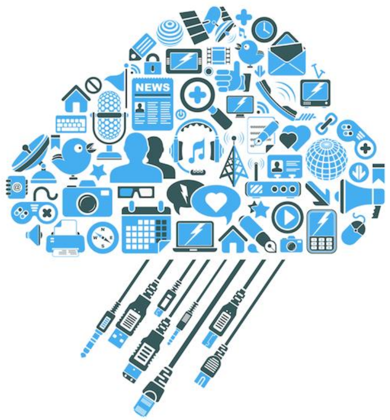


# Rethinking (Predictive) WAN Traffic Engineering

**Yarin Perry**[1], Felipe Vieira Frujeri[2], Chaim Hoch[1], Srikanth Kandula[2], Ishai Menache[2], Michael Schapira[1], Aviv Tamar[3]

1 THE HEBREW UNIVERSITY OF JERUSALEM

2 Microsoft Research

3 TECHNION Israel Institute of Technology

# Traffic Engineering (TE) in Wide-Area Networks (WANs)

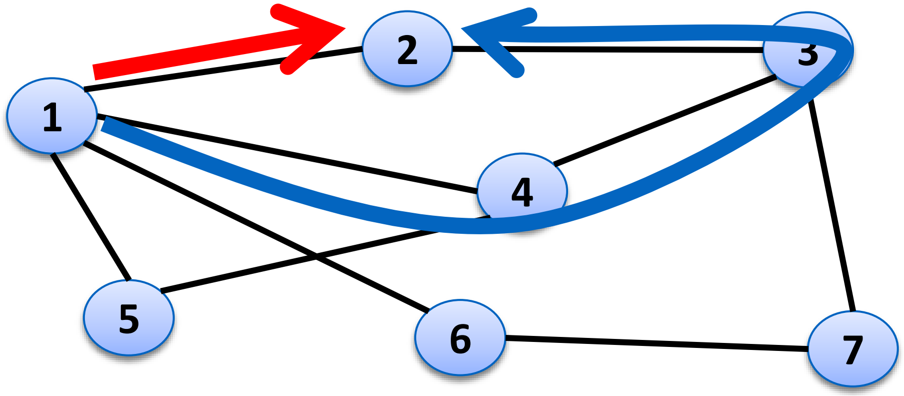
[SWAN, SIGCOMM 2013] [B4, SIGCOMM 2013]

# Classic Traffic Engineering Model

| | 1 | 2 | 3 | … | N |
|---|---|---|---|---|---|
| 1 | 0 | 100 | 3 | … | 13 |
| 2 | 142 | 0 | 5 | … | 0 |
| 3 | 20 | 0 | 0 | … | 32 |
| … | … | … | … | 0 | … |
| N | 12 | 0 | 50 | … | 0 |

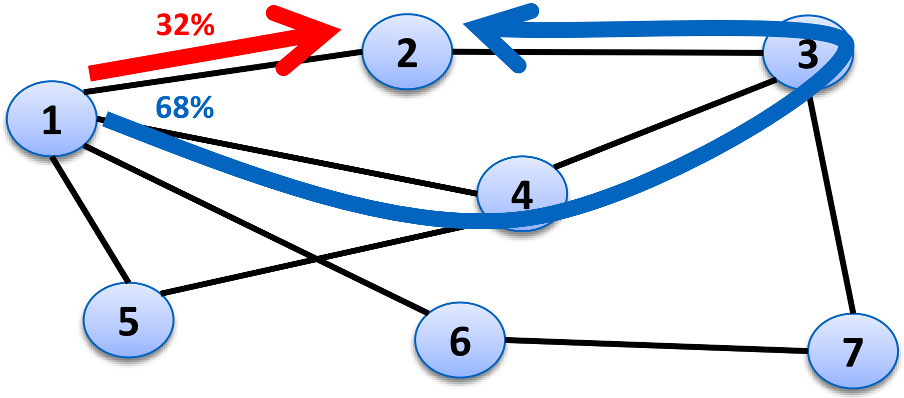
**demand matrix**

+




**network topology + tunnels**

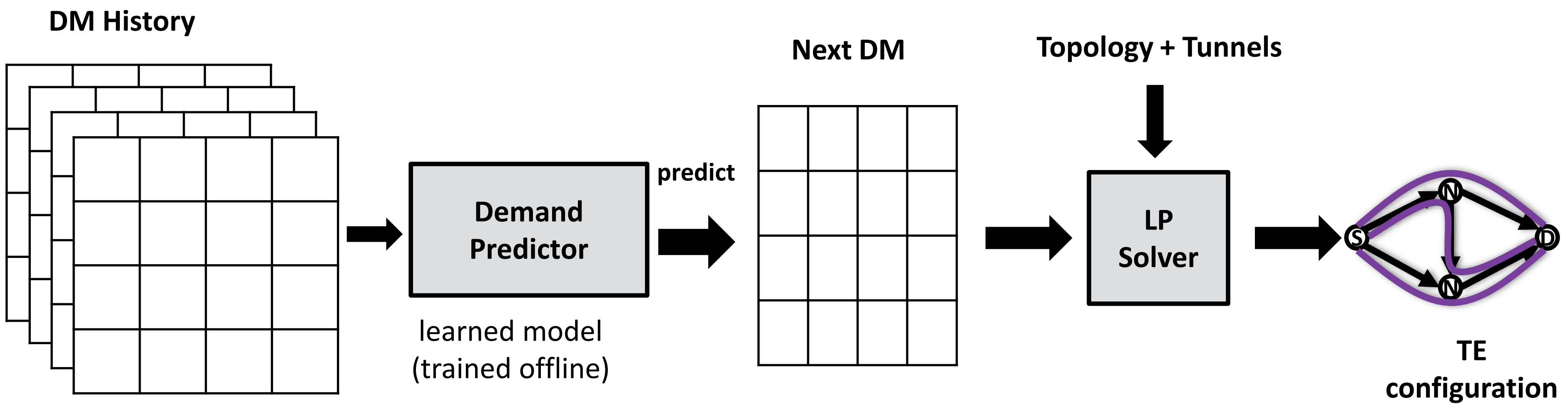
**flow optimization wrt a global objective (via linear programming)**



