $\text{OPT}(D)$ is the optimum congestion for D
- ▶ In the end: μ is the worst ratio between the oblivious routing and optimal routing
- ▶ For the first constraint, no real impact, as long as all matrices $\mathcal{D} \in \Delta$ have the same origin-destination pairs

How to solve the new MCF?

- ▶ Δ is potentially infinite: all traffic matrices that could be routed through the network
 - ▶ Only depends on the topology and the allowed origin-destination pairs
 - ▶ No dependency on the usual traffic, for instance
- ▶ How can you solve an optimisation problem with infinitely many constraints?
- ▶ Iterative algorithm:
 - ▶ Compute a routing and its ratio μ
 - ▶ Find a new traffic matrix
 - ▶ If the congestion is worse than the current μ : add the traffic matrix
 - ▶ Otherwise: converged

How to solve the new MCF?

- ▶ Focus on the subproblem: how to find a matrix D to consider?
- ▶ For each edge: solve a new optimisation program
 - ▶ Objective function: maximise the congestion in that edge
 - ▶ Variables: demand matrix
 - ▶ Constraint: current oblivious routing



How bad is oblivious routing?

- ▶ It is robust against **any** traffic conditions
 - ▶ You would expect that this routing can be arbitrarily bad for any scenario
 - ▶ Theoretical proof: the ratio between the optimal routing for any given network conditions and the behaviour of the oblivious solution to the same conditions is bounded:
$$\mathcal{O}(\log^3 n)$$
where n is the number of nodes in the graph
- ▶ And in practice? According to simulations on random traffic matrices, the promises seem to hold
- ▶ Could still do better: not all situations are possible in practice

Robust routing



Maybe oblivious is too extreme

- ▶ Oblivious routing keeps everything under control
 - ▶ Whatever the uncertainty, it is dealt with
- ▶ In practice, not **that** useful
 - ▶ Operators monitor their networks and know what traffic they usually see
 - ▶ Future traffic can be estimated with a high accuracy ($\pm 5\%$)
- ▶ How about restraining the set of traffic matrices Δ ?
 - ▶ **Robust routing**



Smaller traffic matrix set Δ

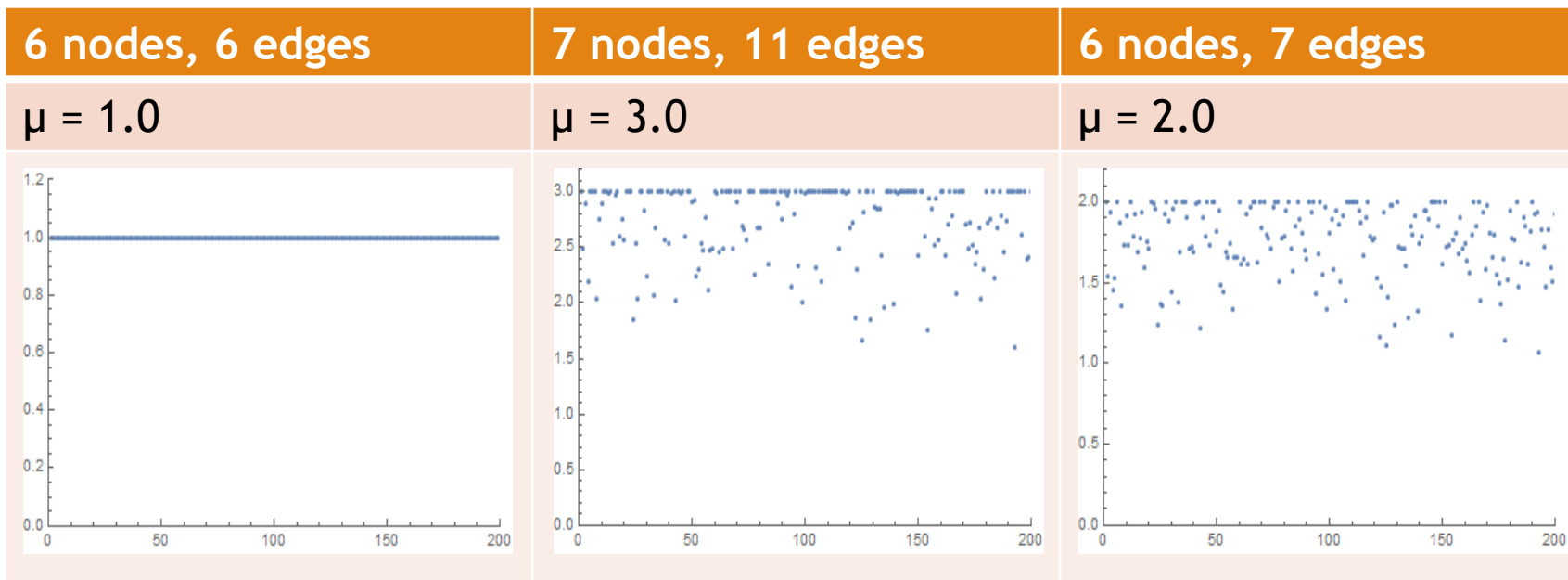
- ▶ Only real constraint:
 - ▶ The subproblem can still be solved “efficiently”
- ▶ Usual uncertainty sets:
 - ▶ Convex combinations of known traffic matrices
 - ▶ Box around some standard traffic (equivalent to intervals for each demand)
- ▶ All of these examples keep the subproblem convex!



