Link-State Routing Can Achieve Optimal Traffic Engineering: From Entropy To IP

Dahai Xu, Ph.D.

Florham Park
AT&T Labs - Research

Joint work with Mung Chiang and Jennifer Rexford (Princeton University)

INFOCOM 2008
Apr. 16, 2008

Outline

Outline

Minimum-cost Multicommodity Flow

- Minimum-cost Multicommodity Flow Problem
  - Classical Convex Optimization problem
  - Aliases
    - Optimal Routing: *Data Networks* [Bertsekas-Gallager]
    - Optimal Traffic Engineering: IP congestion control
    - ...
- Question: can we realize Optimal Routing with link-state routing?

City Traffic Control

- Big cities suffer from traffic congestion during rush hours
- The traffic to a same destination is a commodity

City Traffic Control

- Big cities suffer from traffic congestion during rush hours
- The traffic to a same destination is a commodity
- Traffic control to realize optimal commodity solution:
  - Explicit Routing
  - Road Price

Traffic Control with Explicit Routing

- Before leaving home, every driver signs in a web-site to get an assigned route to the destination
- Could be optimal but with high overhead

Traffic Control with Road Price

- Balance traffic by setting price for each road segment
- More feasible than Explicit Routing

Traffic Control with Road Price

- Balance traffic by setting price for each road segment
- More feasible than Explicit Routing
- Assumption I: all drivers choose the "cheapest" path (even splitting if multiple cheapest paths)
  $\Rightarrow$ Impossible to achieve optimal routing and NP-hard to find road prices [Fortz-Thorup, Infocom-00]

Traffic Control with Road Price

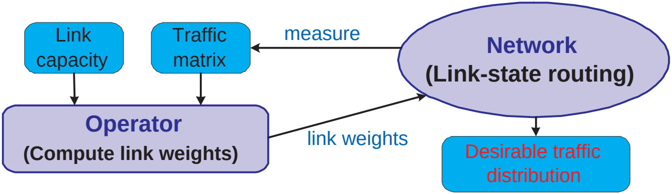
- Balance traffic by setting price for each road segment
- More feasible than Explicit Routing
- Assumption I: all drivers choose the "cheapest" path (even splitting if multiple cheapest paths)
  $\Rightarrow$ Impossible to achieve optimal routing and NP-hard to find road prices [Fortz-Thorup, Infocom-00]
- Assumption II:
  - More drivers choose the "cheapest" path
  - Fewer drivers choose more "expensive" path expecting less congestion (delay)

  $\Rightarrow$ Always achieve optimal routing and Convex Optimization to find road prices [Xu-Chiang-Rexford, Infocom-08]

Link-State Routing

- Routers
  - Exchange link weights (states) with Interior Gateway Protocols (IGPs): e.g. OSPF (Open Shortest Path First)
  - Distributively determine "next hop" to forward a packet/split traffic
- Network operator configures link weights to guide routing
  $\Rightarrow$ Traffic Engineering

Tuning Link Weights



- Traffic Engineering (TE): based on the offered traffic matrix
  - Traffic matrix: rate of traffic between each node pair from measurement
  - Centralized and off-line
  - Network-wide convex optimization objective: minimizes key metrics like max link utilization, sum of M/M/1 delay at each link, etc.

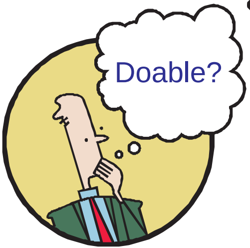
Why Link Weights?

- Low overhead: one parameter for each unidirectional link
- Hop-by-hop forwarding: no tunneling, no history, no per-flow statistics.
- Robust: routers automatically recompute new routes in case of topology changes
- Effective: changing a few link weights is sufficient to alleviate network congestion

Numerous Attempts to Realize Optimal TE with Link-state Routing Protocol

- Wang-Wang-Zhang-INFOCOM-01: *"Internet traffic engineering without full mesh overlaying"*
- Sridharan-Guérin-Diot-INFOCOM-03: *"Achieving Near Optimal Traffic Engineering Solutions in Current OSPF/ISIS Networks"*
- Fong-Gilbert-Kannan-Strauss-Algorithmica-05: *"Better Alternatives to OSPF Routing"*
- Xu-Chiang-Rexford-INFOCOM-07: *"DEFT: Distributed Exponentially-weighted Flow Splitting"*
- ...

