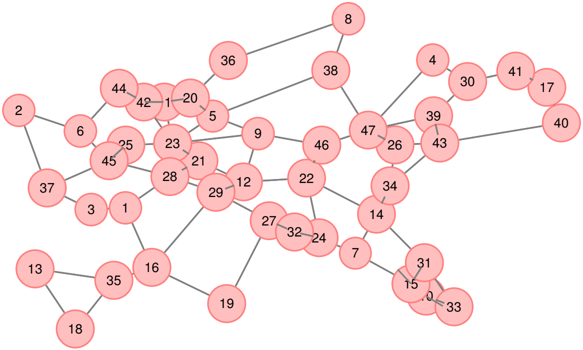
# On Optimization of Semi-Stable Routing in Multicommodity Flow Networks

Artur Tomaszewski[1], Michał Pióro[1]
Davide Sanvito[2], Ilario Filippini[2], Antonio Capone[2]

[1]Institute of Telecommunications, Warsaw University of Technology
[2]Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano

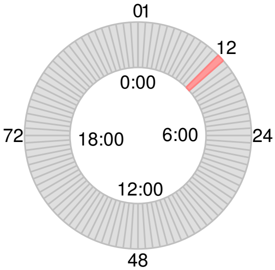
International Network Optimization Conference INOC
Avignon, France, June 12-14, 2019

# network topology




- given: a regional packet network in an EU country with 47 routers and 140 directed 4Gb/s transmission links
- model: a directed graph $\mathcal{G}$ with a set of vertices $\mathcal{V}$ and a set of arcs $\mathcal{E}$; $c(e)$ is the capacity of arc $e \in \mathcal{E}$

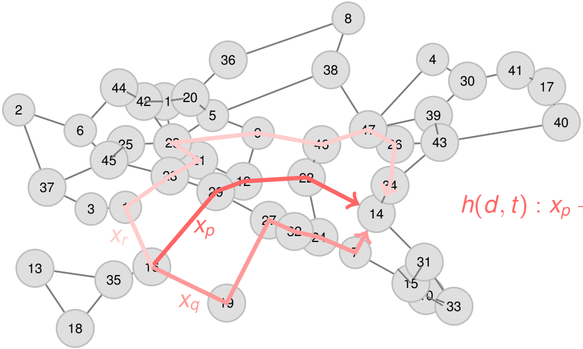
# traffic measurements




| | 1 | 2 | ··· | 47 |
|---|---|---|---|---|
| 1 | | 0.699 | ··· | 1.543 |
| 2 | 0.699 | | ··· | 1.059 |
| ··· | ··· | ··· | ··· | ··· |
| 47 | 1.543 | 1.059 | ··· | |

- given: node traffic measured on Wednesday in 15-minute intervals: 96 inter-node 47*x*47 traffic matrices of interval-average bitrates (0.028Mb/s to 1712.044Mb/s)
- model: a set of time-points (intervals) $\mathcal{T} \equiv \{0, 1, \ldots, T-1\}$, and a set of demands $\mathcal{D}$; $h(d, t)$ is the volume of demand $d \in \mathcal{D}$ at timepoint $t \in \mathcal{T}$

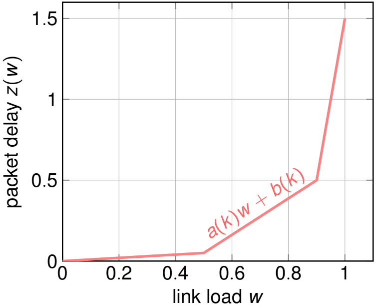
# traffic routing




$h(d,t): x_p + x_q + x_r = 1$

- task: to determine a set of paths for each demand, and for each timepoint the split of demands' traffic – fraction of the demand's volume assigned to each path
- model: $\mathcal{P}(d)$ is a set of allowed paths of demand $d \in \mathcal{D}$; variable $x^t_{dp} \in [0,1]$ denotes the fraction of $d$'s volume assigned to path $p \in \mathcal{P}(d)$ at timepoint $t \in \mathcal{T}$

# performance metric

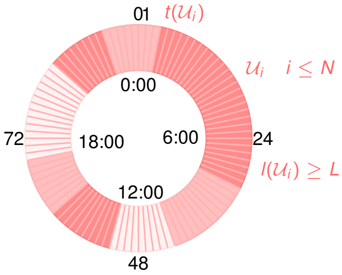



- QoS metric is packet delay being a convex piece-wise linear function of link load:

$$z(w) \equiv \min \{x : x \geq a(k)w + b(k), k \in \mathcal{K}\}$$

- objective: to minimize the total (timepoint-wide and link-wide) network delay
- model: min $\{\sum_{t \in \mathcal{T}} \sum_{e \in \mathcal{E}} z(w_e^t) : w_e^t = \sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}(d,e)} h(d,t) x_{dp}^t / c(e)\}$

