Oblivious Routing: Static Routing Prepared Against Network Traffic and Link Failures

Thibaut Cuvelier, Eric Gourdin

Oblivious Routing: Static Routing Prepared Against Network Traffic and Link Failures

Thibaut Cuvelier and Éric Gourdin, Orange Labs (France)

Internet is very dynamic

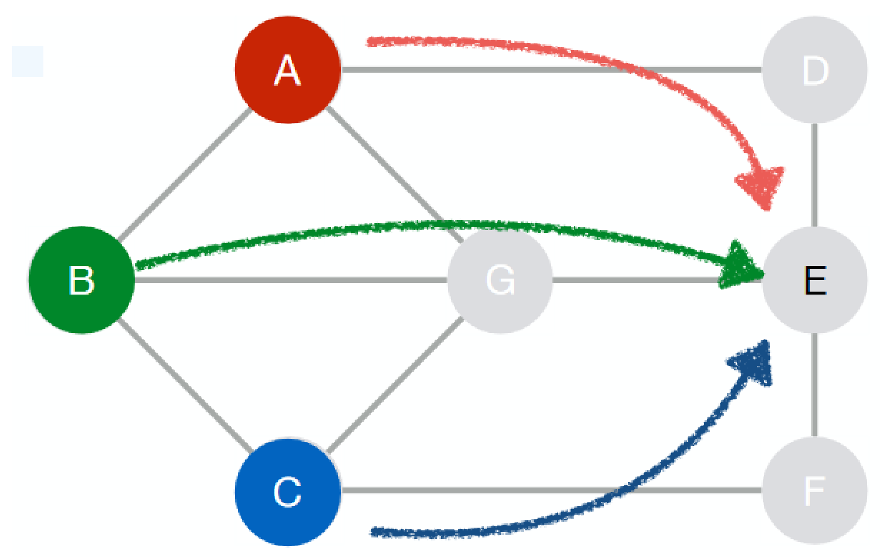
- Routing in Internet challenged by large throughput variations:
  - Day vs. night, week days vs. week ends
  - Popular events (like sports or Black Friday)
- Users expect excellent service around the clock!
- Network operators must fine-tune their routing to guarantee sufficient bandwidth and low enough latency in all conditions

How to compute a routing?

- Very common approach: IGP protocols and shortest path
  - An administrator chooses the "distance" metric between two routers, according to their own criteria: OSPF, IS-IS, RIP
  - **Often not capacity-aware**: only dealing with connectivity, not traffic
- More centralised traffic engineering using real-time measurements? Promised by SDN
  - Use optimisation tools (mostly linear programming — LP)
- **What about uncertainty?** It lies in traffic and failures
  - Discarded with usual protocols! However, SDN proposes to change the situation

Multi-commodity flow (MCF) formulation

- Optimisation variables:
  - flow $f_k(e)$ in each link $e \in E$ for each origin-destination pair $k \in K$, expressed in MB/s
  - μ, a capacity-reduction factor (reduce the capacity of the links by a factor μ)
- Constraints:
  - Link capacity C_e (scaled with μ): $\sum_{k\in K} f_k(e) \le \mu C_e, \quad \forall e \in E$
  - Flow conservation
- Objective: minimise μ, i.e. use the links as little as possible, be far below their capacity
  - Taking capacities as low as possible is equivalent to sending multiple times the demand, and to minimising the maximum congestion (defined as *load / capacity* for a link)



Oblivious routing

- Integrate **traffic uncertainty** into the MCF
- Optimise for all possible traffic matrices respecting the capacities
  - Minimise the worst ratio for all these matrices (congestion oblivious / optimum congestion)
- **Algorithm?** Iteratively limit the capacity of the edges:
  - Compute the demand that leads to the largest congestion for a given edge *e*, repeat for all edges, pick the worst one
  - Generate the corresponding capacity constraint
  - Start again until convergence
- Important theoretical result [Räcke, 2008]: for any traffic matrix, the maximum ratio is O(polylog # nodes)

