COPE: Traffic Engineering in Dynamic Networks

Hao Wang, Haiyong Xie, Lili Qiu,
Yang Richard Yang, Yin Zhang, Albert Greenberg

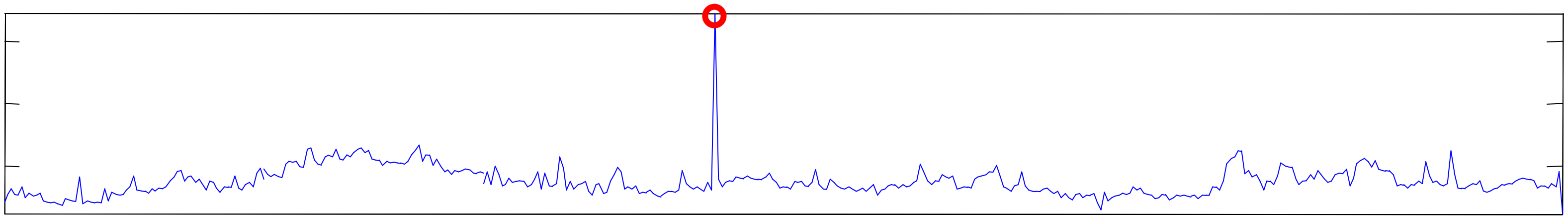
Yale University
UT Austin
AT&T Labs - Research

ACM SIGCOMM 2006

Traffic Engineering (TE)

- Objective
  - Adapting the routing of traffic to avoid congestion and make more efficient use of network resource
- Motivation
  - High cost of network assets & highly competitive nature of ISP market
  - Routing influences efficiency of network resource utilization
    - Latency, loss rate, congestion, …
- Two components
  - Understand traffic demands
  - Configure routing protocols
- This paper focuses on intra-domain TE
  - But the basic approach may also apply in inter-domain TE and network optimization in general

Challenge: Unpredictable Traffic



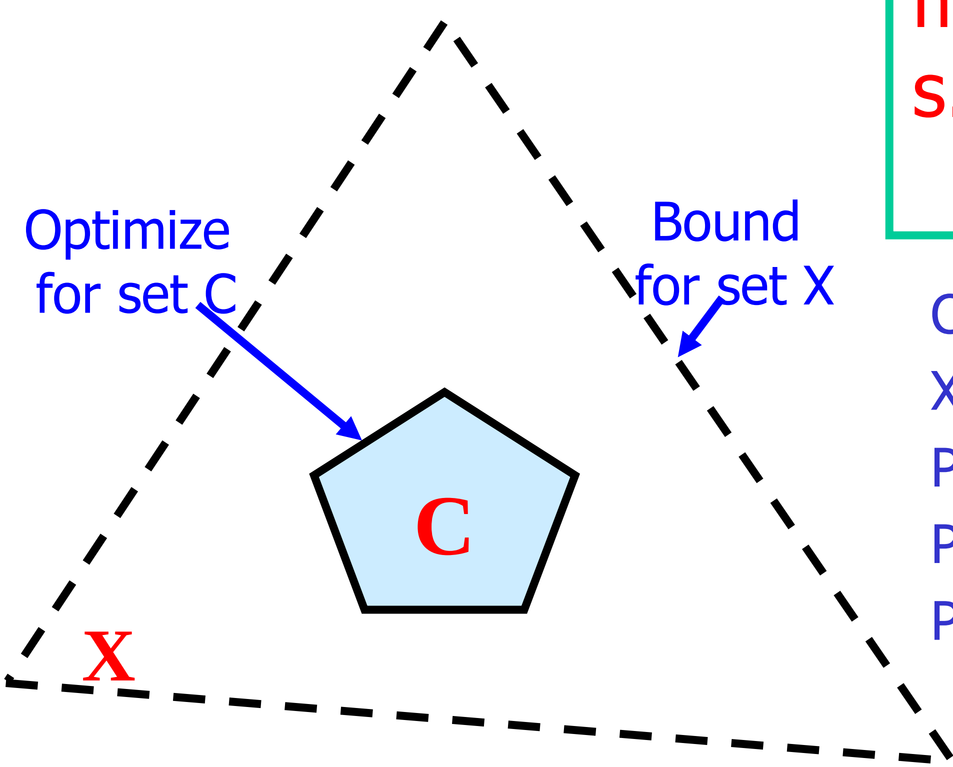
- Internet traffic is highly unpredictable!
  - Can be relatively stable most of the time …
  - However, usually contains spikes that ramp up extremely quickly
    - We identified sudden traffic spikes in the traces of several networks
  - Unpredictable traffic variations have been observed and studied by other researchers
    - [Teixeira et al. '04, Uhlig & Bonaventure '02, Xu et al. '05]
  - Confirmed by operators of several large networks via email survey
  - Abrupt traffic changes often occur when service is most valuable!
- Many possible causes for traffic unpredictability
  - Worms/viruses, DoS attacks, flash crowds, BGP routing changes [Teixeira et al. '05, Agarwal et al. '05], failures in other networks, load balancing by multihomed customers, TE by peers …
- TE needs to handle unpredictable traffic
  - Otherwise, links and/or routers may get unnecessarily overloaded
    - Long delay, high loss, reduced throughput, violation of SLA
  - Customers can remember bad experiences really well …