# semi-stable routing – cluster set




- too frequent routing changes affect network stability, and too many routing configurations increase network management cost
- routing cluster: set $\mathcal{U} \subseteq \mathcal{T}$ of consecutive timepoints that have the same routing configuration; starting timepoint $t(\mathcal{U})$, length $l(\mathcal{U})$, routing $x_{\mathcal{U}}$
- problem: to find a partition of $\mathcal{T}$ into at most $N = 8$ routing clusters, each of length at least $L = 8$, that minimizes the total network delay

# cluster's routing optimization – routing problem

$$RP(\mathcal{U}) : Z(\mathcal{U}) = \min \sum_{t \in \mathcal{U}} \sum_{e \in \mathcal{E}} z_e^t \quad (1a)$$

$$w_e^t = \sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}(d,e)} h(d,t) x_{dp} / c(e) \qquad t \in \mathcal{U}, e \in \mathcal{E} \quad (1b)$$

$$\sum_{p \in \mathcal{P}(d)} x_{dp} = 1 \qquad d \in \mathcal{D} \quad (1c)$$

$$z_e^t \geq a(k) w_e^t + b(k) \qquad t \in \mathcal{U}, e \in \mathcal{E}, k \in \mathcal{K} \quad (1d)$$

$$x_{dp} \in [0, 1] \qquad d \in \mathcal{D}, p \in \mathcal{P}(d) \quad (1e)$$

- an LP optimizing (stable) routing for a given single cluster $\mathcal{U}$: variables $x_{dp}$ define a common split of traffic for all timepoints in $\mathcal{U}$
- merely, a link-path formulation of the multi-state capacitated multicommodity flow problem with a convex piece-wise linear objective function
- static routing: if solved for $\mathcal{U} \equiv \mathcal{T}$, $Z(\mathcal{U})$ is an upper bound of the optimal cost value of the semi-stable routing problem (and $\{\mathcal{U}\}$ is a feasible solution)
- dynamic routing: if solved for every single-timepoint cluster $\mathcal{U} \equiv \{t\}$, $\sum_{\mathcal{U}} Z(\mathcal{U})$ is a lower bound of the optimal cost value of the semi-stable routing problem

# experiments and observations

- the routing optimization problem can be solved efficiently (it takes $I(\mathcal{U})$ seconds on a laptop) using:
  - path generation and a shortest path algorithm
  - warm start for the master problem using the last basis
  - accumulation of generated paths throughout consecutive runs (clusters)
- combining routing optimization with cluster set optimization in one formulation requires introducing multiple sets of routing variables and additional coupling binary variables – it is not an LP any more, and is much larger
- we have tried a number of both exact and heuristic problem formulations combining cluster and routing optimization – they appeared too complex (even their linear relaxations)

# observations

- there are just $O(T^2)$ potential routing clusters – they have lengths between $L$ and $T - L$ and can start at any of $T$ timepoints; moreover, a number of cluster's routing optimization problems can be solved in parallel on multiple processors
- however, with 5-minute measurement intervals and one-week-horizon $T$ can be as large as 2016, whereas $L$ equals 24 if routing reconfiguration after 2 hours is allowed
- we aim at networks with 100 through 200 nodes – the size of the cluster's routing optimisation problem will increase more than 10 times, and the computation time much more

# suggestions

- the semi-stable routing optimization problem should be decomposed into clusters optimization and routing optimization
- the routing optimization problem should not be solved for every potential cluster, and more than that – it should be solved for a small subset of clusters only
- question: how to couple the two problems?

# clusters optimization – partitioning problem

$$PP(\mathscr{C}) : Y(\mathscr{C}) = \min \textstyle\sum_{t \in \mathcal{T}} y^t \tag{2a}$$

$$\textstyle\sum_{t \in \mathcal{T}} u^t \leq N \tag{2b}$$

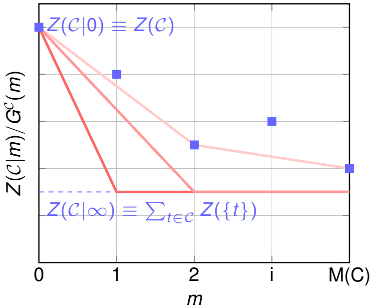
$$\textstyle\sum_{0 \leq k \leq L-1} u^{t \oplus k} \leq 1 \qquad t \in \mathcal{T} \tag{2c}$$