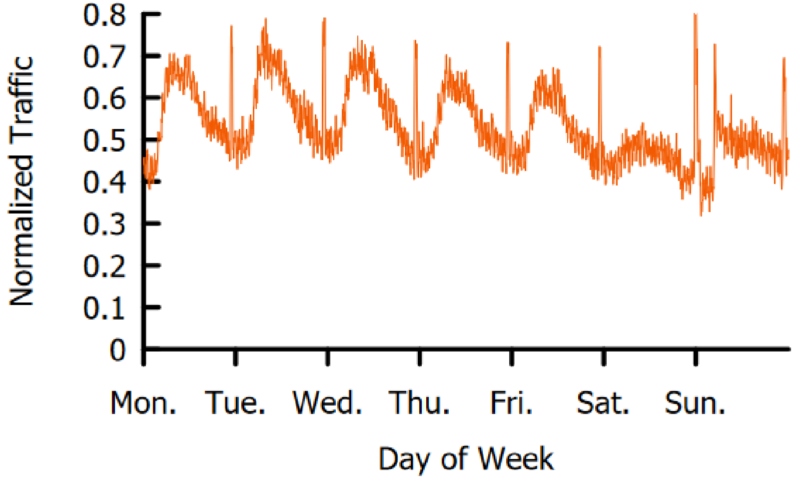

**TE configuration**

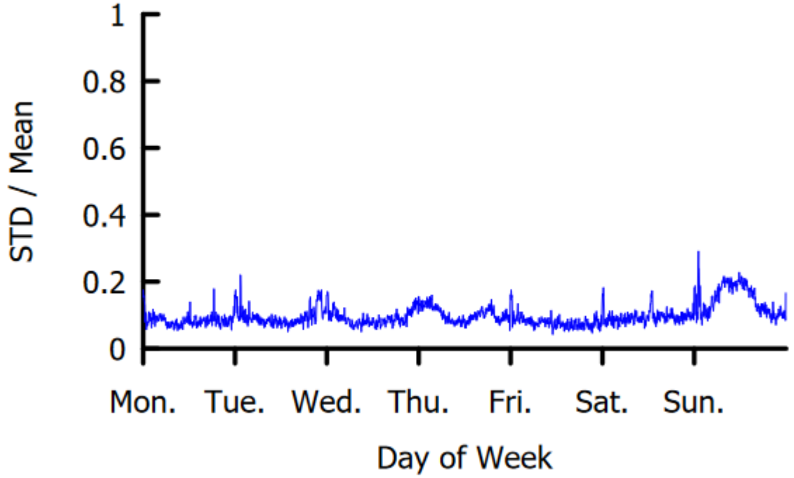
# Wait, how can we know the upcoming demands? Predict them!

[SWAN, SIGCOMM 2013] [B4, SIGCOMM 2013]