Numerical results

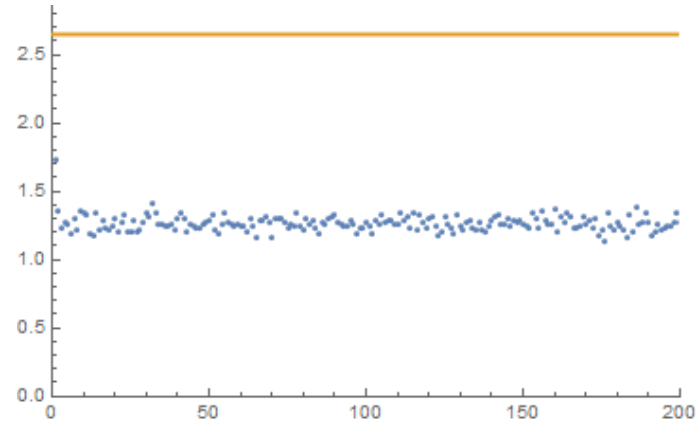
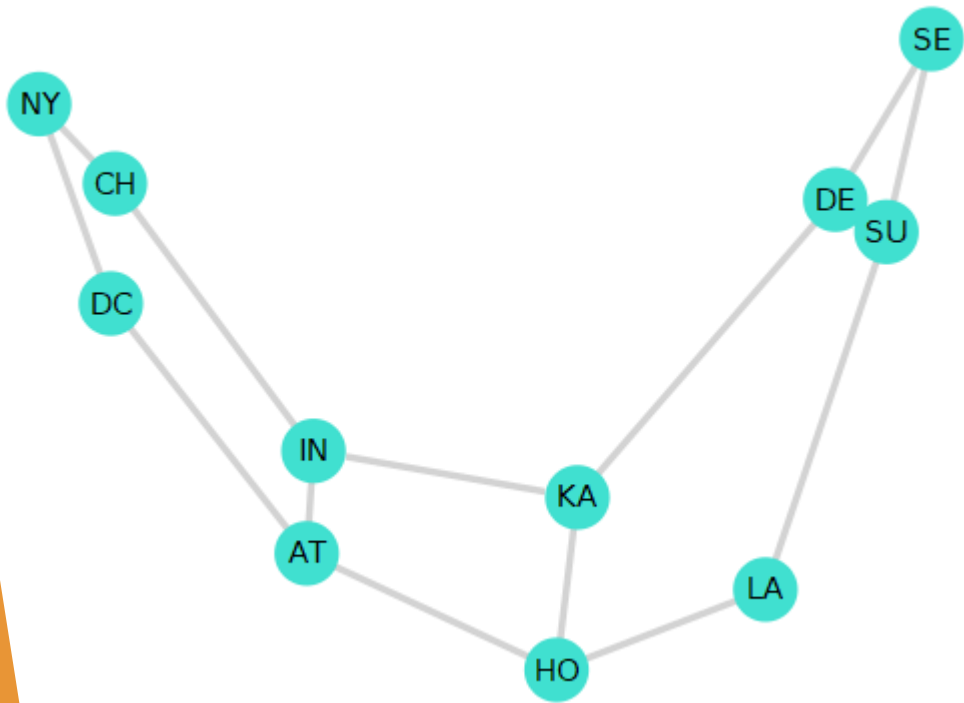
Main theoretical property: does the bound hold?