Existing TE Solutions

- Prediction-based TE
  - Examples:
    - Off-line:
      - Single predicted TM [Sharad et al. '05]
      - Multiple predicted TMs [Zhang et al. '05]
    - On-line: MATE [Elwalid et al. '01] & TeXCP [Kandula et al. '05]
  - Pro: Works great when traffic is predictable
  - Con: May pay a high penalty when real traffic deviates substantially from the prediction
- Oblivious routing
  - Examples:
    - Oblivious routing [Racke '02, Azar et al. '03, Applegate et al. '03]
    - Valiant load-balancing [Kodialam et al. '05, Zhang & McKeown '04]
  - Pro: Provides worst-case performance bounds
  - Con: May be sub-optimal for normal traffic
    - The optimal oblivious ratio of several real network topologies studied in [Applegate et al '03] is ~2

Our Approach: COPE

Common-case Optimization with Penalty Envelope



$\min_f \max_{d \in C} P_C(f, d)$
s.t. (1) f is a routing
(2) $\forall x \in X: P_X(f, x) \leq PE$

- C: common-case (predicted) TMs
- X: all TMs of interest
- $P_C(f,d)$: common-case penalty function
- $P_X(f,x)$: worst-case penalty function
- PE: penalty envelope

Model

- Network topology: graph $G = (V,E)$
  - V: set of routers
  - E: set of network links
- Traffic matrices (TMs)
  - A TM is a set of demands: $d = \{ d_{ab} \mid a,b \in V \}$
  - d_{ab}: traffic demand from a to b
  - Can extend to point-to-multipoint demands
- MPLS-style, link-based routing
  - $f = \{ f_{ab}(i,j) \mid a,b \in V, (i,j) \in E \}$
  - $f_{ab}(i,j)$: the fraction of demand from a to b (i.e., d_{ab}) that is routed through link (i,j)
  - Paper includes ideas on how to approximate OSPF-style (i.e., shortest path implementable) routing

Routing Performance Metrics

- Maximum Link Utilization (MLU):

$$U(f,d) = \max_{(i,j)\in E} \sum_{a,b\in V} d_{ab} f_{ab}(i,j) / c_{ij}$$

- Optimal Utilization

$$OU(d) = \min_{f \text{ is a routing}} U(f,d)$$

- Performance Ratio

$$PR(f,d) = \frac{U(f,d)}{OU(d)}$$

COPE Instantiation

$\min_f \max_{d \in C} P_C(f, d)$
s.t. (1) f is a routing; and (2) $\forall x \in X$: $P_C(f, x) \leq$ PE

C: convex hull of multiple past TMs
- A linear predictor predicts the next TM as a convex combination of past TMs (e.g., EWMA)
- Aggregation of all possible linear predictors → the convex hull

X: all possible non-negative TMs
- Can add access capacity constraints or use a bigger convex hull

$P_C(f,d)$: penalty function for common cases
- maximum link utilization: $U(f,d)$
- performance ratio: $PR(f,d)$

$P_X(f,x)$: penalty function for worst cases
- performance ratio: $PR(f,x)$

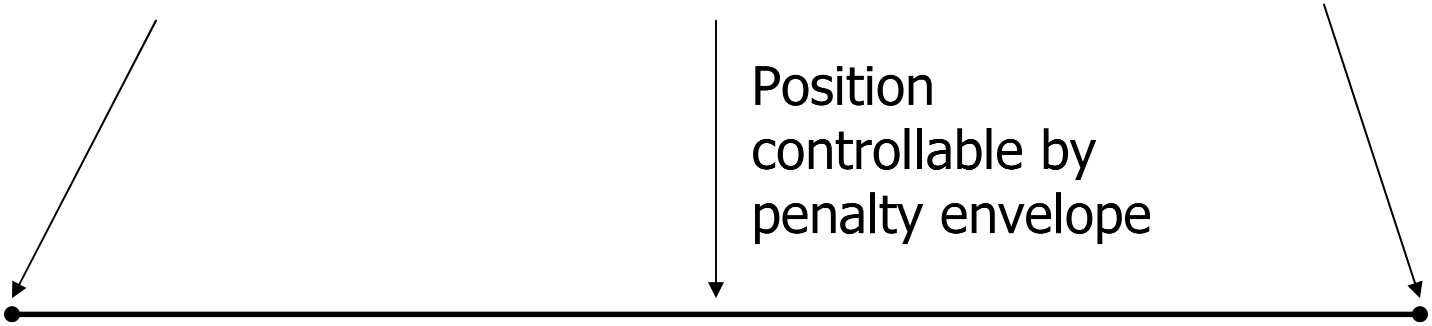
PE: penalty envelope
- $PE = \beta \min_f \max_{x \in X} P_X(f,x)$
- $\beta \geq 1$ controls the size of PE w.r.t. the optimal worst-case penalty
  - $\beta=1$ → oblivious routing
  - $\beta=\infty$ → prediction-based TE

Current COPE Implementation

1. Collect TMs continuously
2. Compute COPE routing for the next day by solving a linear program (LP)
   - Common-case optimization
     - Common case: convex hull of multiple past TMs
       - All TMs in previous day + same/previous days in last week
     - Minimize either MLU or PR over the convex hull
   - Penalty envelope
     - Bounded PR over all possible nonnegative TMs
   - See paper for details of our LP formulation
3. Install COPE routing
   - Currently done once per day → an off-line solution
   - Can be made on-line (e.g., recompute routing upon detection of significant changes in TM)

COPE Illustrated

Best common-case + Poor worst-case	Good common-case + Bounded worst-case	Poor common-case + Best worst-case
Prediction-based TE	**COPE**	**Oblivious Routing**