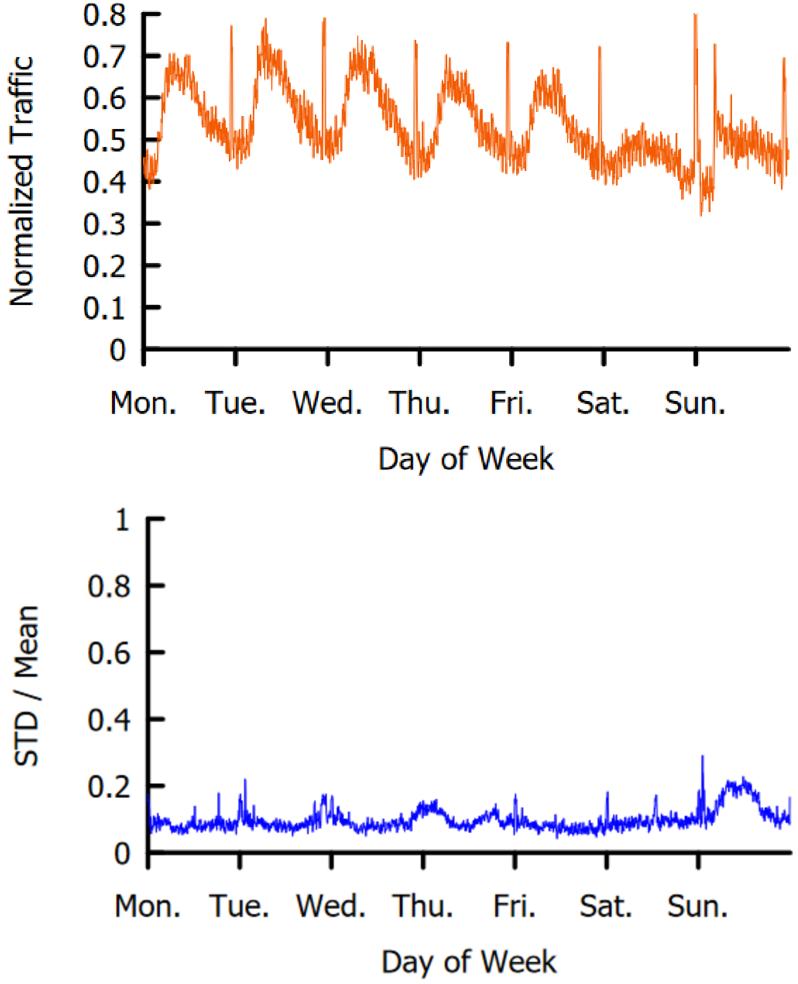
# A Tale of Two Traffic Patterns



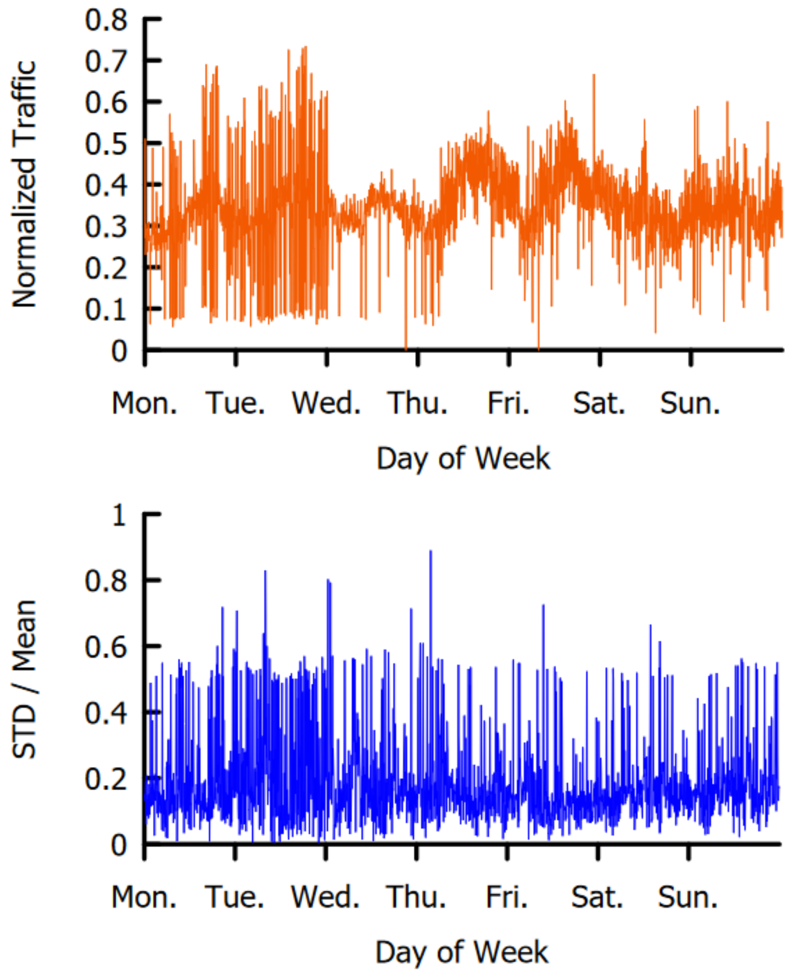




**predictable**
inter-datacenter
traffic

# A Tale of Two Traffic Patterns

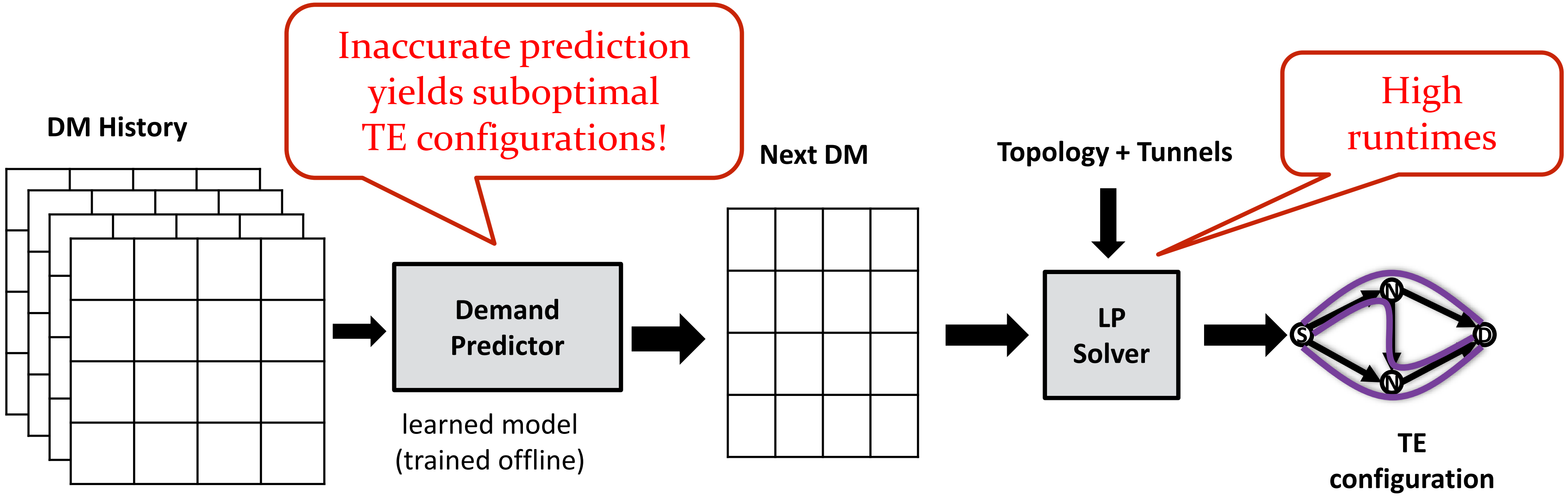



**predictable**
inter-datacenter
traffic

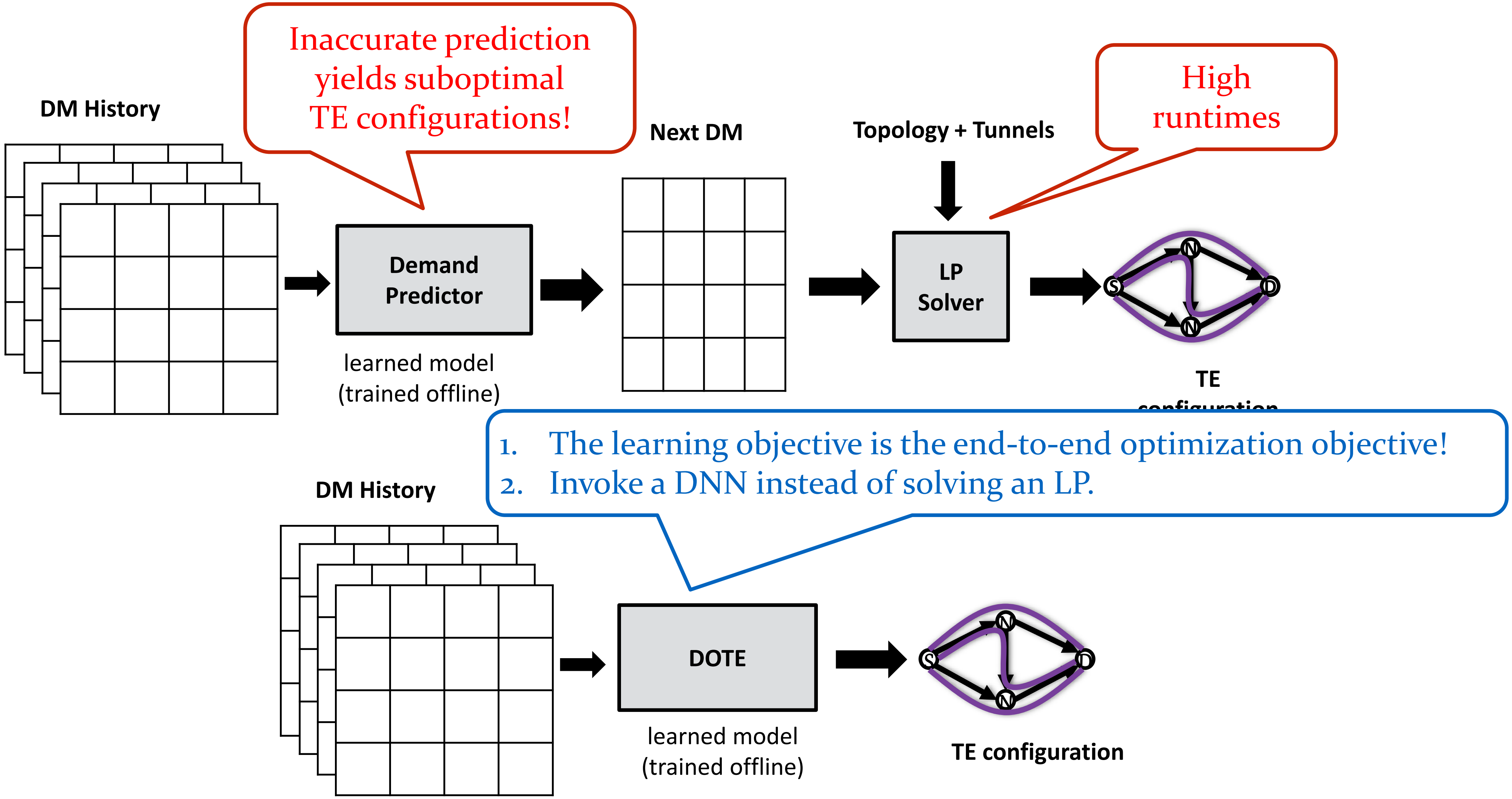



**unpredictable**
customer-facing
traffic

# Drawbacks of Demand-Prediction-Based TE

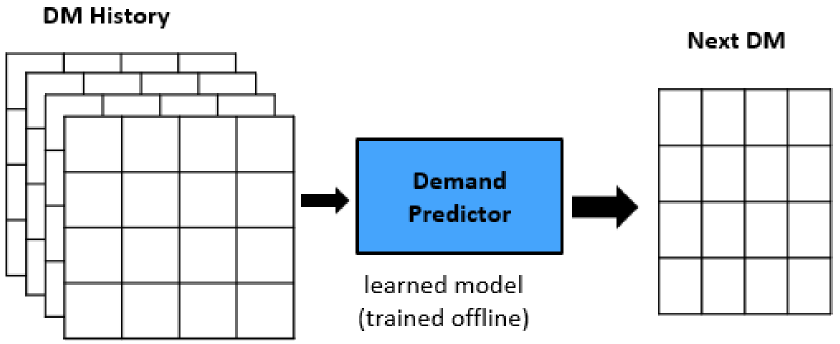


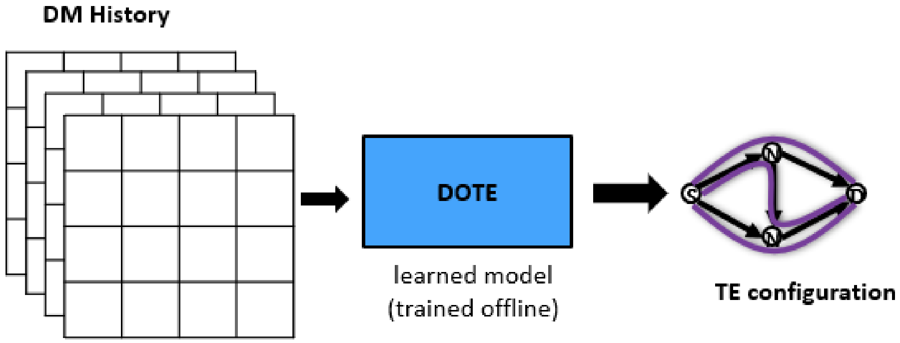
# DOTE: Direct Optimization for Traffic Engineering



# DOTE's Offline Training

| | Training data | Performance metric |
|---|---|---|
| Demand Prediction | Empirically observed sequences of past DMs | Prediction loss (e.g., RMSE) |
| DOTE | | The end-to-end TE objective (maximize total flow, minimize MLU, etc.) |



# DOTE's Offline Training

- Training is a **stochastic gradient descent (SGD)** process:
  - Uniformly **sample** m sequences of k+1 DMs $(D_t, \dots, D_{t-k})$ from an empirical dataset of past realized DMs.
  - **Update** the DNN's parameters (link weights):

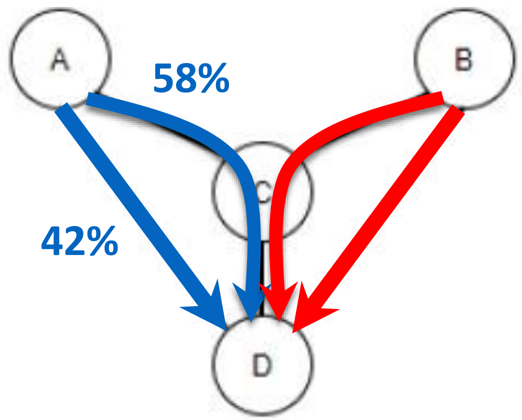
