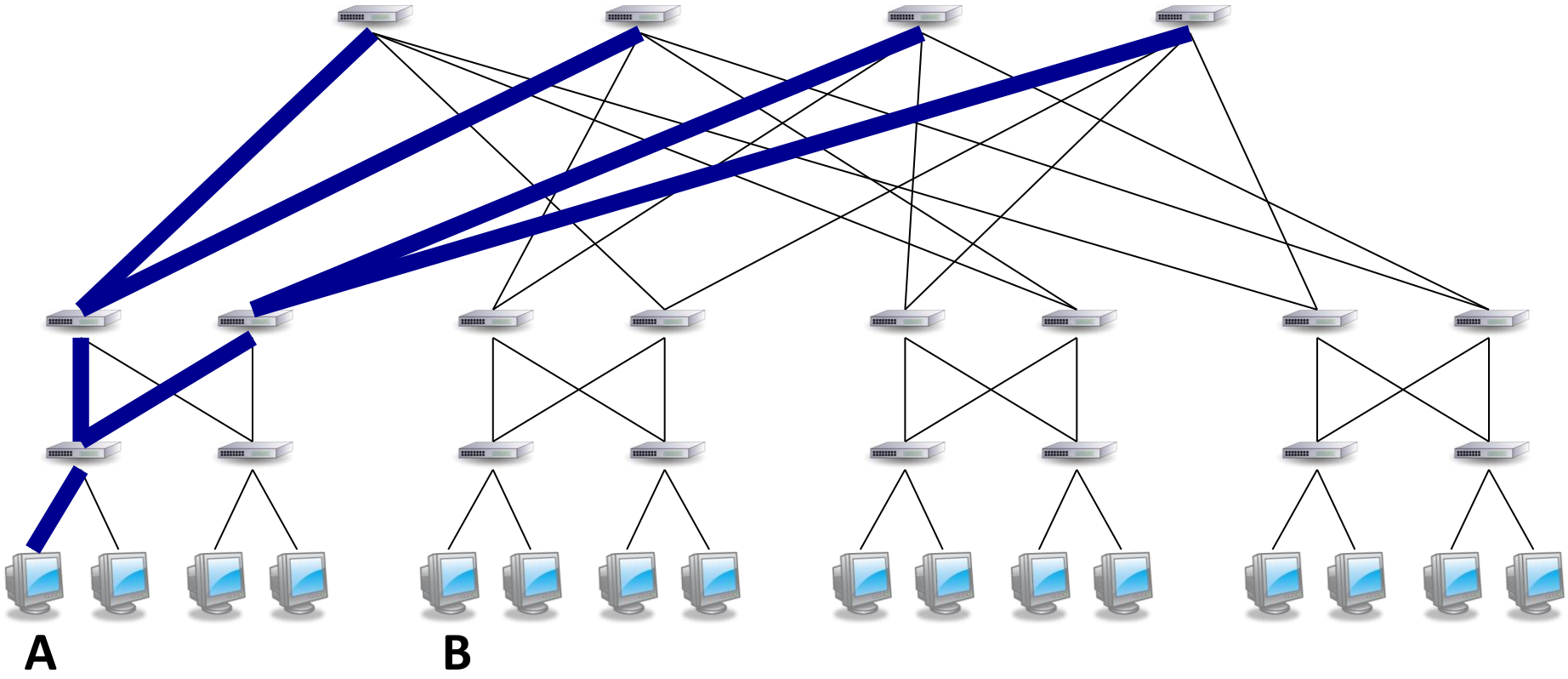
Bridging the Theory-Practice Gap in Multi-Commodity Flow Routing

Siddhartha Sen, DISC 2011

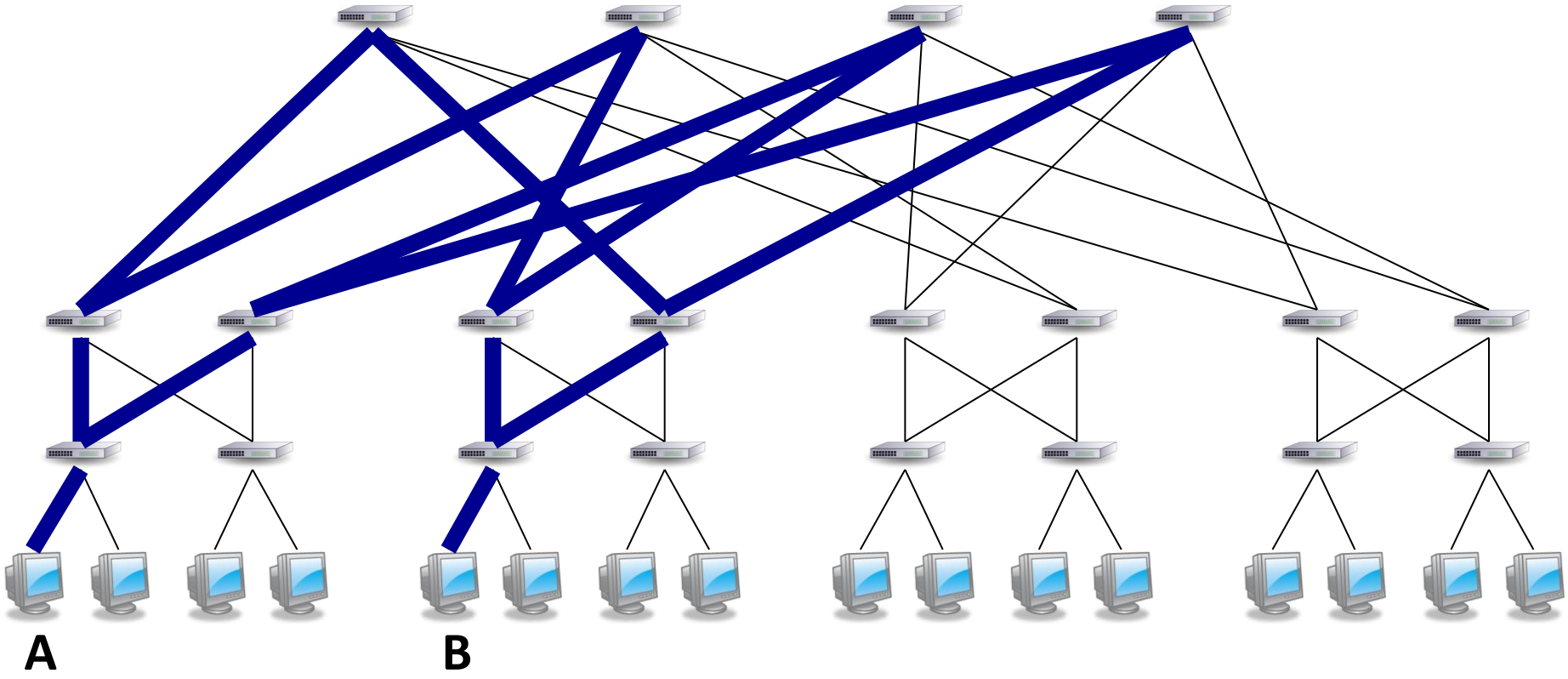
Joint work with Sunghwan Ihm, Kay Ousterhout, and Mike Freedman