Spectrum of TE with unpredictable traffic

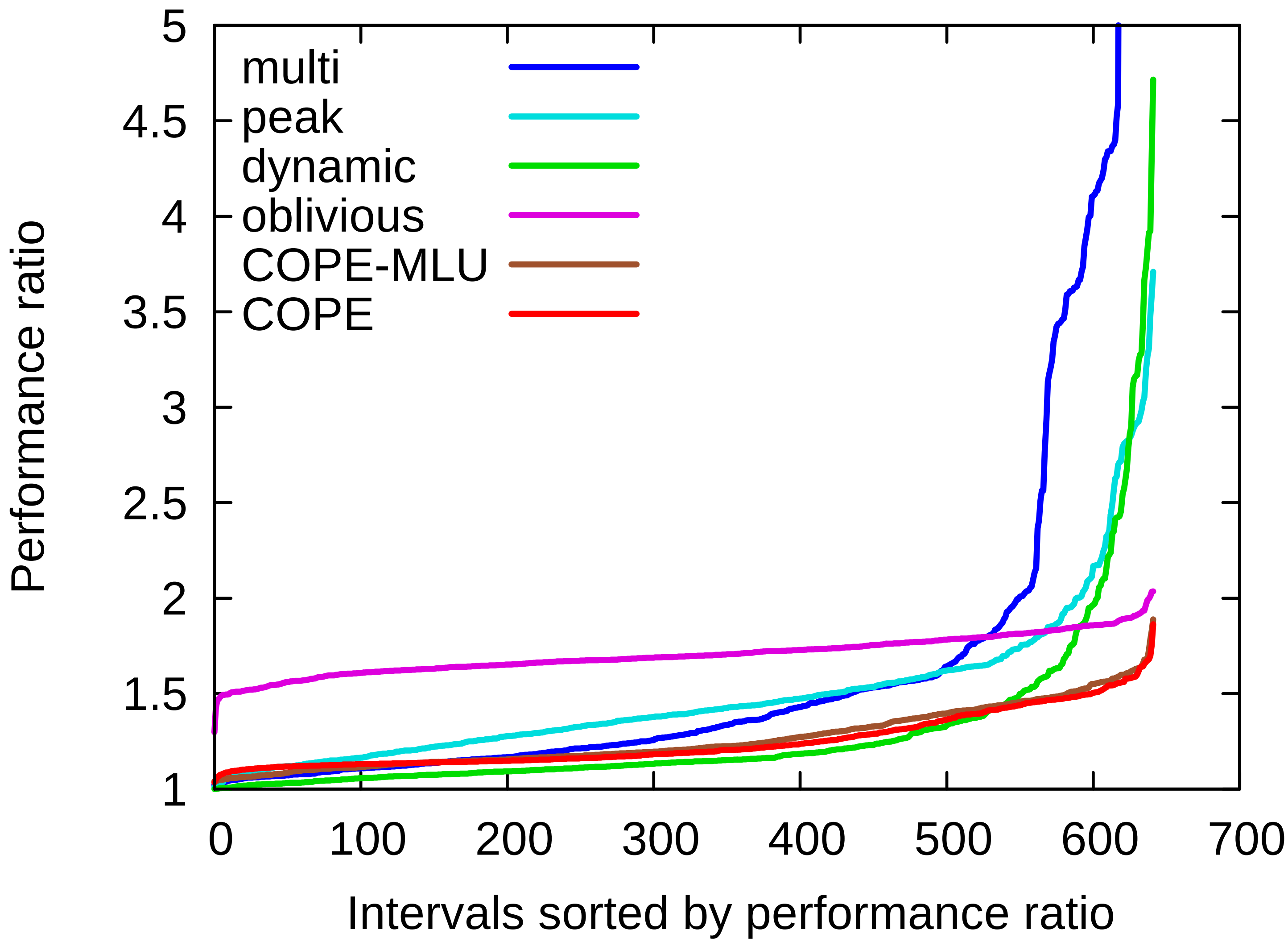
There are enough unexpected cases → Penalty envelope is required
The worst unexpected case too unlikely to occur →
Too wasteful to "optimize" for the worst-case (at the cost of poor common-case performance)

Evaluation Methodology

- TE Algorithms
  - COPE: COPE with $P_C(f,d) = PR(f,d)$ (i.e. performance ratio)
  - COPE-MLU: COPE with $P_C(f,d) = U(f,d)$ (i.e. max link utilization)
  - Oblivious routing: $\min_f \max_x PR(f,x)$ ($\approx$ COPE with $\beta=1$)
  - Dynamic: optimize routing for TM in previous interval
  - Peak: peak interval of previous day + prev/same days in last week
  - Multi: all intervals in previous day + prev/same days in last week
  - Optimal: requires an oracle