$$\theta = \theta - \underbrace{\alpha}_{\text{step size}} \frac{1}{m} \sum_{i \in \text{sample}} \nabla_\theta \underbrace{L}_{\text{TE objective}}\big(D_t^i, \underbrace{\pi_\theta\left(D_{t-1}^i, \dots, D_{t-k}^i\right)}_{\text{DNN output (splitting ratios)}}\big)$$

- Our realization is simple, efficient, and seems broadly applicable in TE.

# Intuition for DOTE's offline training

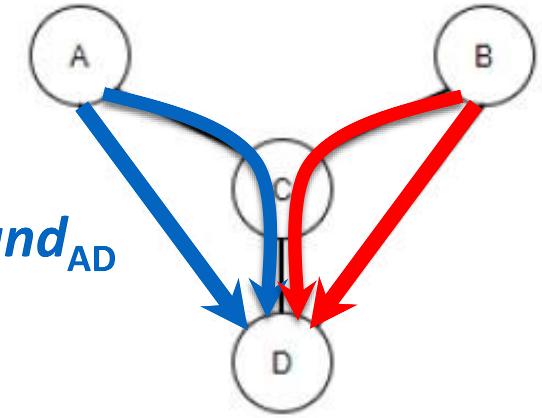
- Each of nodes A and B can send traffic to node D via its direct link or through C. All link capacities are 1.

- At the beginning of each time epoch, traffic splitting ratios must be determined for each source-destination pair.



# Intuition for DOTE's offline training

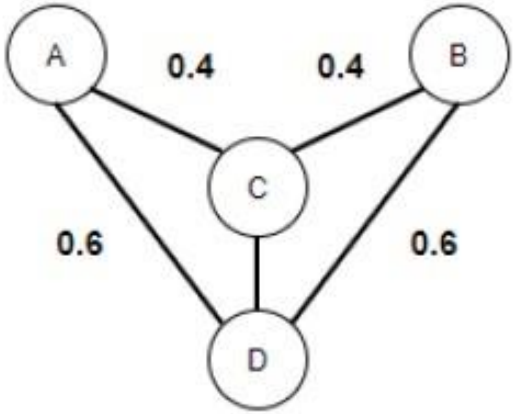
- Suppose A and B's demands are drawn (i.i.d) from a fixed probability distribution: $\left(\frac{5}{3}, \frac{5}{6}\right)$ with probability ½, $\left(\frac{5}{6}, \frac{5}{3}\right)$ with probability ½
- **The TE system has no a priori knowledge of this distribution!**
- <u>**Goal**</u>: minimize the maximum-link-utilization (MLU), i.e., $\max_e \frac{f_e}{c_e}$