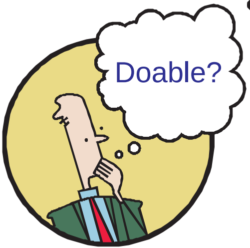
Open Questions



Optimal TE with ONLY link weights?

Find link weights in a tractable way?

Open Questions



Optimal TE with ONLY link weights?

Find link weights in a tractable way?

YES !

NEM/PEFT [Xu-Chiang-Rexford, Infocom-08]

Outline

Notation

- Directed graph: N nodes and E links
- Inputs

$D(s,t)$	Traffic demand from s to t
$c_{u,v}$	Capacity of link (u,v)

- Variables

$w_{u,v}$	Weight for link (u,v)
$f^t_{u,v}$	Commodity flow on link (u,v) destined to t
$f_{u,v}$	$\triangleq \sum_t f^t_{u,v}$, Total flow on link (u,v)

Optimal TE Via Multicommodity-Flow

COMMODITY Problem:

minimize	$\Phi(\{f_{u,v}, c_{u,v}\})$	convex objective
subject to	$\sum_{v:(s,v)\in\mathbb{E}} f^t_{s,v} - \sum_{u:(u,s)\in\mathbb{E}} f^t_{u,s} = D(s,t)$	flow conservation
	$f_{u,v} \triangleq \sum_{t\in\mathbb{V}} f^t_{u,v} \leq c_{u,v}$	capacity constraint
variables	$f_{u,v} \geq f^t_{u,v} \geq 0.$	link flow, commodity flow
input	$D(s,t), c_{u,v}$	demand, capacity

Optimal TE Via Multicommodity-Flow

COMMODITY Problem:

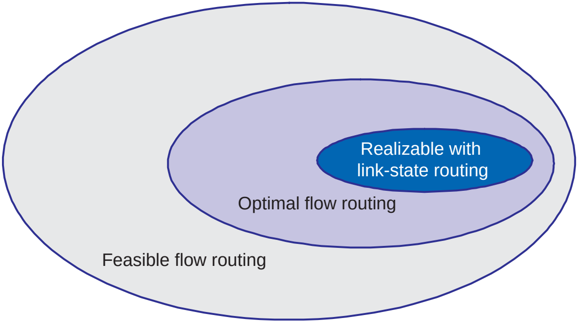
minimize	$\Phi(\{f_{u,v}, c_{u,v}\})$	convex objective
subject to	$\sum_{v:(s,v)\in\mathbb{E}} f^t_{s,v} - \sum_{u:(u,s)\in\mathbb{E}} f^t_{u,s} = D(s,t)$	flow conservation
	$f_{u,v} \triangleq \sum_{t\in\mathbb{V}} f^t_{u,v} \leq c_{u,v}$	capacity constraint
variables	$f_{u,v} \geq f^t_{u,v} \geq 0.$	link flow, commodity flow
input	$D(s,t), c_{u,v}$	demand, capacity

- Convex optimization (efficiently solvable).
- Can be realized with explicit routing: set up N^2E tunnels
- Link-state routing: E parameters

Necessary Capacity

- Necessary Capacity
  - $\widetilde{c}_{u,v} \triangleq f_{u,v}$: Total traffic on each link in optimal solution of COMMODITY
  - Minimal set of link capacities to realize optimal TE
- Set link weights with only necessary capacities

Intuition Behind the Theory



- Numerous ways of flow-level routing to realize optimal TE (different traffic distribution on the paths)
- Choose the flow-level routing which can be realized with link-state routing.
- How? Pick an additional objective function for these optimal flow-level routings

Network Entropy Maximization

- Assume we can enumerate all the paths from s to t, $P^i_{s,t}$. (only for analysis purpose)
- $x^i_{s,t}$: probability (fraction) of forwarding a packet of demand $D(s,t)$ to the i-th path ($P^i_{s,t}$)

subject to	$\sum_{s,t,i:(u,v)\in P^i_{s,t}} D(s,t)x^i_{s,t} \le \widetilde{c}_{u,v}$	capacity constraint
	$\sum_i x^i_{s,t} = 1$	flow conservation
variables	$0 \le x^i_{s,t} \le 1.$	forwarding probability