- ▶ In theory, solving the oblivious problem gives a **bound** on how bad the congestion can be when compared to optimal routing
 - ▶ To check whether this is the case: 200 experiments with random traffic (the origin-destination pairs are not random)
 - ▶ Three “synthetic” graphs



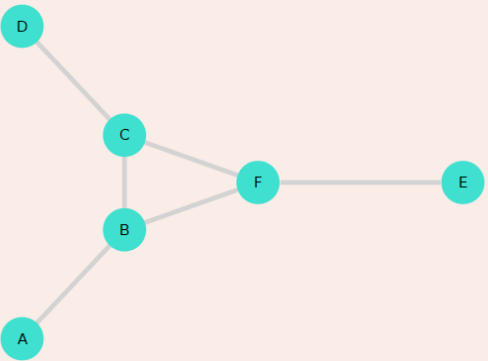
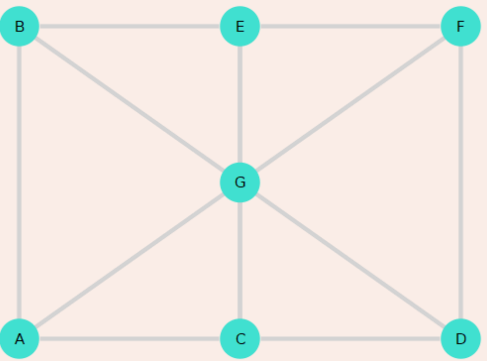
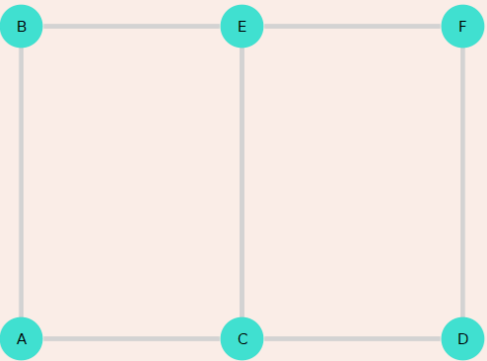
Main theoretical property: does the bound hold?

- ▶ And for a more realistic graph? Take Abilene, for instance
 - ▶ $\mu = 2.65$



More realistic uncertainty

- ▶ Consider a bounded uncertainty set Δ
 - ▶ Maximum deviation of 10% around a nominal value
 - ▶ Traffic can be predicted within 10% with high reliability

6 nodes, 6 edges	7 nodes, 11 edges	6 nodes, 7 edges
Oblivious $\mu = 1.0$	Oblivious $\mu = 3.0$	Oblivious $\mu = 2.0$
Robust $\mu = 1.0$	Robust $\mu = 1.65$	Robust $\mu = 1.1$
		

Conclusions

Conclusion and future work

- ▶ Oblivious routing is a very interesting paradigm
 - ▶ Implemented at Facebook between their datacenters, for instance
 - ▶ But not really well studied for now
 - ▶ Able to provide a trade-off between robustness and performance
- ▶ To tune this trade-off: robust routing
- ▶ The end of the story? What about failures (link, node)?

References

- ▶ Theoretical results about oblivious routing:
Räcke, Harald. "Minimizing congestion in general networks." In *The Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002*, pp. 43-52. IEEE, 2002.
 - ▶ In particular, Theorem 8
- ▶ First LP-based algorithm for oblivious routing:
Azar, Yossi, Edith Cohen, Amos Fiat, Haim Kaplan, and Harald Räcke. "Optimal oblivious routing in polynomial time." *Journal of Computer and System Sciences* 69, no. 3 (2004): 383-394.
- ▶ Semi-oblivious routing at Facebook:
Kumar, Praveen, Yang Yuan, Chris Yu, Nate Foster, Robert Kleinberg, Petr Lapukhov, Chiun Lin Lim, and Robert Soulé. "Semi-Oblivious Traffic Engineering: The Road Not Taken." In *USENIX NSDI*. 2018.

DO YOU HAVE ANY
QUESTIONS FOR ME?