Princeton University

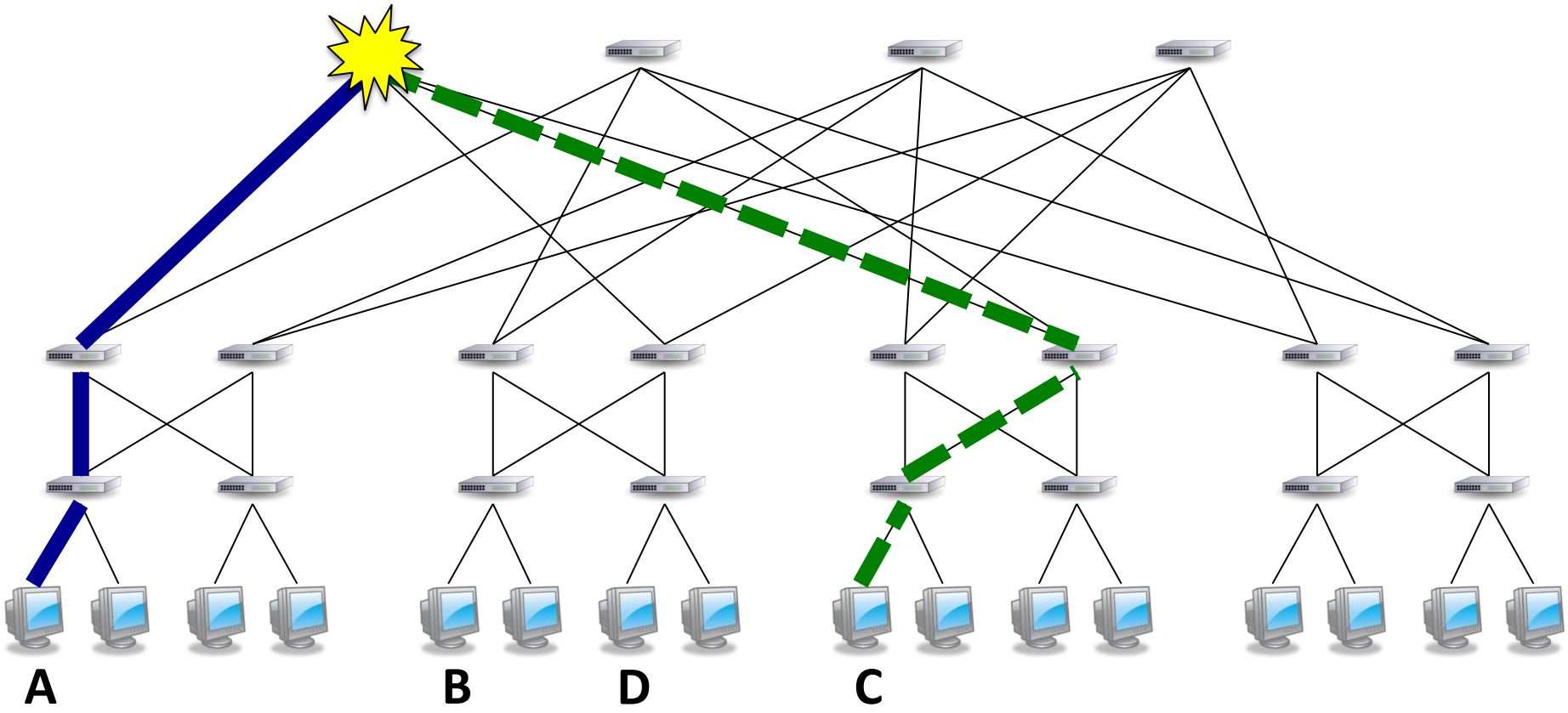
Routing flows in data center networks



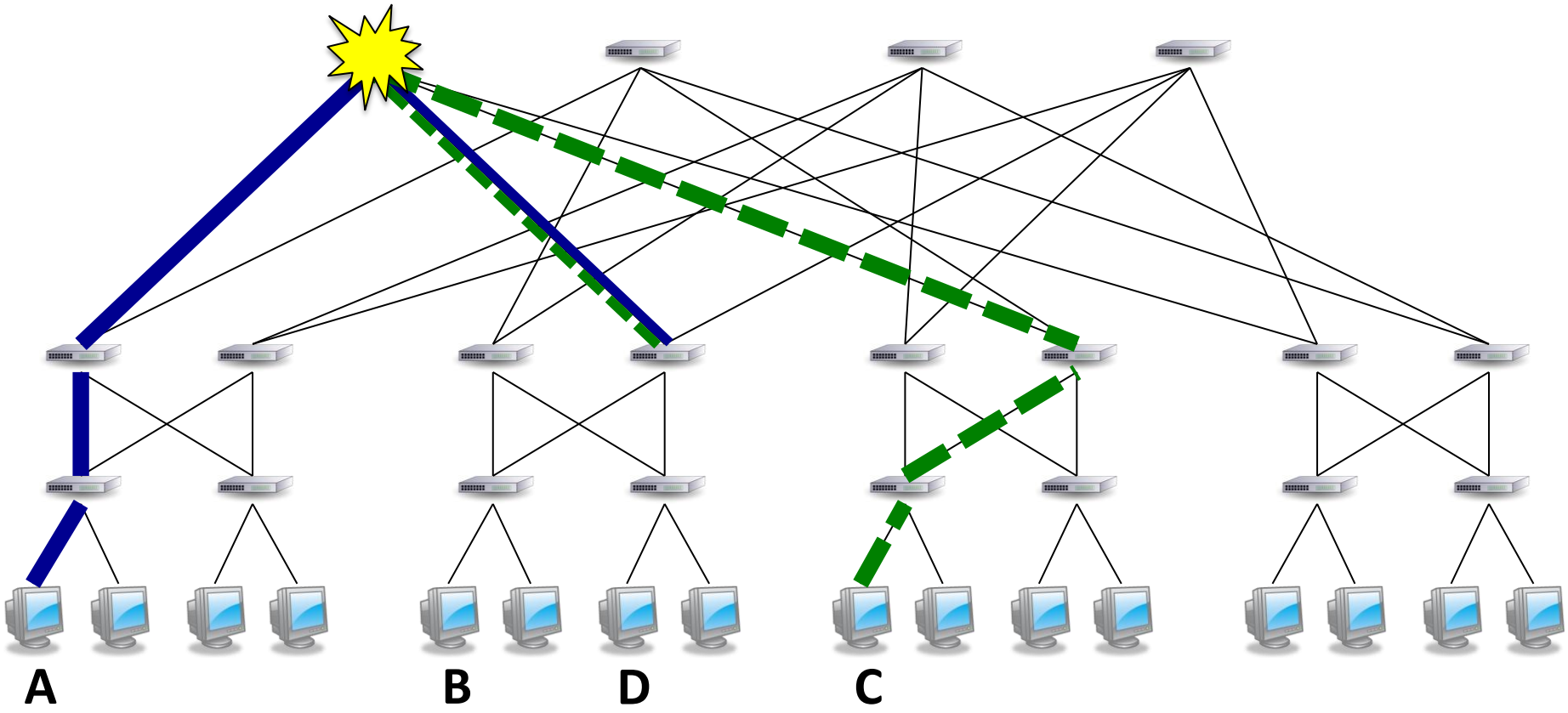
Routing flows in data center networks