Network Entropy Maximization

- Assume we can enumerate all the paths from s to t, $P^i_{s,t}$. (only for analysis purpose)
- $x^i_{s,t}$: probability (fraction) of forwarding a packet of demand $D(s,t)$ to the i-th path ($P^i_{s,t}$)
- $z(x) = -x \log x$: Entropy function

Network Entropy Maximization (NEM)

maximize	$\sum_{s,t} D(s,t) \left(\sum_{P^i_{s,t}} z(x^i_{s,t}) \right)$	total entropy
subject to	$\sum_{s,t,i:(u,v)\in P^i_{s,t}} D(s,t) x^i_{s,t} \leq \widetilde{c}_{u,v}$	capacity constraint
	$\sum_i x^i_{s,t} = 1$	flow conservation
variables	$0 \leq x^i_{s,t} \leq 1.$	forwarding probability

NEM features

- NEM problem always has a global optimal solution.
  - Feasible solution: any optimal solution of COMMODITY problem
  - $z(x)$ is a concave function
  - Convex Optimization
- Solving directly is not efficient (Infinite path enumeration with cycles)

NEM features

- NEM problem always has a global optimal solution.
  - Feasible solution: any optimal solution of COMMODITY problem
  - $z(x)$ is a concave function
  - Convex Optimization
- Solving directly is not efficient (Infinite path enumeration with cycles)
- Solve dual problem (with E dual variables)

Optimal Solution of NEM

- Necessary Condition

$$\frac{x^i_{s,t}}{x^j_{s,t}} = \frac{e^{-\sum_{(u,v)} K^{(u,v)}_{P^i_{s,t}} \lambda_{u,v}}}{e^{-\sum_{(u,v)} K^{(u,v)}_{P^j_{s,t}} \lambda_{u,v}}}.$$

- $\lambda_{u,v}$: dual variable for necessary capacity constraint
- $K^{(u,v)}_{P^i_{s,t}}$: number of times $P^i_{s,t}$ passes through link (u,v)

Optimal Solution of NEM

- Necessary Condition

$$\frac{x^i_{s,t}}{x^j_{s,t}} = \frac{e^{-\sum_{(u,v)} K^{(u,v)}_{P^i_{s,t}} \lambda_{u,v}}}{e^{-\sum_{(u,v)} K^{(u,v)}_{P^j_{s,t}} \lambda_{u,v}}}.$$

- $\lambda_{u,v}$: dual variable for necessary capacity constraint
- $K^{(u,v)}_{P^i_{s,t}}$: number of times $P^i_{s,t}$ passes through link (u,v)

Penalizing Exponential Flow-spliTting (PEFT)

$$\text{PEFT:} \qquad x^i_{u,t} = \frac{e^{-p^i_{u,t}}}{\sum_j e^{-p^j_{u,t}}}.$$

- $p^i_{u,t}$: sum of $\lambda_{u,v}$ along the ith path

Algorithm for Optimizing Link Weights

Optimize Over Link Weights

```
Compute necessary capacities c̃ by solving COMMODITY problem
w ← Any set of link weights
f ← Traffic_Distribution(w)
while f ≠ c̃ do
    w ← Link_Weight_Update(f)
    f ← Traffic_Distribution(w)
end while
```

Algorithm for Optimizing Link Weights

Optimize Over Link Weights

Compute necessary capacities $\widetilde{\mathbf{c}}$ by solving COMMODITY problem
$\mathbf{w} \leftarrow$ Any set of link weights
$\mathbf{f} \leftarrow$ Traffic_Distribution($\mathbf{w}$)
while $\mathbf{f} \neq \widetilde{\mathbf{c}}$ **do**
    $\mathbf{w} \leftarrow$ Link_Weight_Update($\mathbf{f}$)
    $\mathbf{f} \leftarrow$ Traffic_Distribution($\mathbf{w}$)
end while

Link-Weight_Update(f)

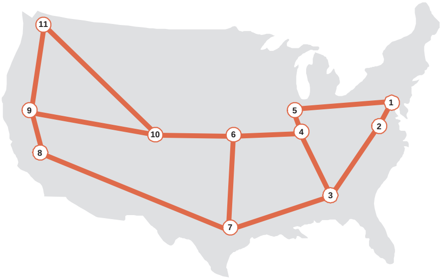
for each link (u, v) **do**
    $w_{u,v} \leftarrow w_{u,v} - \alpha\,(\widetilde{c}_{u,v} - f_{u,v})$
end for
Return new link weights $\mathbf{w}$