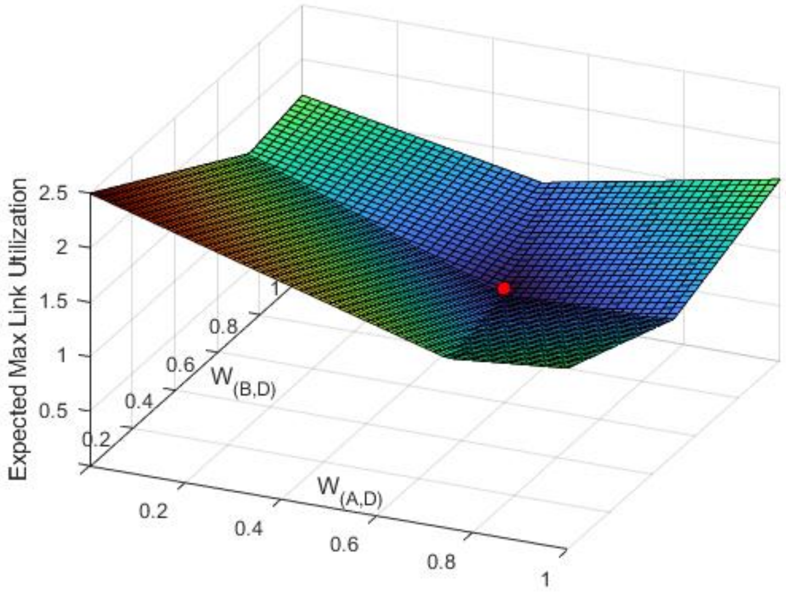



$f_{AD} = w_{AD} \times \textit{demand}_{AD}$

# Intuition for DOTE's offline training

- **<u>Observation</u>**: the *<u>expected</u>* maximum-link-utilization (MLU) is convex in the splitting ratios!
- **Gradient descent reaches the optimum** (no explicit demand prediction required)



# Intuition for DOTE's offline training

- **But wait, how can we estimate the <u>expected</u> MLU gradient?**
  - The system does not know the distribution over demands!
  - Hence, the expected MLU function is also not known

- **<u>Observation</u>**: Can compute MLU gradient for any past demand realization

- Averaging over gradients → approximates **expected** MLU gradient

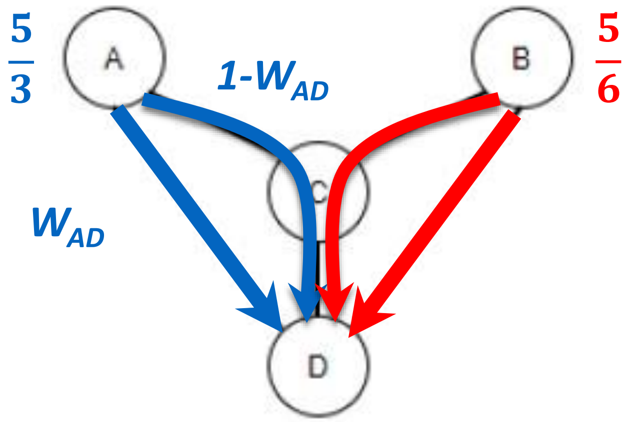
# Intuition for DOTE's offline training

- Consider a **specific** demand realization: ***demand***$_{AD}$ **=** $\frac{5}{3}$ and ***demand***$_{BD}$ **=** $\frac{5}{6}$

- The MLU as a function of the splitting ratios **for this demand realization can be expressed in closed form**:

$$\max\{\frac{5}{3}W_{AD}, \frac{5}{3}(1-W_{AD})+\frac{5}{6}(1-W_{BD}), \frac{5}{6}W_{BD}\}$$

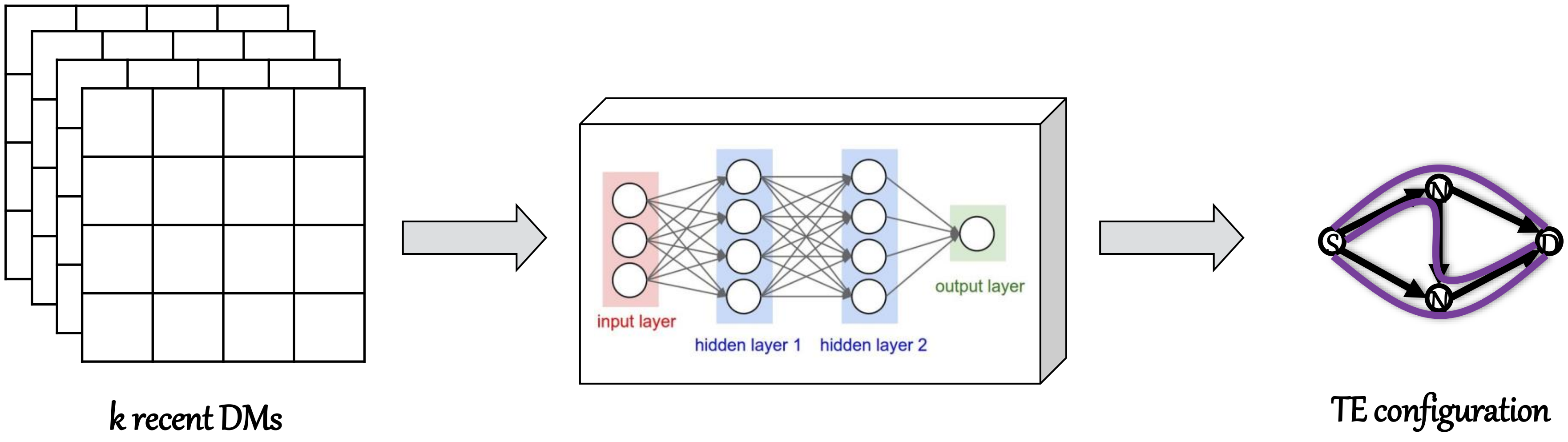
- The (sub)gradient of the MLU **for this demand realization** can thus be computed



# Generalizing from the toy example

- DOTE extends to **arbitrary network topologies, tunnel choices,** and **distributions over traffic demands.**

- DOTE addresses **temporal patterns in traffic** by harnessing the power of deep learning.
    - Gradient descent is now used to **optimize the DNN link weights**



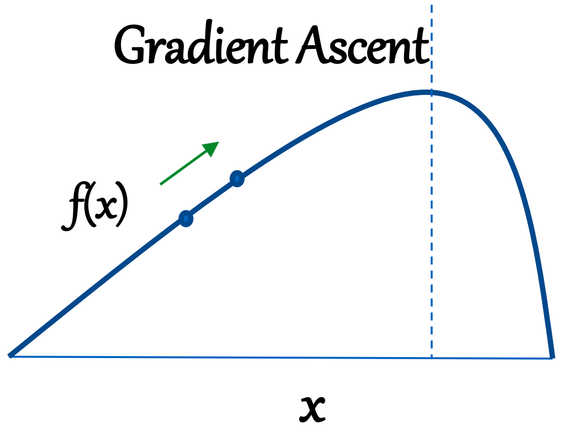
# Extending DOTE to other optimization objectives

- What about optimization objectives other than MLU?
  - Maximizing total flow
  - Maximizing concurrent flow

- DOTE's output now also specifies a "**rate cap**" for each communicating pair.