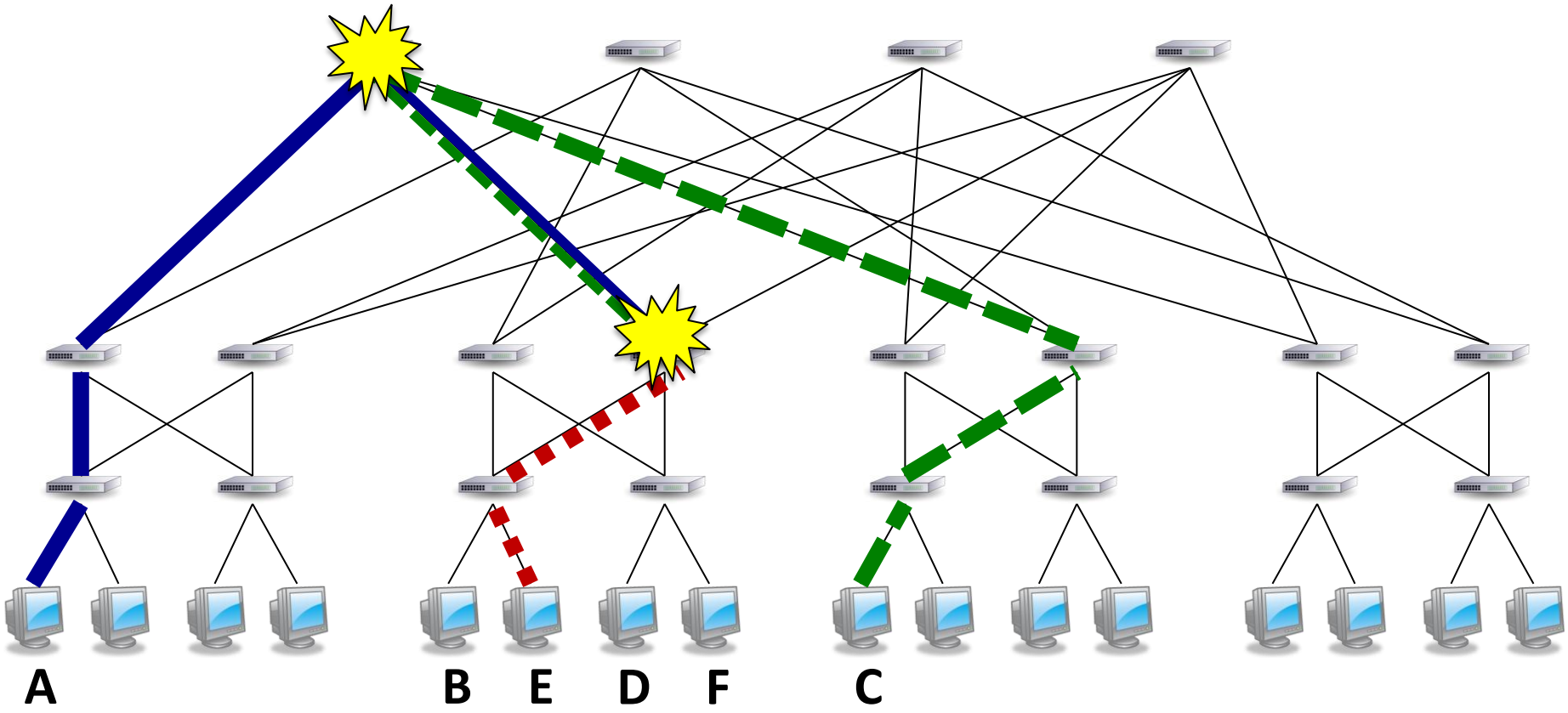
Network utilization suffers when flows collide…



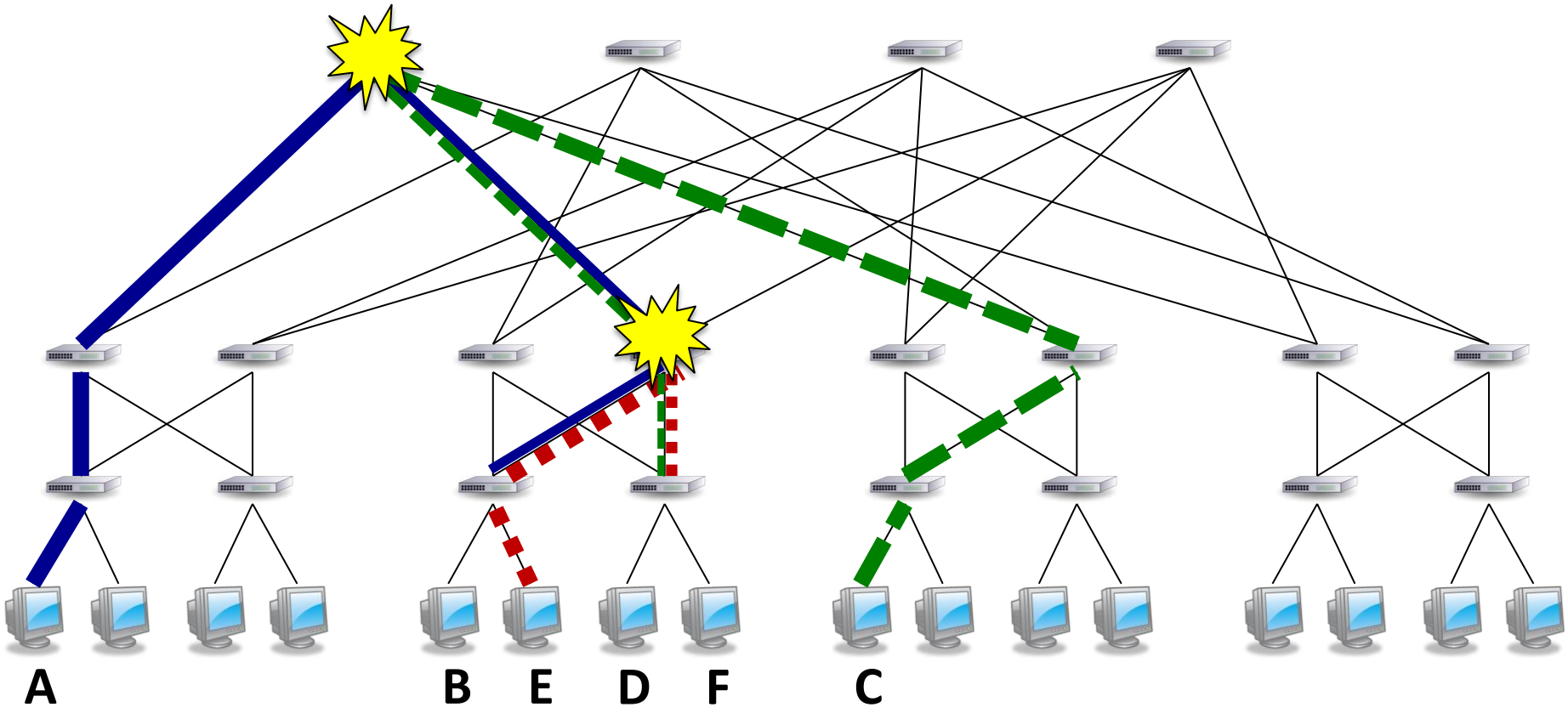
Network utilization suffers when flows collide…