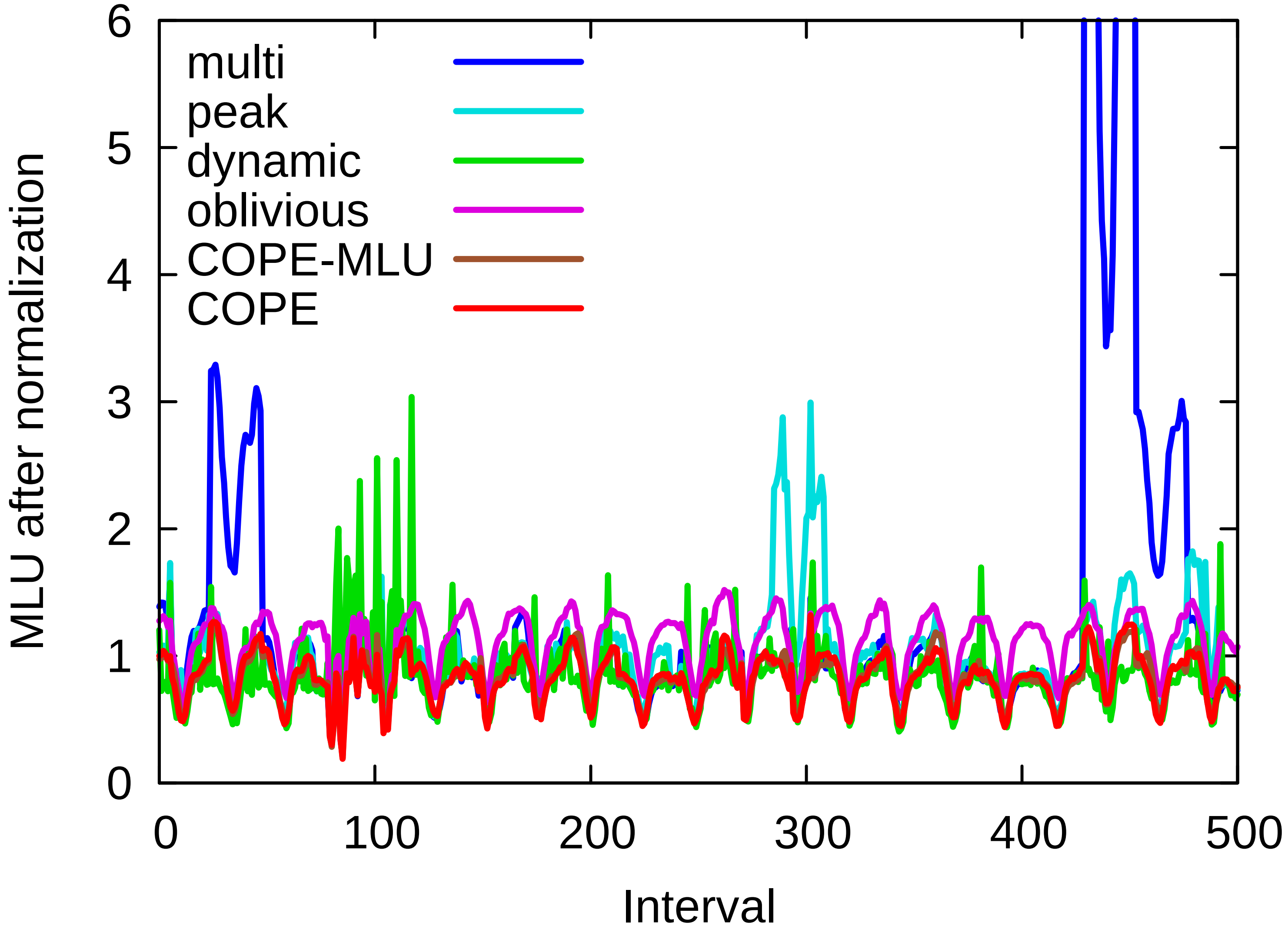
- Dataset
  - US-ISP
    - hourly PoP-level TMs for a tier-1 ISP (1 month in 2005)
    - Optimal oblivious ratio: 2.045; default penalty envelope: 2.5
  - Abilene
    - 5-min router-level TMs on Abilene (6 months: Mar – Sep. 2004)
    - Optimal oblivious ratio: 1.853; default penalty envelope: 2.0

US-ISP: Performance Ratio



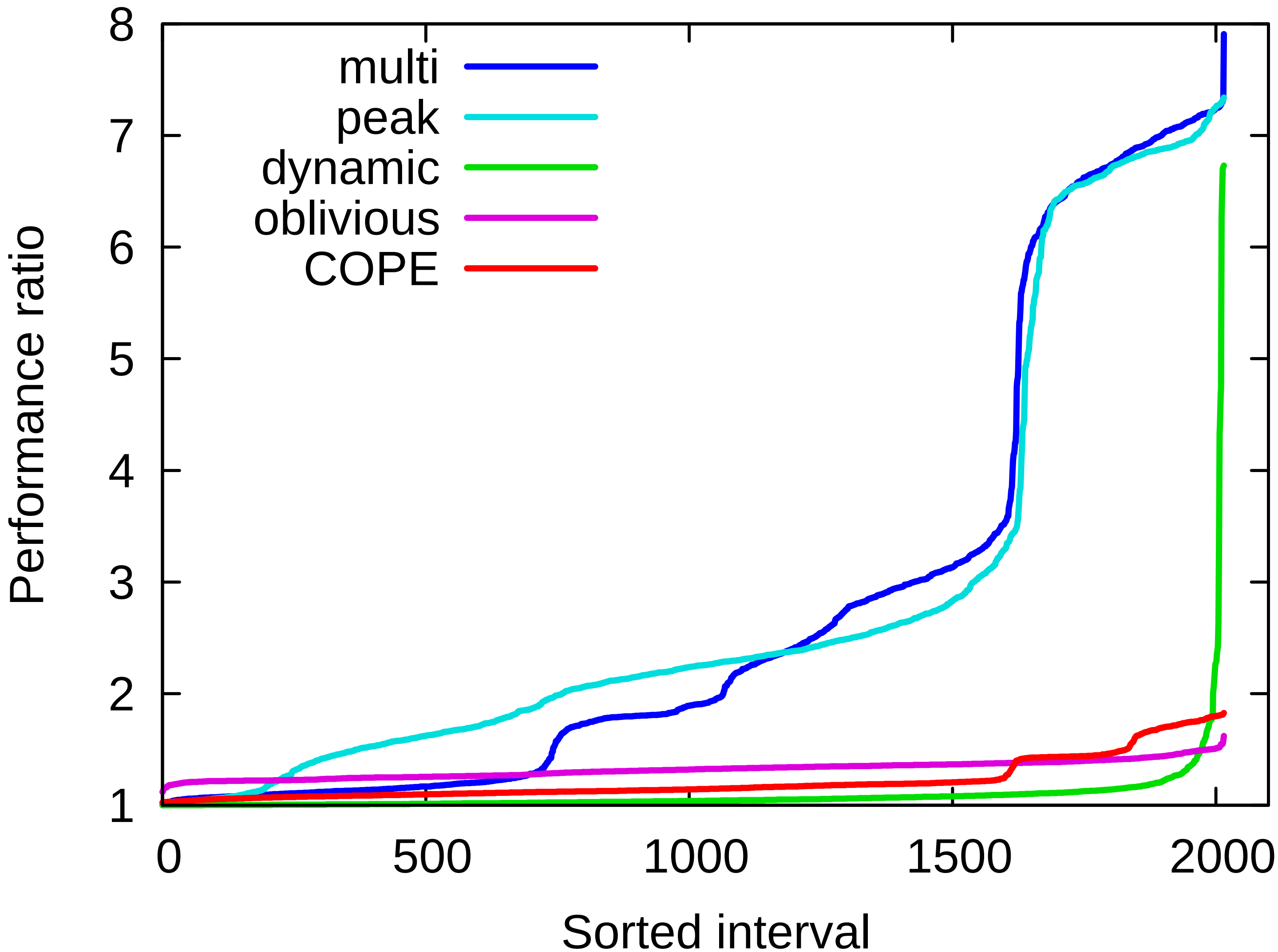
Common cases: COPE is close to optimal/dynamic and much better than others
Unexpected cases: COPE beats even OR and is much better than others