Capacity variant

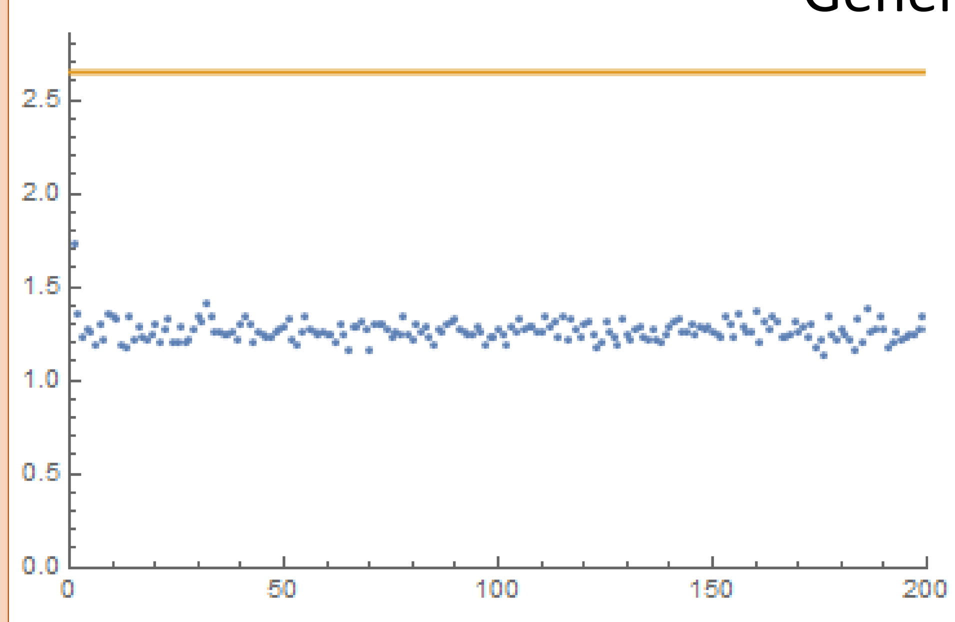
- Integrate **failure uncertainty** into the MCF, i.e. the **capacity** of the links may vary
- Optimise for all possible capacity matrices accepting a given (fixed) demand
  - The routing must maximise the amount of traffic for all these matrices
- **Algorithm?** Iteratively impose minimum of flows for each demand, i.e. demand constraints:
  - Compute the capacity that leads to the lowest amount of traffic for a given demand *d*, repeat for all demands, pick the worst one
  - Generate the corresponding constraint
  - Start again until convergence

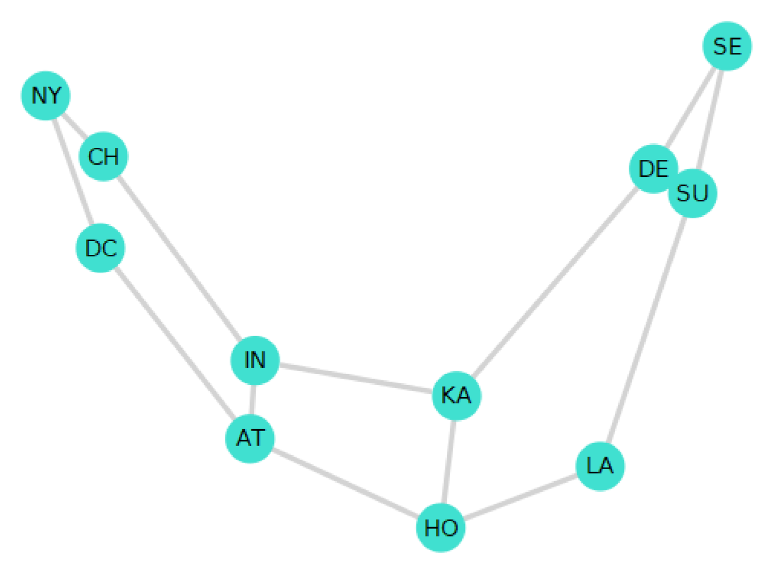
In practice

- Oblivious routing can scale (1,000 nodes, 1,000s edges, 10,000s demands) on a high-end laptop
- Abilene network: 3.5 seconds
  - 11 nodes, 14 edges, 103 demands
- AT&T network: 50 seconds
  - 26 nodes, 50 edges, 103 demands
- Real proprietary network: 1.5 hours