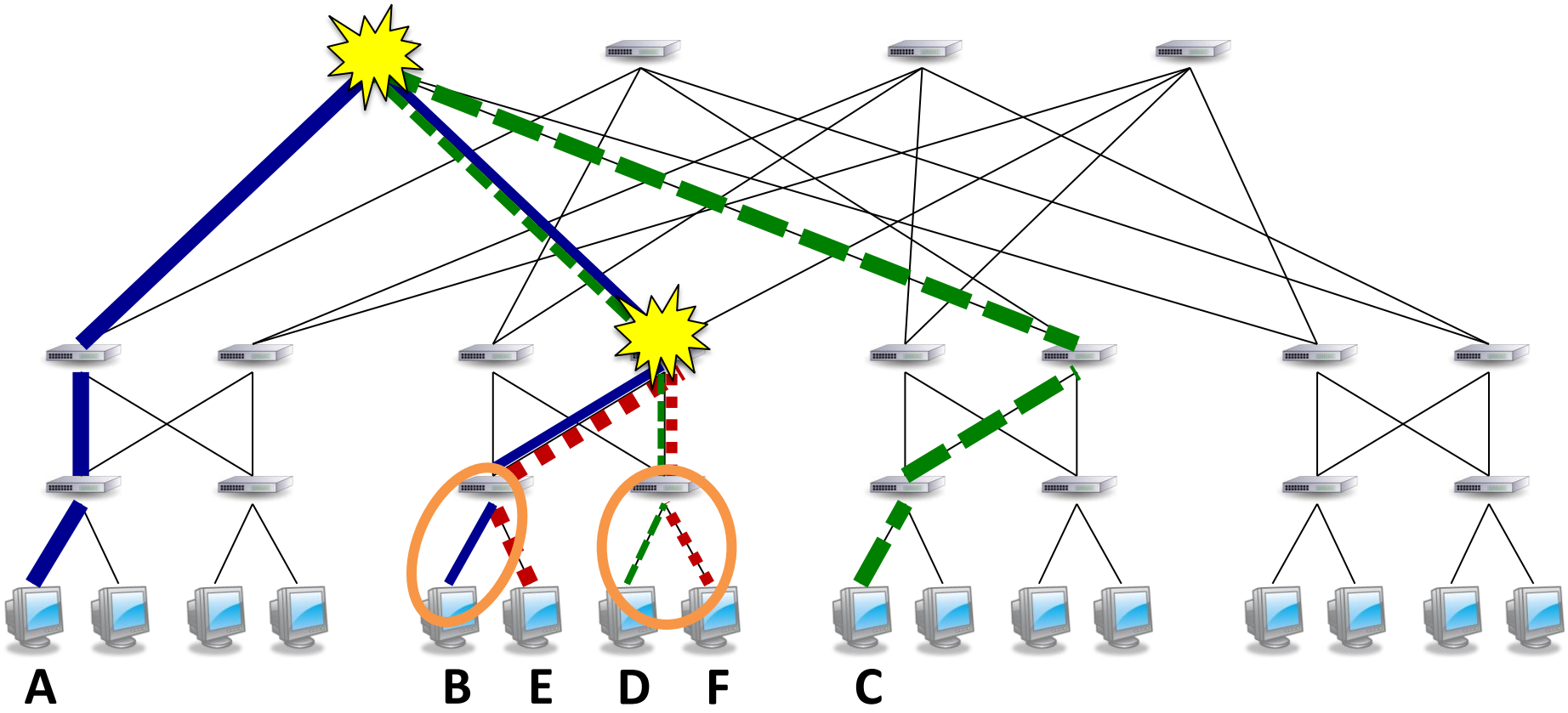
Network utilization suffers when flows collide…



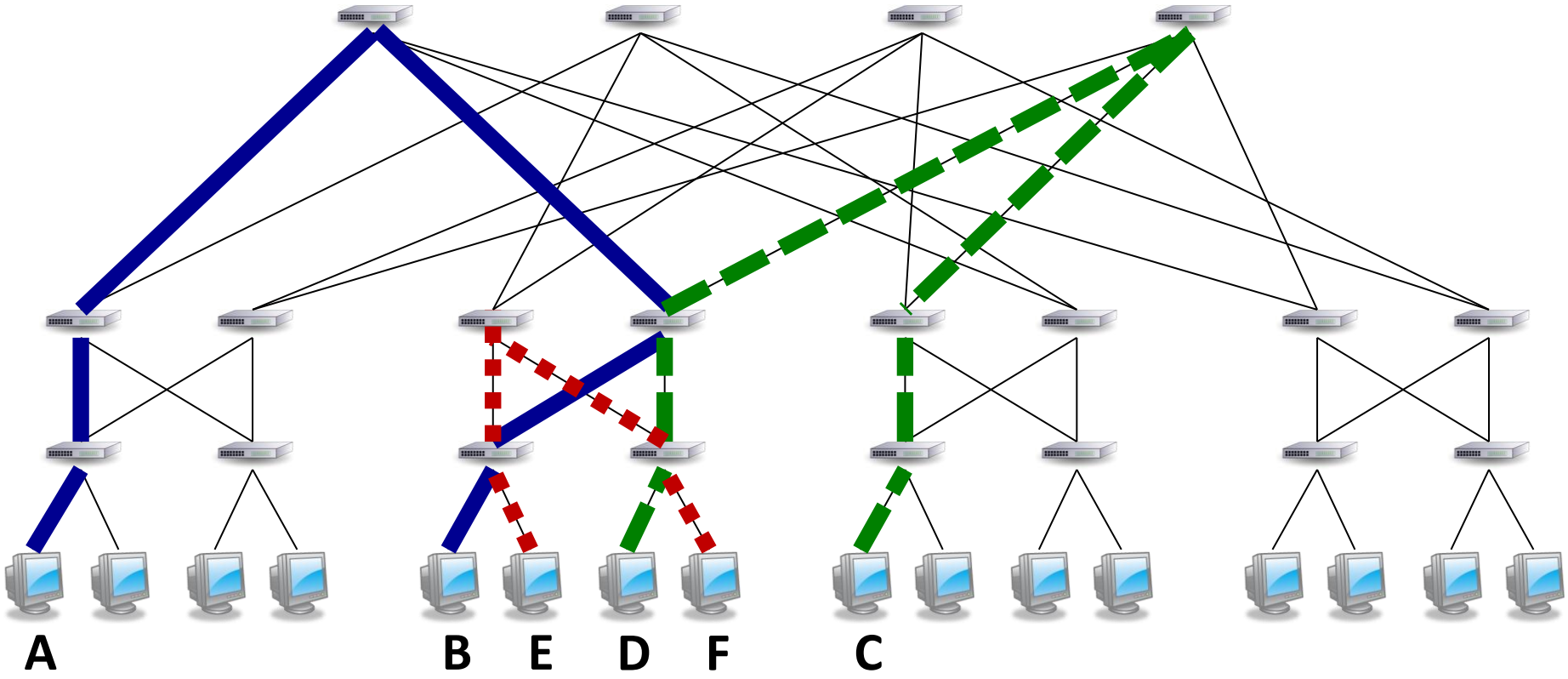
Network utilization suffers when flows collide…



