Demand-oblivious routing: distributed vs. centralized approaches

Gábor Rétvári and Gábor Németh
{retvari, nemethgab}@tmit.bme.hu

High Speed Networks Laboratory
Department of Telecommunications and Media Informatics
Budapest University of Technology and Economics
Budapest, HUNGARY
Introduction

Routing optimization is hard without a good traffic matrix
Rate-adaptive routing: adapt routing to the actual demands
Build on demand-oblivious routing and play out the “distributed-centralized” trade-off
Our main tool: network geometry
Network geometry

Associate geometric objects with capacititated networks
Infer interesting properties

\[(s_1, d_1) = (3, 4)\]
\[(s_2, d_2) = (1, 4)\]
The flow polytope

The set of legitimate routings

More precisely, the set of path-flows \( u \) the network can accommodate, subject to link capacities
The throughput polytope

The set of admissible traffic matrices

More precisely, the set of aggregate flows $\theta$ realizable in the network, subject to link capacities
Capacity scaling

Scaling the link capacities equals scalar multiplying the corresponding polytopes
Rate-adaptive routing

Adjust path flows according to actual user demands
A routing function tells how to map a traffic matrix to path-flows
\[ u = S(\theta) \]
We only treat affine routing functions
\[ u = F\theta + g \]
where \( F \) is a matrix and \( g \) is a constant transposition
For the \( k \)th user: \( u_k = S_k(\theta) = F_k\theta + g_k \)
Already broad enough to describe single path routing, ECMP, oblivious routing, and many more
Adaptive routing: distributed model

The flow sent to a path depends on local information exclusively

\[ S \text{ is distributed if } \frac{\partial S_k}{\partial \theta_l} = 0 \text{ wherever } k \neq l \]
Demand-oblivious routing

Use the same set of traffic splitting ratios without respect to the traffic matrix

Choose the one that minimizes the link over-utilization experienced over any admissible traffic matrix

\[
\begin{pmatrix}
  u_1 \\
  u_2 \\
  u_3
\end{pmatrix} = \begin{pmatrix}
  \frac{1}{3} & 0 \\
  \frac{2}{3} & 0 \\
  0 & 1
\end{pmatrix} \begin{pmatrix}
  \theta_1 \\
  \theta_2
\end{pmatrix} + \begin{pmatrix}
  0 \\
  0 \\
  0
\end{pmatrix}
\]

Distributed and semi-static, so reasonably scalable
The problem with oblivious routing

An oblivious routing function might order infeasible routing to some admissible traffic matrices
A geometric interpretation

Scale the flow polytope $M$ up until it eventually contains all the possible path flows $S(T)$

$$\min \alpha : S(T) \subseteq \alpha M$$
Adaptive routing: centralized model

Let the routing function depend on global information

\[
\begin{pmatrix}
    u_1 \\
    u_2 \\
    u_3
\end{pmatrix} =
\begin{pmatrix}
    1 & 1 \\
    0 & -1 \\
    0 & 1
\end{pmatrix}
\begin{pmatrix}
    \theta_1 \\
    \theta_2
\end{pmatrix} +
\begin{pmatrix}
    -1 \\
    1 \\
    0
\end{pmatrix}
\]
Compound routing functions

Associate different routings to different regions of the throughput polytope: $S = \{(R^i, S^i) : i \in \mathcal{I}\}$

$R_1$: if $\theta_1 + \theta_2 \leq 1$ then

$$
\begin{pmatrix}
    u_1 \\
    u_2 \\
    u_3 
\end{pmatrix} =
\begin{pmatrix}
    1 & 0 \\
    0 & 0 \\
    0 & 1 
\end{pmatrix}
\begin{pmatrix}
    \theta_1 \\
    \theta_2 
\end{pmatrix}
$$

$R_2$: if $\theta_1 + \theta_2 \geq 1$ then

$$
\begin{pmatrix}
    u_1 \\
    u_2 \\
    u_3 
\end{pmatrix} =
\begin{pmatrix}
    0 & -1 \\
    1 & 1 \\
    0 & 1 
\end{pmatrix}
\begin{pmatrix}
    \theta_1 \\
    \theta_2 
\end{pmatrix} +
\begin{pmatrix}
    1 \\
    1 \\
    0 
\end{pmatrix}
$$
Compound, centralized routing functions

**Theorem:** for any network, there is a continuous, compound, centralized affine routing function that can route any admissible traffic matrix without link over-utilization.

Distributed:
- Simple
- Scalable
- But inefficient

Centralized:
- Stable
- Feasible
- Optimizable
- Not quite scalable
Scalability of centralized adaptive routing

The number of regions and routing functions needed for optimal adaptive routing usually increases exponentially with the complexity of the network.

![Graph showing the relationship between number of regions and number of users. The x-axis represents the number of users, ranging from 1 to 9, and the y-axis represents the number of regions, ranging from 10^1 to 10^6. The graph shows an exponential increase in the number of regions as the number of users increases.]
Hybrid centralized-distributed model

The central controller computes $S = \{(R^i, S^i) : i \in I\}$, where individual routing functions $S^i$ are distributed.

Observes the actual traffic matrix $\theta$, chooses the region $\theta \in R_i$ and downloads the corresponding $S^i$ to the routers.
Hybrid oblivious routing algorithm

HYBRID_OBLIVIOUS_ROUTING($T$)

function HYBRID_OBLIVIOUS_ROUTING($X$)

Compute an oblivious routing function $S$ for $X$

if $\alpha$ falls beyond some configured limit then

store $S$ and return

end if

$(k, t_k) \leftarrow \text{BEST\_CUT}(X)$

HYBRID_OBLIVIOUS_ROUTING($X \cap T \cap \{\theta : \theta_k \leq t_k\}$)

HYBRID_OBLIVIOUS_ROUTING($X \cap T \cap \{\theta : \theta_k \geq t_k\}$)

end function
Hybrid oblivious routing algorithm

\[ R_1 : \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} \]

\[ R_2 : \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} + \begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix} \]
Only a few cuts can make a difference

The oblivious ratio steadily improves as we add more cuts

![Graph showing the oblivious ratio against the number of regions for different cases.](image-url)
Conclusions

Rate-adaptive routing: discover the distributed-centralized spectrum
Demand-oblivious routing is scalable but inefficient
We presented the first ever optimal rate-adaptive routing algorithm
  – provably feasible, stable and optimizable
  – heavily centralized, so hard to implement
  – scales poorly
The hybrid distributed-centralized scheme seems to unify the advantages of the two