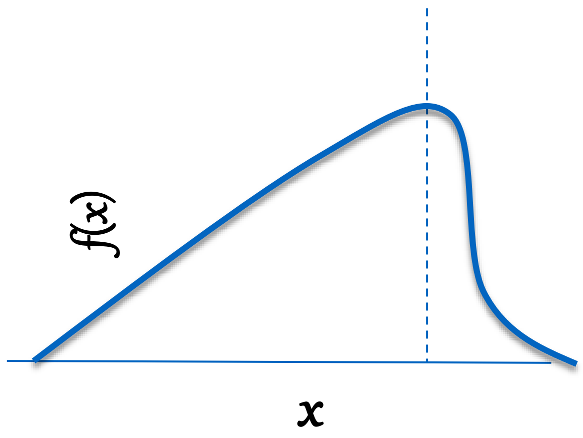
- By **normalizing the rate caps**, DOTE avoids violating link capacities.

- <u>**Key challenge**</u>: the induced function is <u>**not**</u> concave!



# Extending DOTE to other optimization objectives

- We prove that **<u>for any specific demand realization</u>**, the resulting TE performance function is **<u>quasiconcave</u>**.
    - (Normalized) gradient ascent reaches the optimum [Nesterov 84]




**A quasiconcave, but not convace, function**

# Extending DOTE to other optimization objectives

- We prove that **<u>for any specific demand realization</u>**, the resulting TE performance function is **<u>quasiconcave</u>**.
  - (Normalized) gradient descent reaches the optimum [Nesterov 84]
- We prove that **quasiconcavity also holds when <u>averaging across demand realizations</u>**.
  - The sum of quasiconcave functions need not be quasiconcave!
- This implies that (normalized) **<u>stochastic</u> gradient descent also converges to the optimum** [Hazan – Levy – Shalev-Shwartz, NeurIPS 2015]

# Empirical Evaluation of DOTE

- We extensively evaluate DOTE using empirical data.
- Our empirical evaluation spans
  - different network topologies (10s-100s of nodes)
  - **O($10^4$) production demand matrices** (Abilene, GEANT, 2 MSFT WANs)
  - several tunneling schemes (shortest-paths, edge-disjoint, SMORE)
  - **different flow optimization objectives**

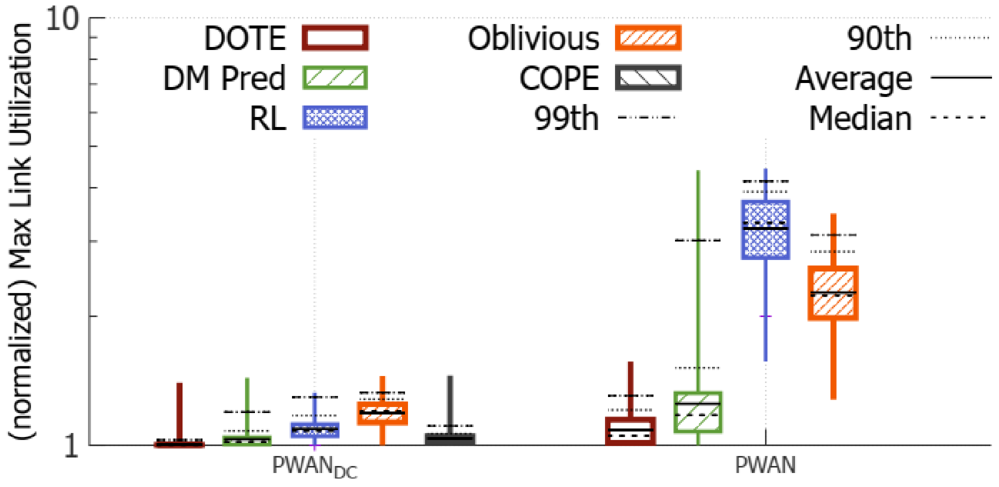
    (maximizing flow, maximizing concurrent-flow, minimizing MLU)

# Compare with …

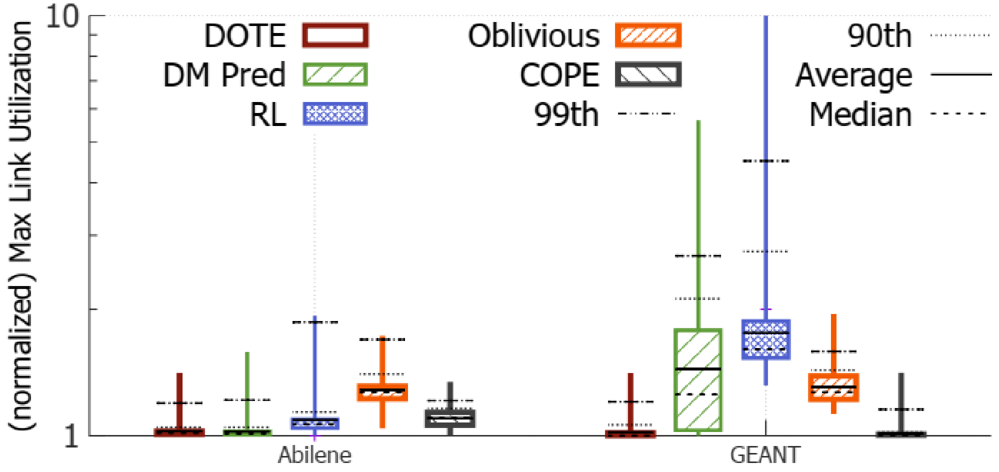
- **Demand prediction**
  [Hong et al., SIGCOMM 13] [Jain et al., SIGCOMM 13] [Kumar et al., NSDI 18] [Kumar et al., SOSR 18] …
- **Oblivious routing** [Appelgate-Cohen., SIGCOMM 03]
- **Hybrid approaches:**
  **COPE** [Wang et al., SIGCOMM 06], **SMORE** [Kumar et al., NSDI 18]
- **Reinforcement learning** [Valadarsky et al., HotNets 17]
- **Omniscient oracle** with perfect knowledge of future demands

# Minimizing Maximum Link Utilization

| | #Nodes | #Edges | Length | Granularity |
|---|---|---|---|---|
| **Abilene** | 11 | 14 | 4.5months | 5 min. |
| **GEANT** | 23 | 37 | 4 months | 15 min. |
| ***PWAN*** | O(100) | O(100) | O(1) months | minutes |
| ***$PWAN_{DC}$*** | O(10) | O(10) | O(1) months | minutes |