US-ISP: Maximum Link Utilization



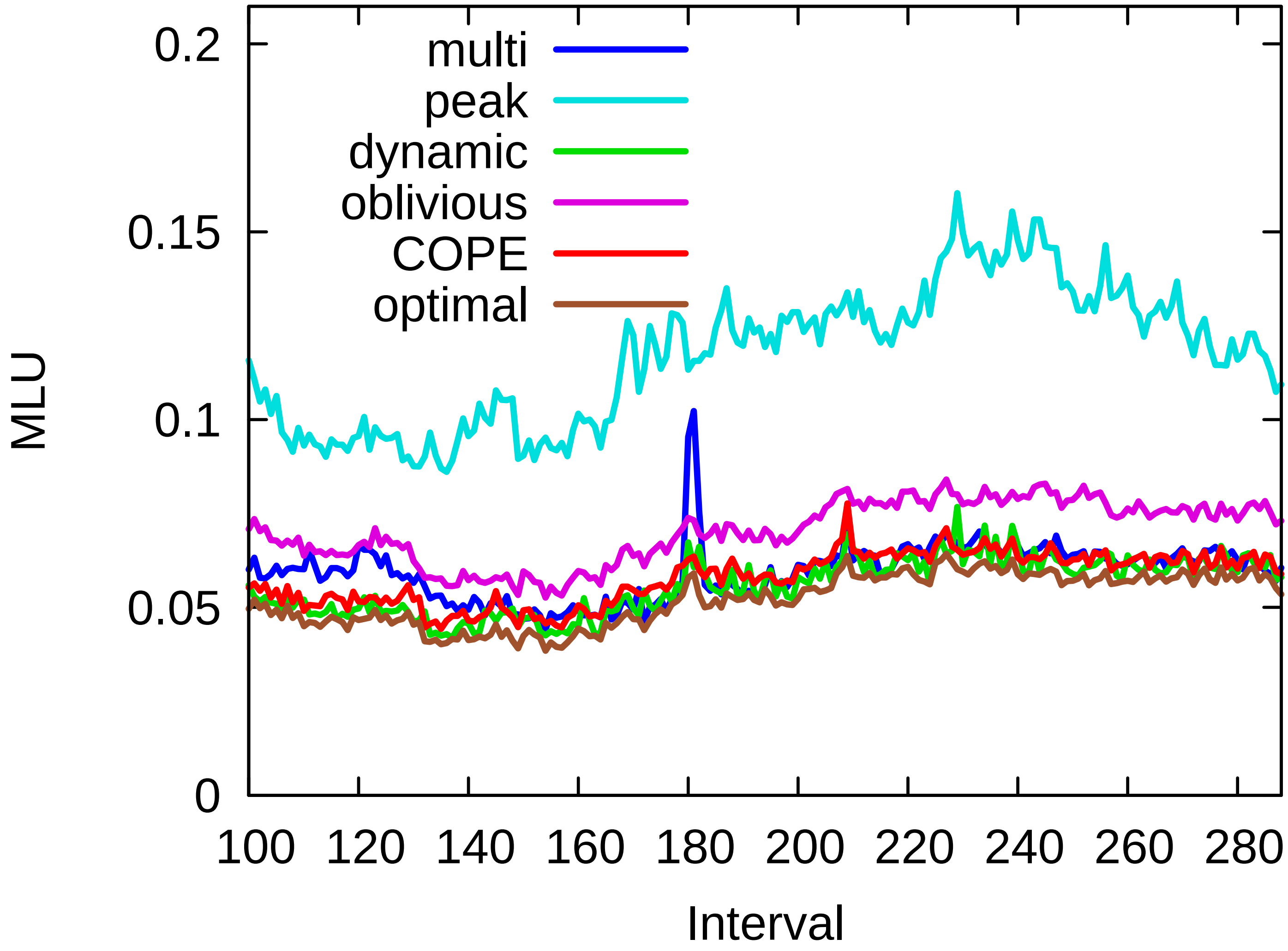
Common cases: COPE is close to optimal/dynamic and much better than others
Unexpected cases: COPE beats even OR and is much better than others