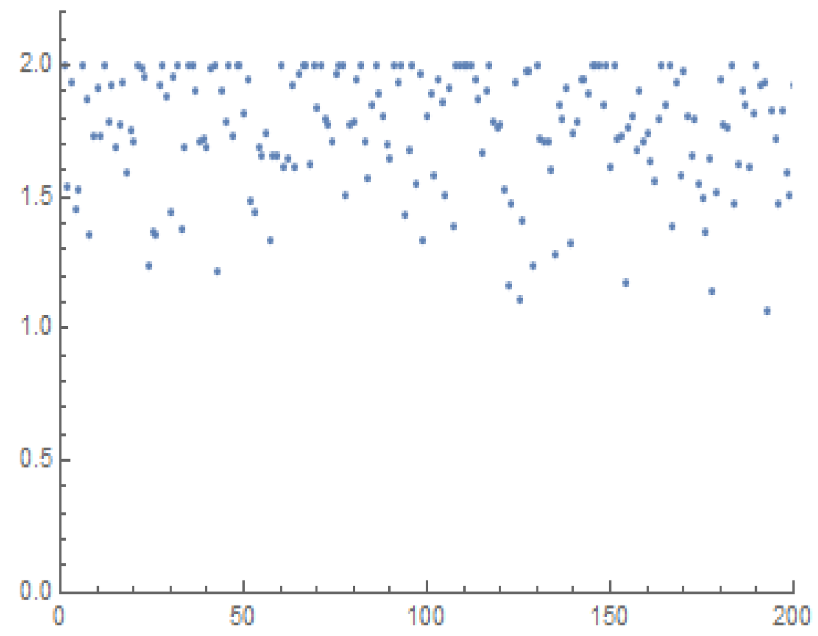
Numerical experiments (demand uncertainty only)

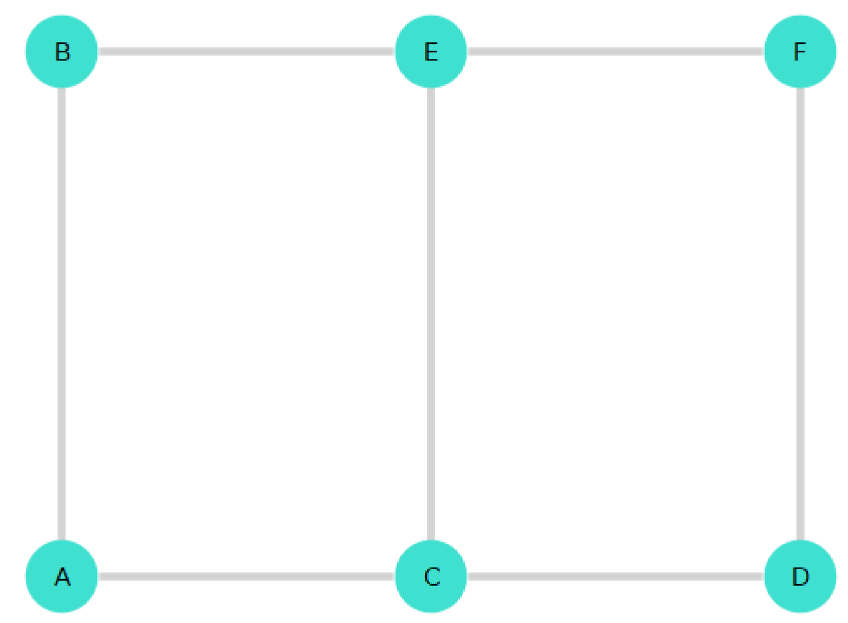
Generate a plausible traffic matrix, compute the ratio, rinse and repeat 200 times





Abilene





Extensions

- Generalise to other performance metrics: α-fairness, quality of experience (QoE). What happens to the theoretical guarantees?
- Optimising for all possible traffic matrices that respect the capacity constraints is probably too demanding
  - ⇨ Exploit traffic history to derive a less conservative uncertainty set
  - The traffic matrices must be a convex combination of previously seen matrices, within a "distance" to an average matrix, etc.
- Implement the optimisation algorithm in a distributed fashion (which is probably required by SDN for scalability)

Räcke, H. (2008). Optimal Hierarchical Decompositions for Congestion Minimization in Networks. In Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing (pp. 255–264). New York, NY, USA: ACM.