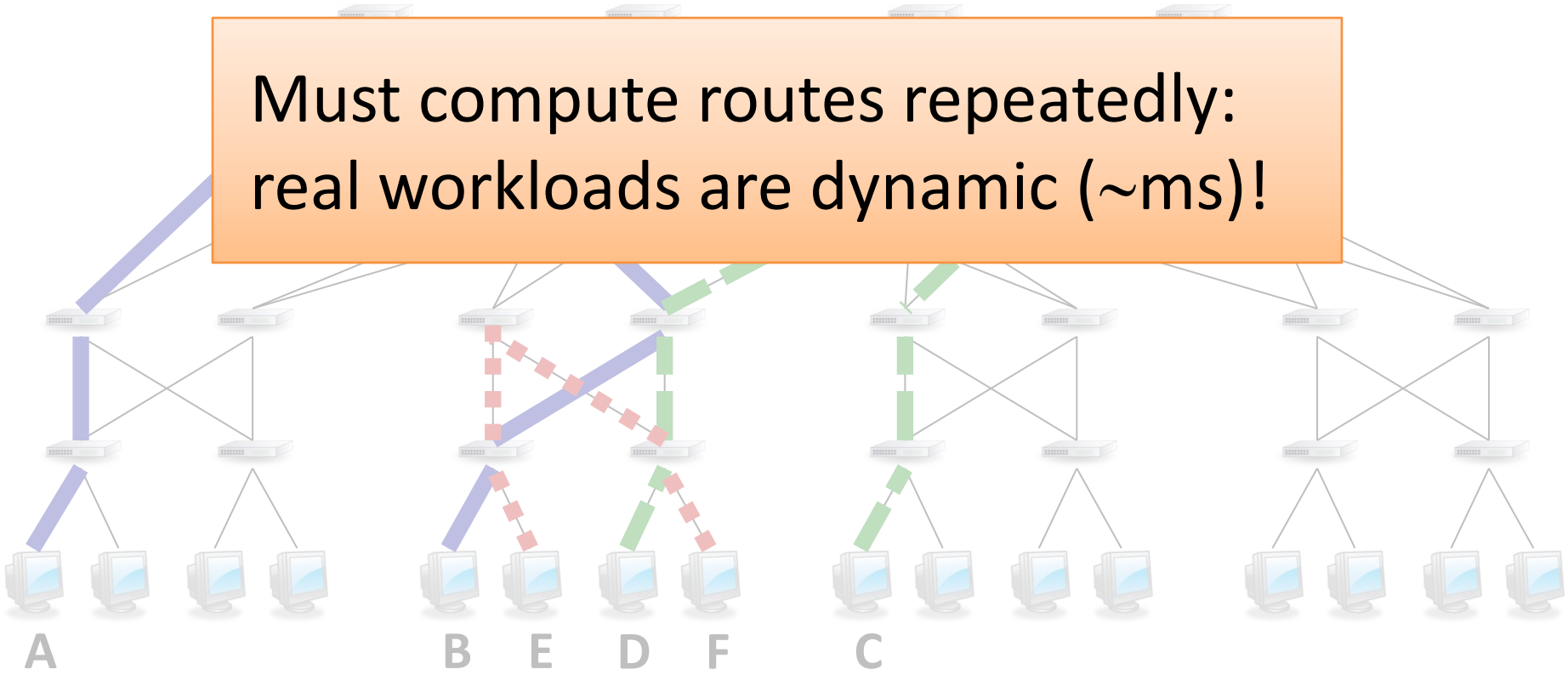
Network utilization suffers when flows collide…



… but there is available capacity!



… but there is available capacity!



Multi-commodity flow problem

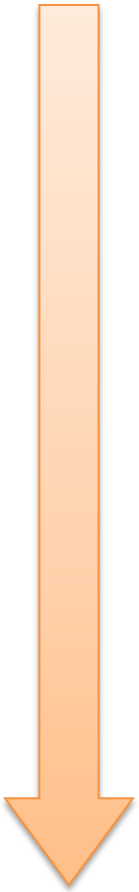
- **Input:** Network $G = (V,E)$ of switches and links
  Flows $K = \{(s_i,t_i,d_i)\}$ of source, target, demand tuples

- **Goal:** Compute flow that maximizes minimum **fraction** of any d_i routed

- Requires fractionally splitting flows, otherwise no O(1)-factor approximation

Prior solutions

- Sequential model
- Billboard model
- Routers model



more decentralized

Prior solutions

- Sequential model
  - Theory: [Vaidya89, PlotkinST95, GargK07, …]
  - Practice: [BertsekasG87, BurnsOKM03, Hedera10, …]
- Billboard model
  - Theory: [AwerbuchKR07, AwerbuchK09, …]
  - Practice: [MATE01, TeXCP05, MPTCP11, ...]
- Routers model
  - Theory: [AwerbuchL93, AwerbuchL94, AwerbuchK07, …]
  - Practice: [REPLEX06, COPE06, FLARE07, …]

Prior solutions

- Sequential model
  - Theory: [Vaidya89, PlotkinST95, GargK07, …]
-
- Routers model
  - Theory: [AwerbuchL93, AwerbuchL94, AwerbuchK07, …]
  - Practice: [REPLEX06, COPE06, FLARE07, …]

Theory-practice gap:

1. Models unsuitable for dynamic workloads
2. Splitting flows difficult in practice

Contributions

- Demonstrate why prior solutions fail, both theoretically and practically
- Propose theoretical + practical fixes
- Devise algorithms in new framework

Contributions

- Demonstrate why prior solutions fail, both theoretically and practically
- Propose theoretical + practical fixes
- Devise algorithms in new framework

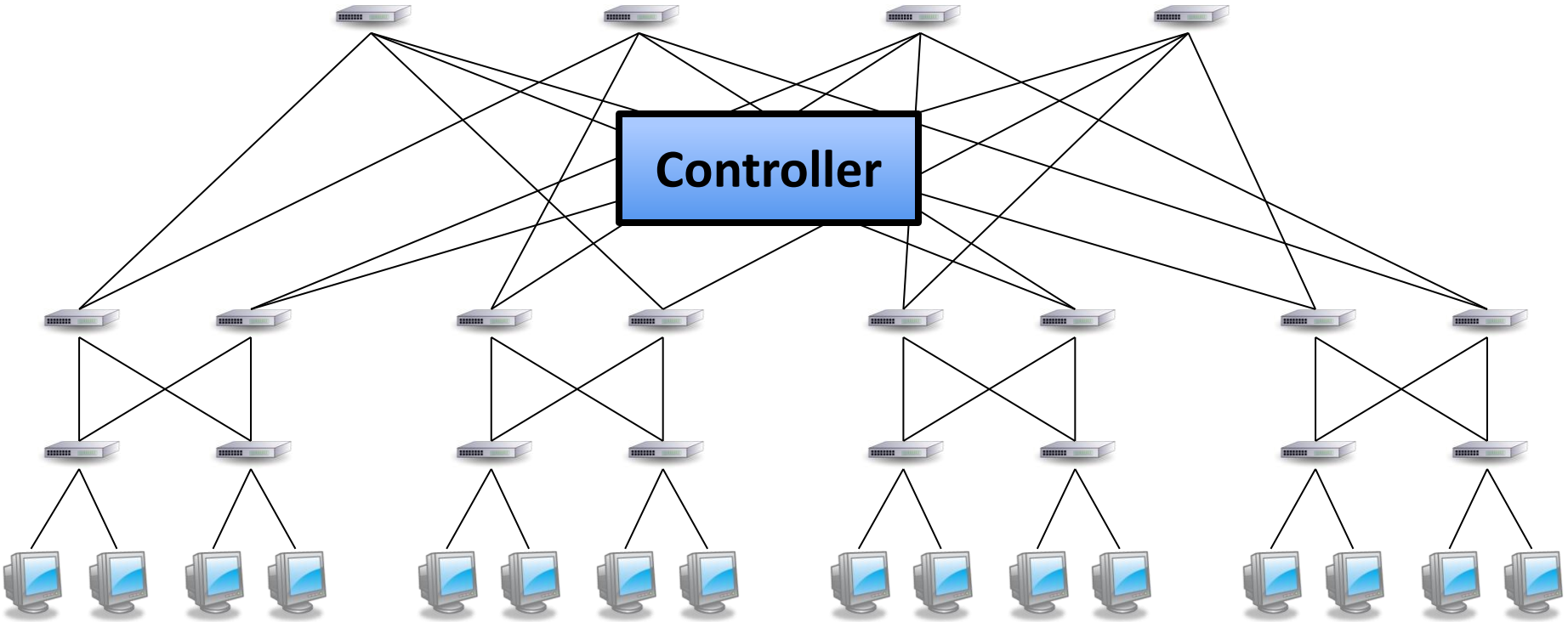
***Goal**: Provably optimal + practical multi-commodity flow routing*