Abilene: Performance Ratio



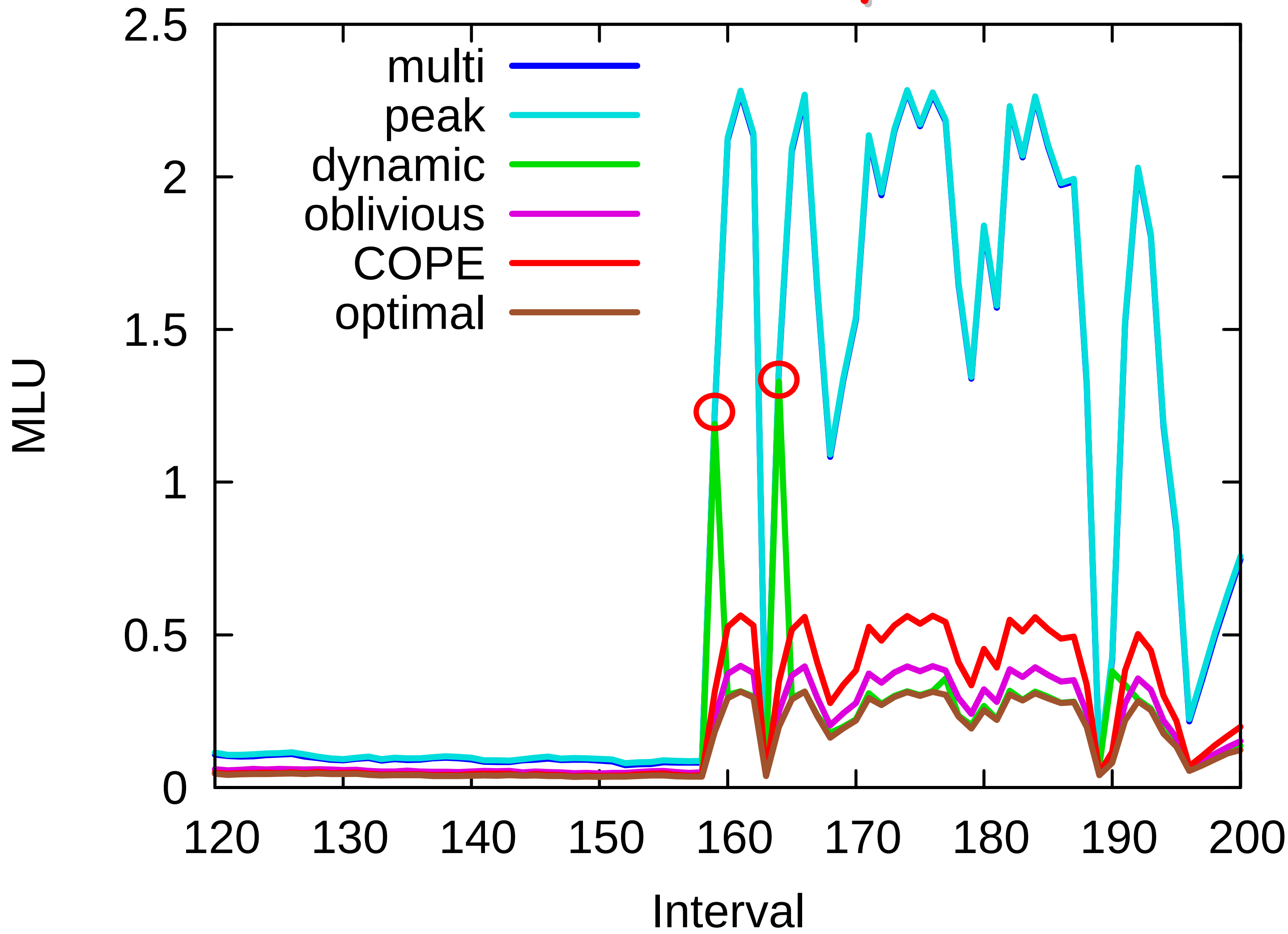
Common cases: COPE is close to optimal/dynamic and much better than others
Unexpected cases: COPE is close to OR and much better than others