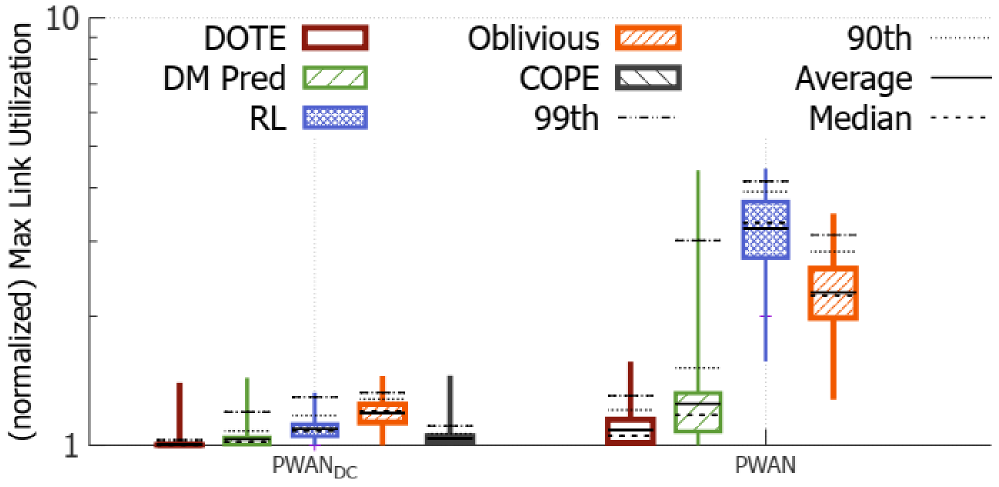

- DOTE **closely approximates** the (infeasible) omniscient oracle
- Bigger improvement on WANs with **more variable** demands



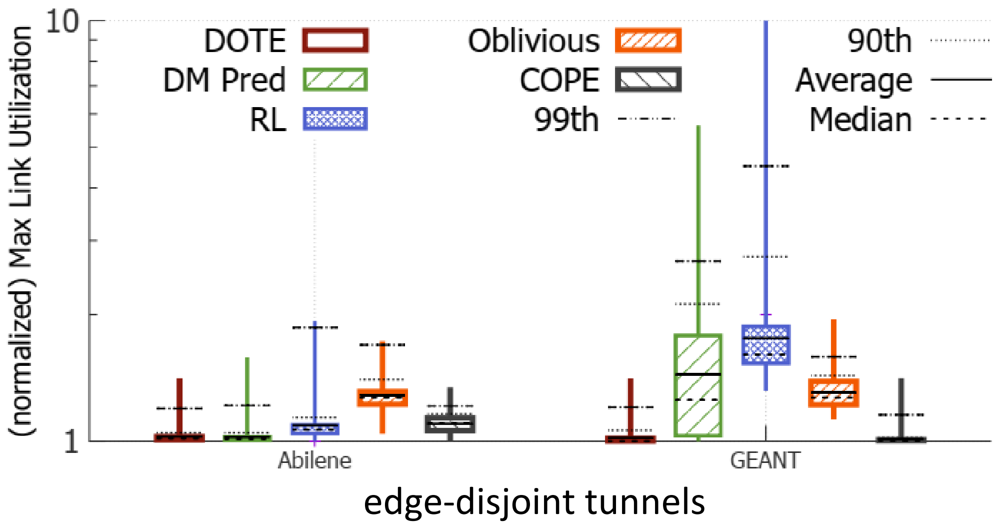

edge-disjoint tunnels

# Minimizing Maximum Link Utilization

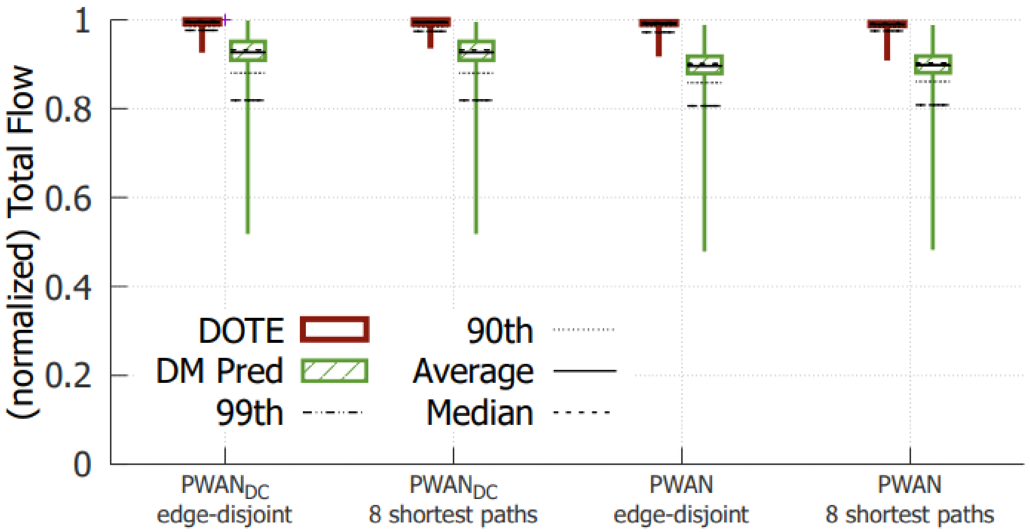
- **DOTE outperforms other TE approaches** in terms of TE quality
  - COPE could not scale to PWAN

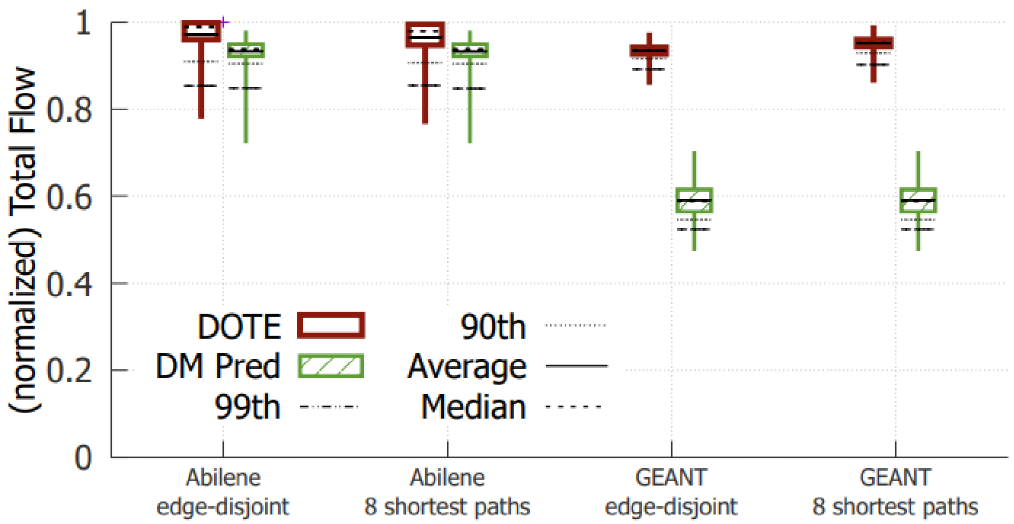



| | 99th | 90th | 75th | median |
|---|---|---|---|---|
| DOTE | +32% | +22% | +18% | +9% |
| Demand Prediction | +201% | +52% | +32% | +18% |