Problems

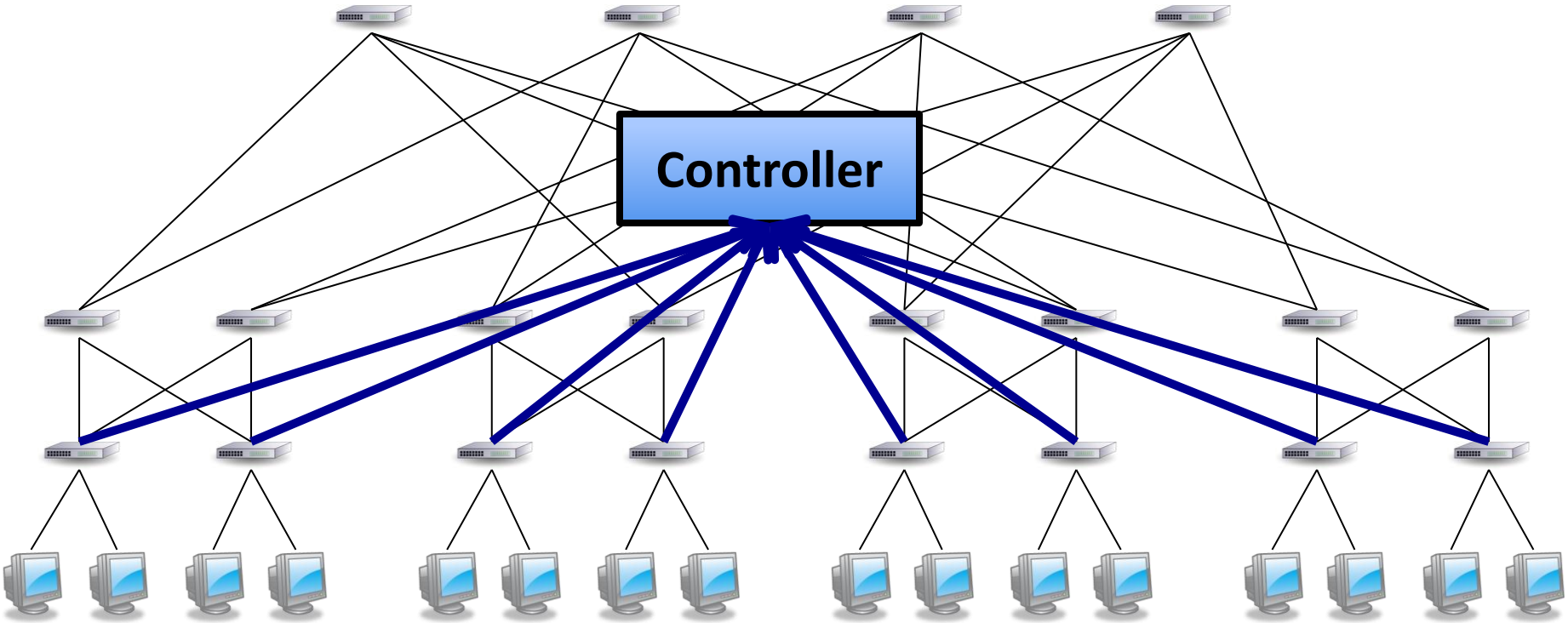
1. Dynamic workloads

Solutions

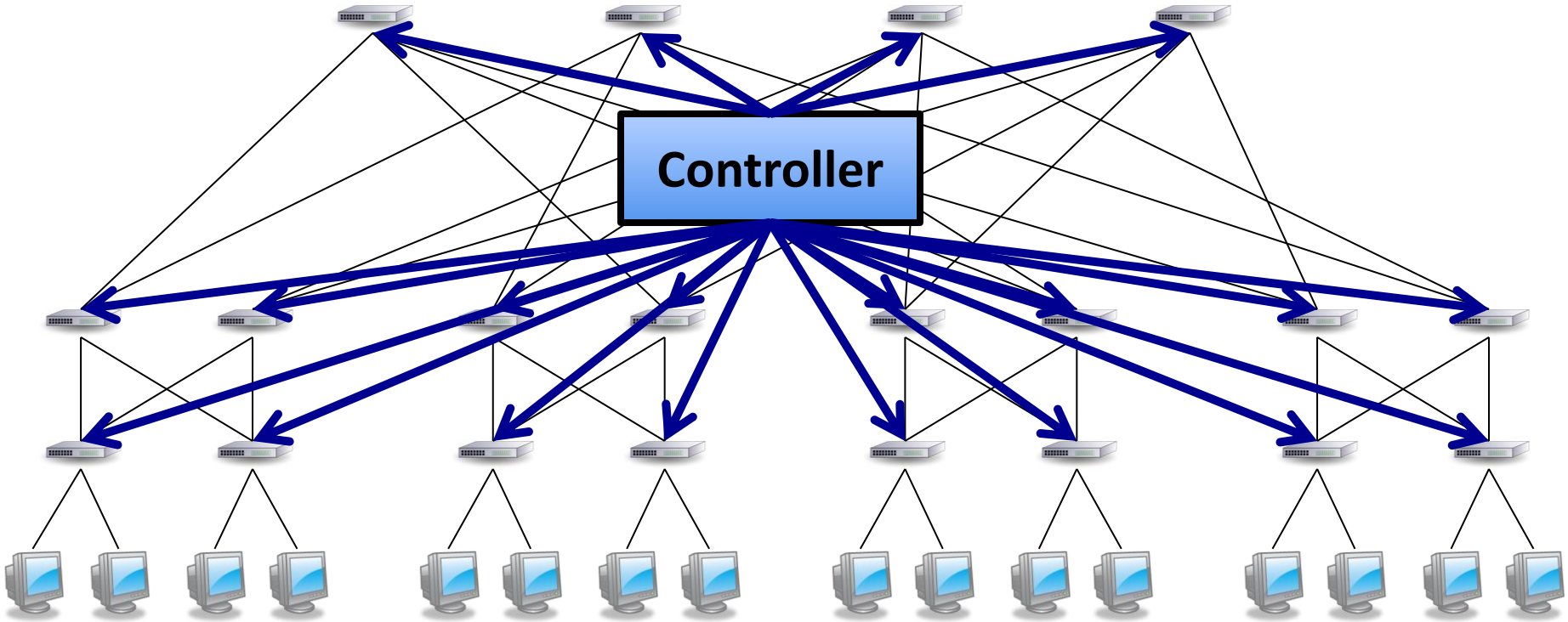
Sequential solutions don’t scale