Abilene: MLU in Common Cases



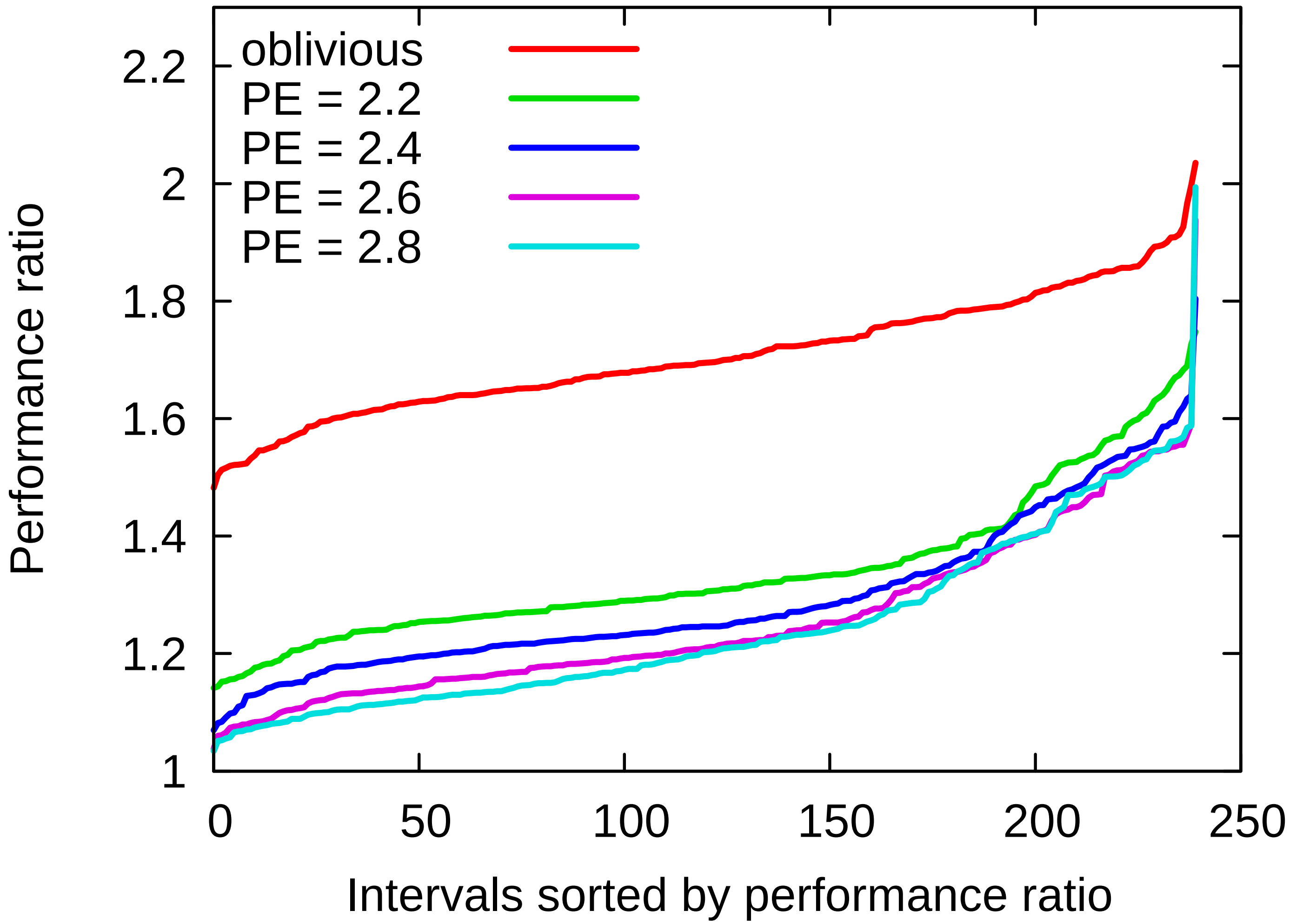
Common cases: COPE is close to optimal/dynamic and much better than others

Abilene: MLU in Unexpected Cases



Unexpected cases: COPE is close to OR and much better than others

US-ISP: Sensitivity to PE

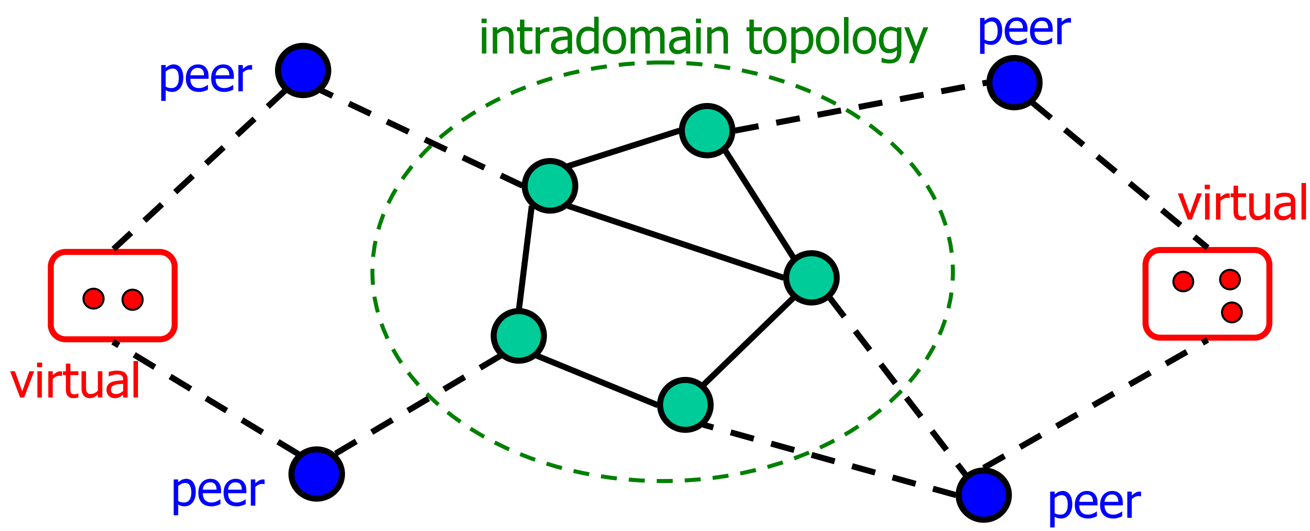


COPE is insensitive to PE; even a small margin in PE can significantly improve the common-case performance