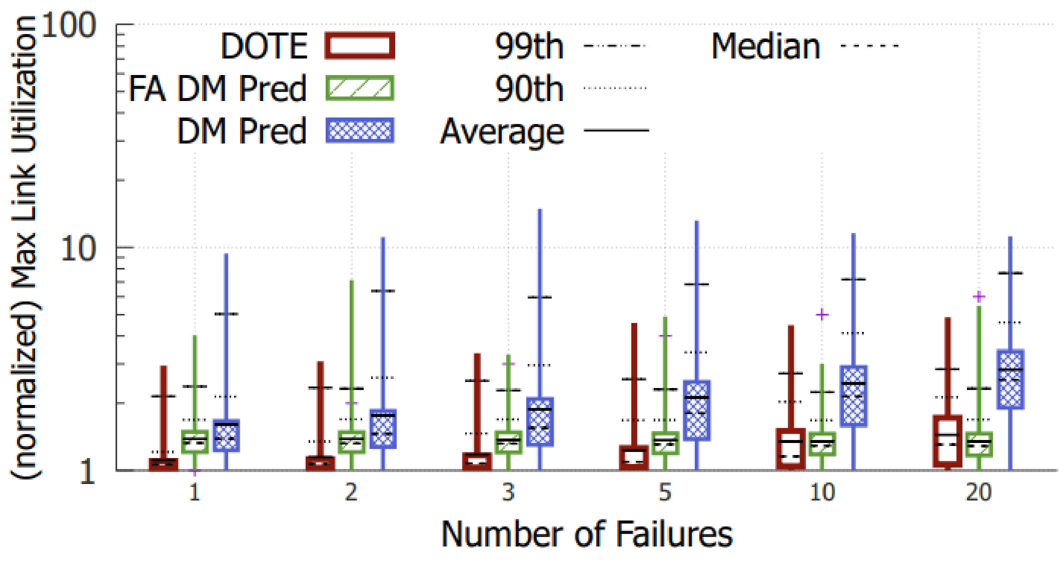


# Maximizing Total Flow Carried

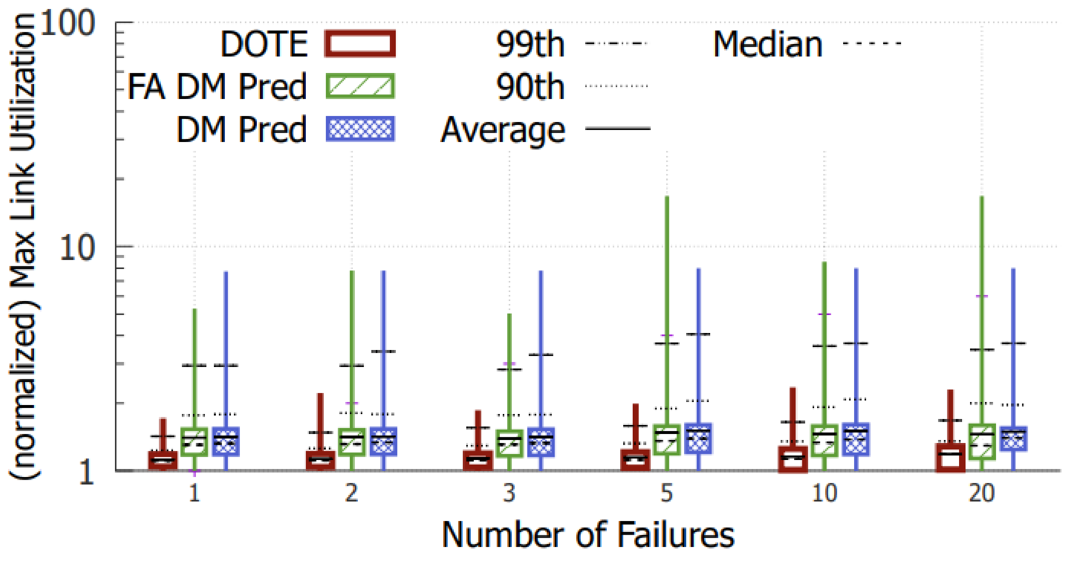


# Performance under link failures (for PWAN)




8 shortest paths




edge-disjoint

- **FA DM Pred** is demand-prediction-based TE with **perfectly knows future failures**
- DOTE outperforms “FA DM Pred”.
- **Takeaway**: **Demand variability** has more effect than **network failures** (up to a certain number of failures)

# DOTE also improves runtimes!

| **WAN** | Online Lat. (s) | | | |
|---|---|---|---|---|
| | DOTE | LP | #nodes | #edges |
| Abilene | 0.0005 | 0.02 | 11 | 14 |
| $PWAN_{DC}$ | 0.003 | 0.05 | O(10) | O(10) |
| Geant | 0.002 | 0.04 | 23 | 37 |
| PWAN | 0.2 | 1 | O(100) | O(100) |
| KDL | 2 | 30 | 754 | 895 |

# See paper for

- More results on **TE quality**
  - Additional tunnel selection schemes
  - Additional performance metrics
- Robustness to **noisy traffic**
  - Different topologies, levels of noise
- Robustness to **natural traffic drift**
  - Different topologies, tunnel selection schemes, and performance metrics
- More results on **resiliency to link failures**
  - Different topologies, tunnel selection schemes, and performance metrics
- **Comparison of demand-prediction methods** for TE
- More results on **runtimes**
  - Additional benchmarks (oblivious routing, COPE)

# Conclusion

- DOTE is **novel approach to WAN TE: directly optimizing TE configurations** (subsumes demand prediction)
- A simple learning method that extends to multiple TE objectives
- **DOTE's TE quality improves over the state-of-the-art, closely approximating the omniscient oracle**
- DOTE also significantly improves online runtimes for TE.

# Future Research

- Learning tunnels?
- Learning to cope with failures?
- Incorporating the network topology into the DNN?
- Accelerating (offline and online) runtimes?

Thank you!

- Paper: https://www.usenix.org/system/files/nsdi23-perry.pdf
- Code: https://github.com/PredWanTE/DOTE