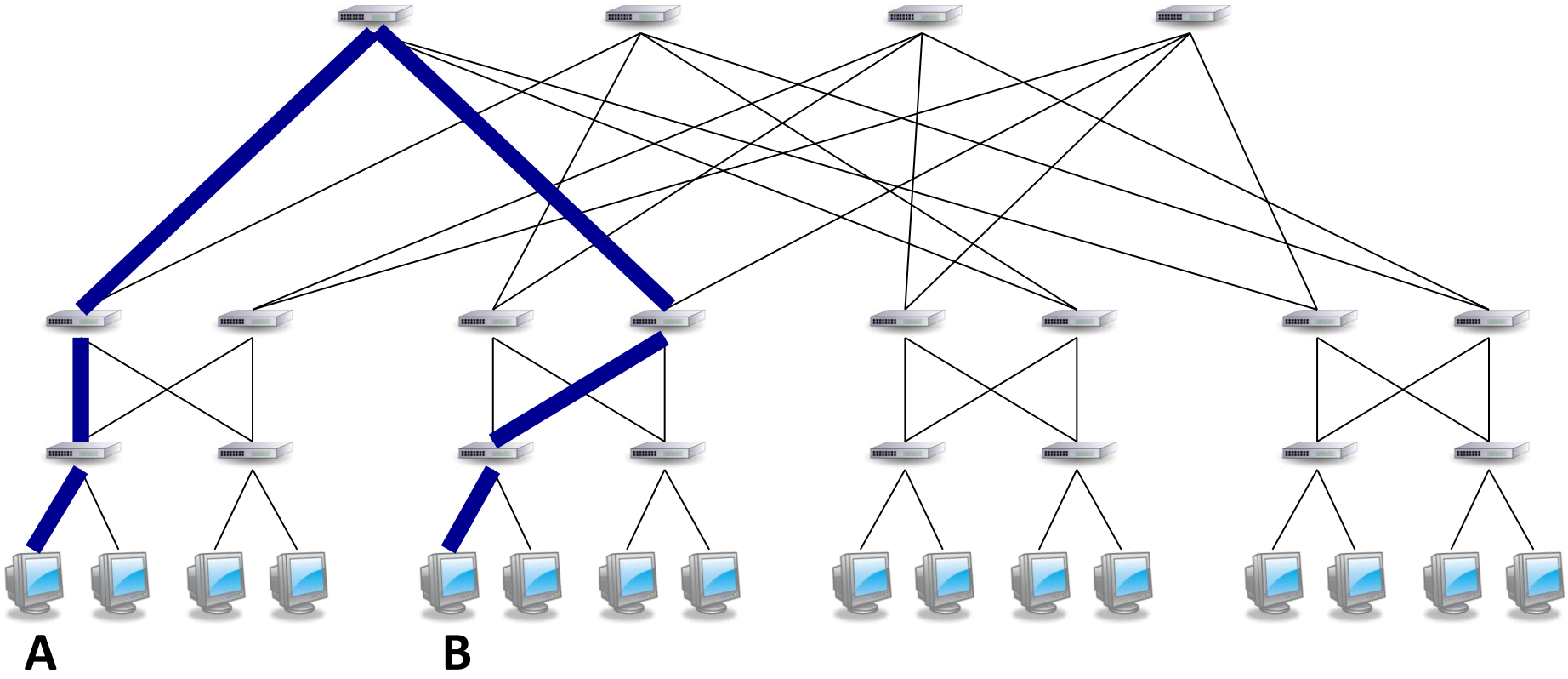
Sequential solutions don’t scale



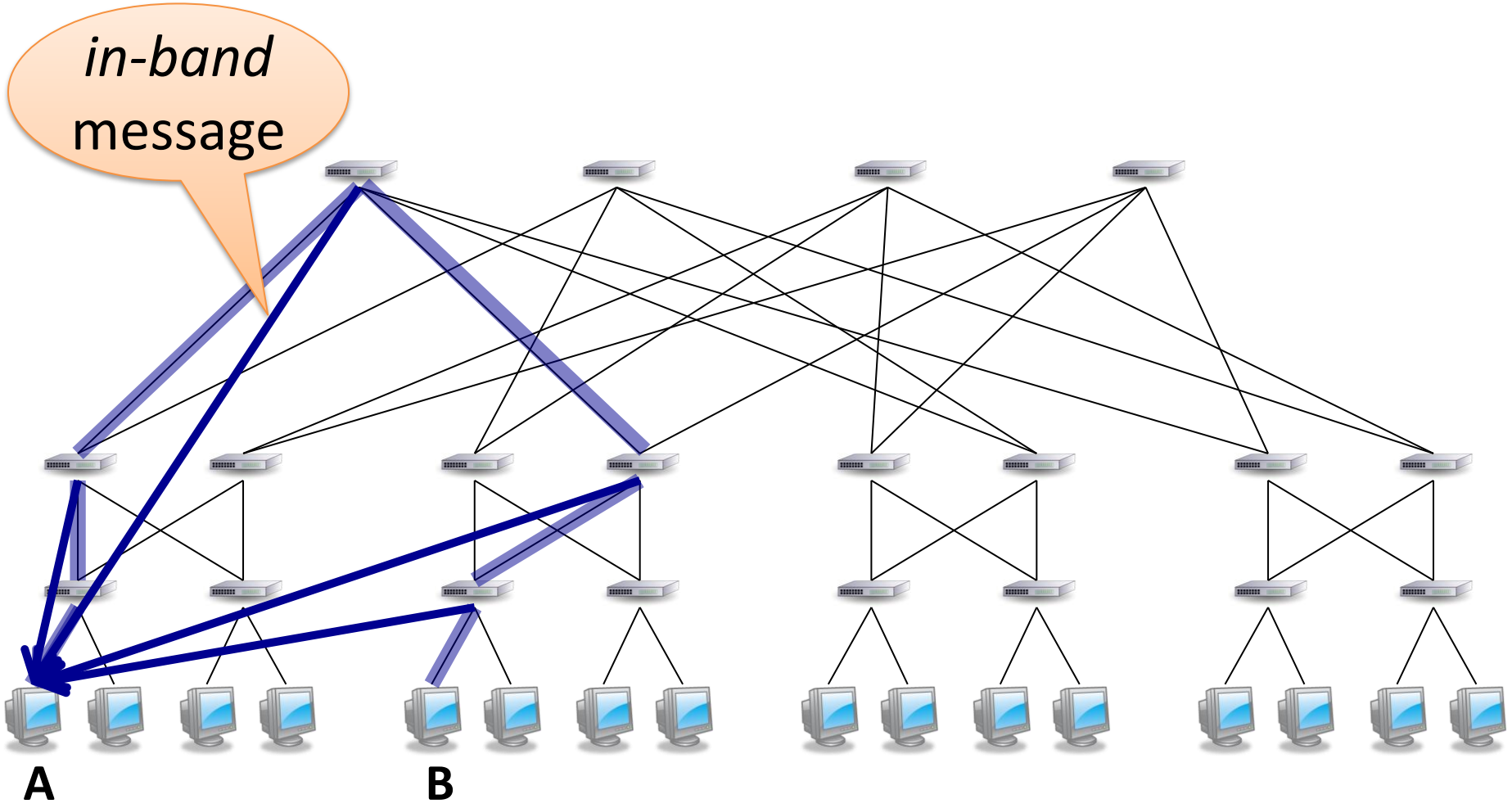
Sequential solutions don’t scale



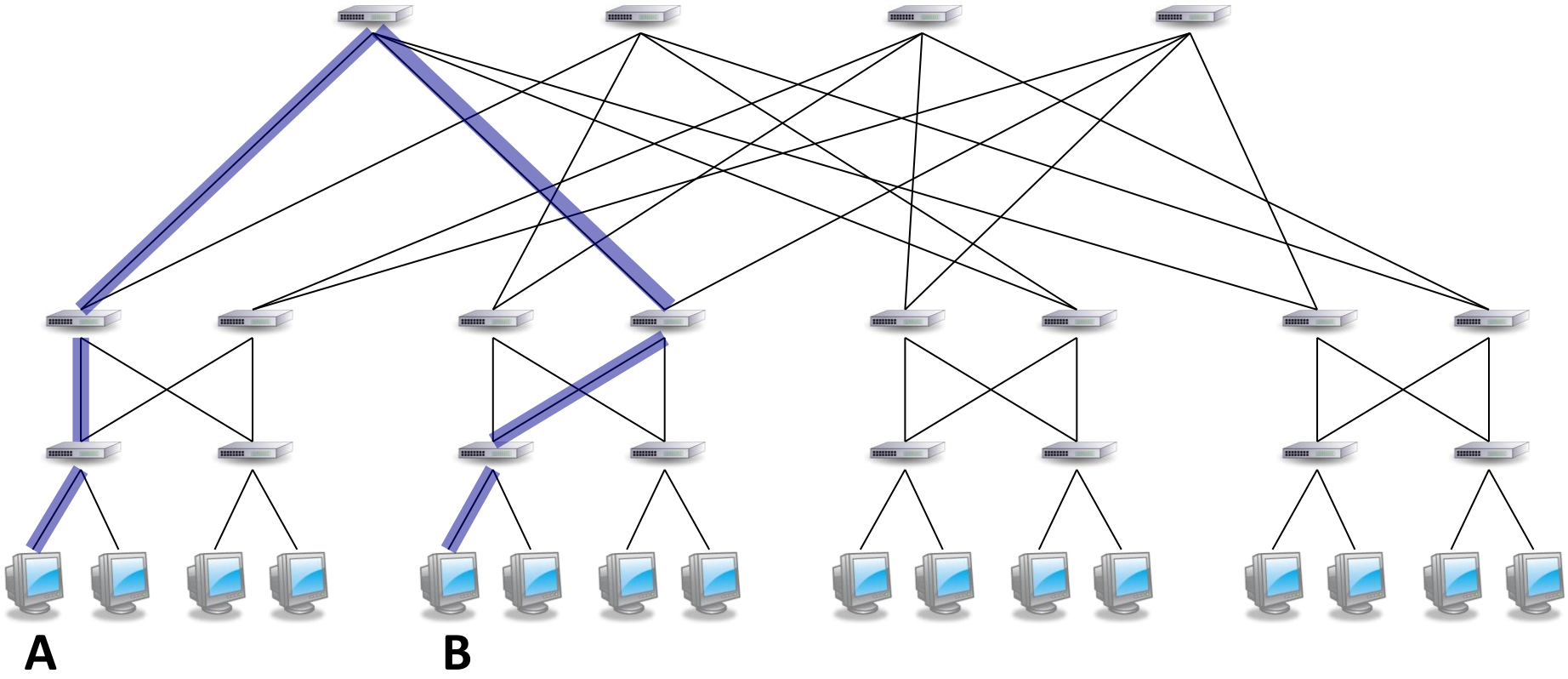