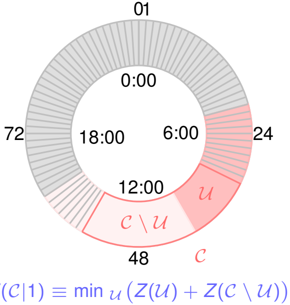
$$\textstyle\sum_{t \in \mathcal{C}} y^t \geq G^{\mathcal{C}}\left(\sum_{t \in \mathcal{C} \setminus \{t(\mathcal{C})\}} u^t\right) \qquad \mathcal{C} \in \mathscr{C} \tag{2d}$$

$$u^t \in \mathbb{B}, y^t \in \mathbb{R}_+ \qquad t \in \mathcal{T} \tag{2e}$$

- a MIP with binary variables – $u^t$ decides if a routing cluster starts at $t$; there can be at most $N$ starts in total and only 1 start in any set of $L$ consecutive timepoints
- $\mathscr{C}$ is a family of control clusters and $G^{\mathcal{C}}$ is an approximation function that provides a lower bound of the total delay over cluster $\mathcal{C}$ such that $G^{\mathcal{C}}(0) = Z(\mathcal{C})$
- property 1: $Y(\mathscr{C})$ is a lower bound of the optimal semi-stable routing cost, and $u^{t*}$'s define a feasible solution and provide an upper bound of the optimal semi-stable routing cost
- property 2: if all clusters defined by $u^{t*}$'s are contained in the control family $\mathscr{C}$, they are an optimal partitioning of $\mathcal{T}$, and $Y(\mathscr{C})$ is its true cost

# delay-lower-bound function




- $G^{\mathcal{C}}(m)$ is a lower bound of the total delay over control cluster $\mathcal{C}$ when $\mathcal{C}$ overlaps $m+1$ routing clusters, $m \le M(\mathcal{C}) := \lceil \frac{l(\mathcal{C})-1}{L} \rceil$, such that $G^{\mathcal{C}}(0) = Z(\mathcal{C})$
- $G^{\mathcal{C}}$ needs to be a convex piece-wise linear function with easy-to-determine coefficients
- we require that $G^{\mathcal{C}}(m) \le Z(\mathcal{C}|m)$; e.g., it can be a convex envelope of points with values $Z(\mathcal{C}|0)$, $Z(\mathcal{C}|1)$, $Z(\mathcal{C}|\infty)$, which can be efficiently computed

# basic method

8010 paths

| PREPROCESSING | BRANCH-AND-BOUND | POSTPROCESSING |
|---|---|---|
| $RP(\mathcal{C}) : 1 \leq l(\mathcal{C}) \leq L+1$ | $PP(\{\mathcal{C}\}) : L \leq l(\mathcal{C}) \leq L+1$ | $RP(\mathcal{U}^*)$ |
| 1h:7m:46s 3.33% | 2s | 1m:16s 0.25% |

- preprocessing: solve $RP(\mathcal{C})$ LPs for all 964 (11%) clusters $\mathcal{C}$ of size less or equal $L+1=9$, for which $M(\mathcal{C})=1$
- branch-and-bound: solve $PP(\mathscr{C})$ MIP; family $\mathscr{C}$ consists of 192 clusters of length $L=8$ and $L+1=9$, and function $G^{\mathcal{C}}$ is based on $Z(\mathcal{C}|0)$ and $(Z(\mathcal{C}|1)$
- postprocessing: solve $RP(\mathcal{U}^*)$ LPs for all clusters $\mathcal{U}^*$ that constitute the resulting partitioning $\{\mathcal{U}^*\}$
- large space for modifications: selection of control clusters sizes, selection of control clusters starting timepoints, definition of the bounding function, extension of the control family in a loop, etc.

# extended method

48 incumbents / 32 user cuts

| PREPROCESSING | BRANCH-AND-BOUND-AND-CUT |
|---|---|
| $RP(\mathcal{C}) : 1 \leq I(\mathcal{C}) \leq T/N$ | $PP(\{\mathcal{C}\}) : L \leq I(\mathcal{C}) \leq T/N$ & $RP(\mathcal{U}^l)$ |
| 2h:44m:45s 3.33% | 8m:50s 0.0% |

- preprocessing: solve $RP(\mathcal{C})$ LPs for all 1,152 (15%) clusters $\mathcal{C}$ of size less or equal $T/N = 12$ (average size with $N$ clusters)
- branch-and-bound-and-cut: solve $PP(\mathscr{C})$ MIP generating cuts whenever an incumbent solution $\{\mathcal{U}^l\}$ is found; family $\mathscr{C}$ consists of 480 clusters of length $L = 8$ through $T/N = 12$, and function $G^{\mathcal{C}}$ is based on $Z(\mathcal{C}|0)$ and $(Z(\mathcal{C}|1)$
- cut generation: solve $RP(\mathcal{U}^l)$ for each $\mathcal{U}^l$ that does not belong to $\mathscr{C}$, and if its total delay is greater than the cost approximated by $PP$, reject the incumbent and add $\mathcal{U}$ to $\mathscr{C}$ by adding lazy constraints / user cuts