COPE with Interdomain Routing

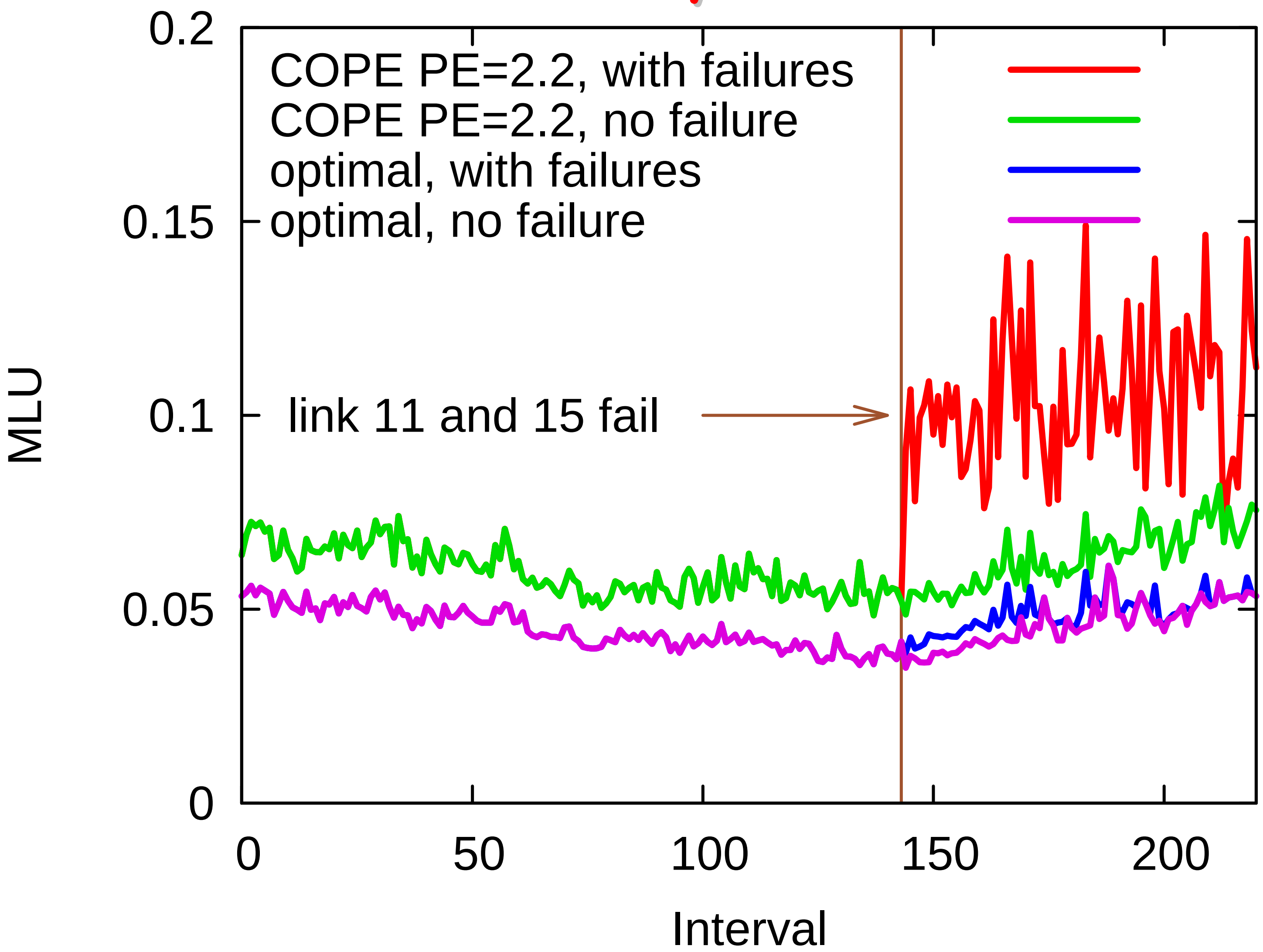
- Motivation
  - Changes in availability of interdomain routes can cause significant shifts of traffic within the domain
    - E.g. when a peering link fails, all traffic through that link is rerouted
- Challenges
  - Point-to-multipoint demands → need to find splitting ratios among exit points
  - The set of exit points may change → topology itself is dynamic
  - Too many prefixes → cannot enumerate all possible exit point changes

COPE with Interdomain Routing: A Two-Step Approach



1. Apply COPE on an extended topology to derive good splitting ratios
   - Group dest prefixes with same set of exit points into a virtual node
   - Derive pseudo demands destined to each virtual node by merging demands to prefixes that belong to this virtual node
   - Connect virtual node to corresponding peer using virtual link with infinite BW
   - Compute extended topology *G'* as
     G' = intradomain topology + peers + peering links + virtual nodes + virtual links
   - Apply COPE to compute routing on *G'* for the pseudo demands
   - Derive splitting ratios based on the routes
2. Apply COPE on point-to-point demands to compute intradomain routing
   - Use the splitting ratios obtained in Step 1 to map point-to-multipoint demands into point-to-point demands

Preliminary Evaluation



COPE can significantly limit the impact of peering link failures

Conclusions & Future Work

- COPE =
  Common-case Optimization with Penalty Envelope

- COPE works!
  - Common cases: close to optimal; much better than oblivious routing and prediction-based TE with comparable overhead
  - Unexpected cases: much better than prediction-based TE, and sometimes may beat oblivious routing
  - COPE is insensitive to the size of the penalty envelope; even a small margin in PE improves common-case performance a lot
  - COPE can be extended to cope with interdomain routes

- Lots of ongoing & future work
  - Efficient implementation of COPE
  - COPE with MPLS and VPN
  - COPE with OSPF
  - COPE with online TE
  - COPE for other network optimization problems

Thank you!