Outline

Traffic Engineering Schemes

- Optimal TE: Solve COMMODITY problem as a Linear Program (Tunnel-based)
- PEFT TE: Our algorithm (Link-weight-based)
- OSPF TE: Local search [Fortz-Thorup-2000] (Link-weight-based)

Network Topologies



Abilene Network

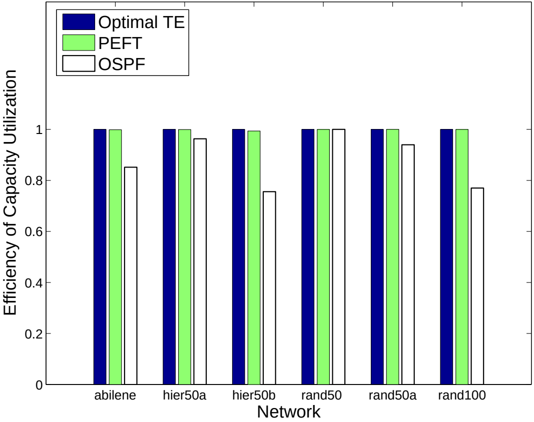
Name	Topology	Node #	Link #	Link Capacity
abilene	Backbone	11	28	10Gbps
hier50a	2-level	50	148	local access(200), long-haul (1000)
hier50b	2-level	50	212	local access(200), long-haul (1000)
rand50	Random	50	228	1000
rand50a	Random	50	245	1000
rand100	Random	100	403	1000

Traffic Matrices

- Abilene Network: measured data on Nov. 15th, 2005
- Other networks: same as [Fortz-Thorup-2000]
- 7 test cases for each network: uniformly decrease link capacity/increase demand

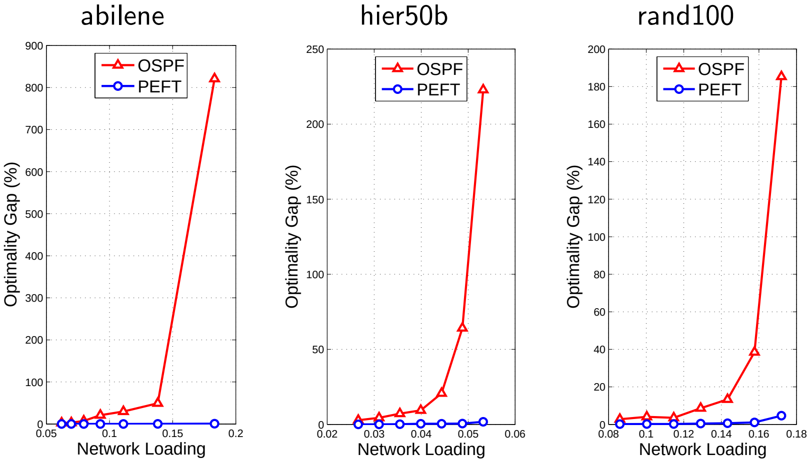
Minimize Maximum Link Utilization

- Efficiency of capacity utilization: Percentage of traffic demand satisfied when a link utilization reaches 100%.
- PEFT achieves optimal TE, and increases Internet capacity over OSPF by 15% for Abilene and 24% for Hier50b



Minimize Total Link Congestion Cost

- Optimality gap (compared against optimal TE)



Running Time

- TE with PEFT requires at most 2 minutes even for the largest network tested.
- The algorithm to find link weights for PEFT routing is 2000 times faster than local search algorithms (public version in TOTEM) for OSPF routing.

Outline

Conclusion

- Until now, Minimum-cost multicommodity flow can be realized by a link-state routing protocol (PEFT) from solving NEM.

Conclusion

- Until now, Minimum-cost multicommodity flow can be realized by a link-state routing protocol (PEFT) from solving NEM.
- Open Problems
  - Computational Complexity of NEM/PEFT: Polynomial?
  - Solve NEM/PEFT + COMMODITY problem altogether?
  - Whether DEFT [Xu-Chiang-Rexford, Infocom-07] can achieve optimal traffic engineering as well?
- More Information
  http://www.research.att.com/~dahaixu

Backup: Calculate Traffic Distribution for PEFT

- Random walk: A trajectory taking successive steps in random directions: Markov process
- Exponential Penalty on using cycles, e.g. $e^{-30} \approx 10^{-13}$