Billboard solutions require link utilization information…



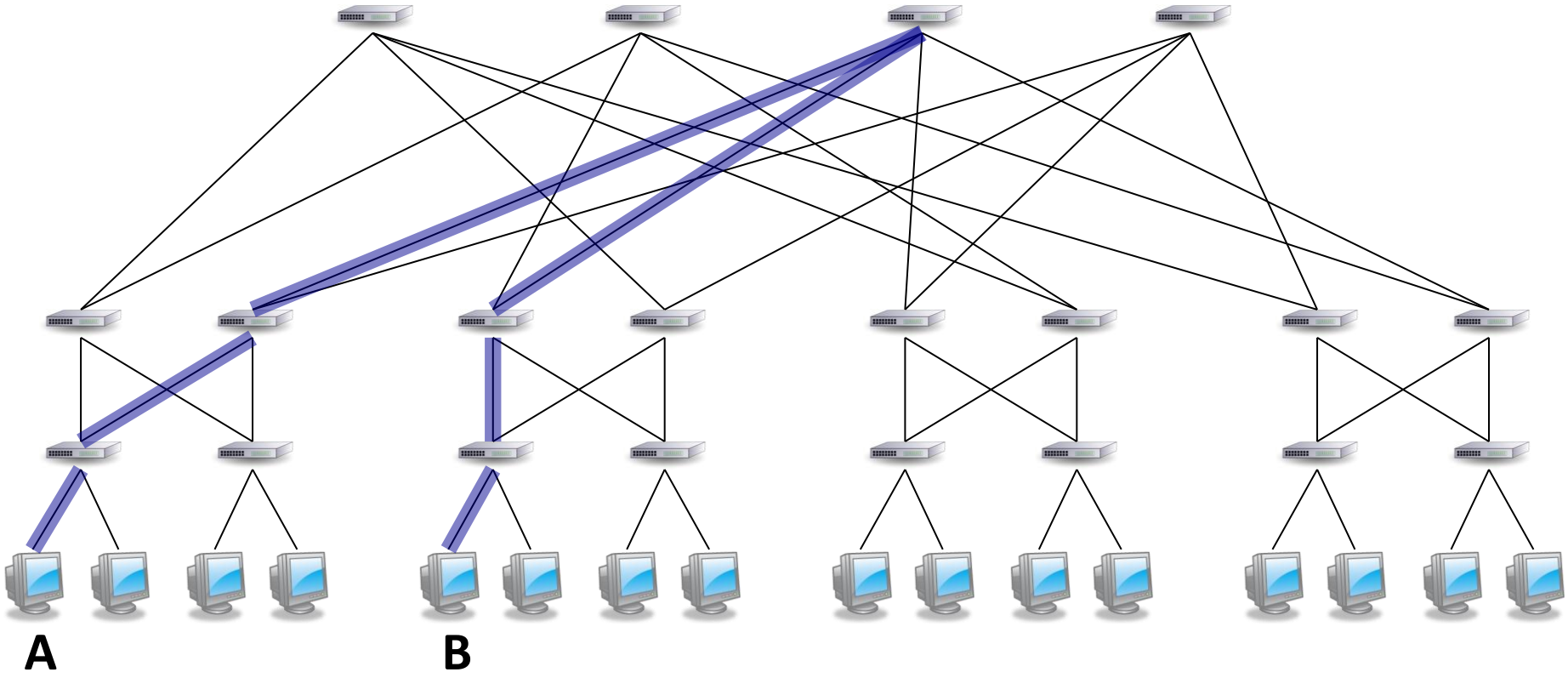
Billboard solutions require link utilization information…



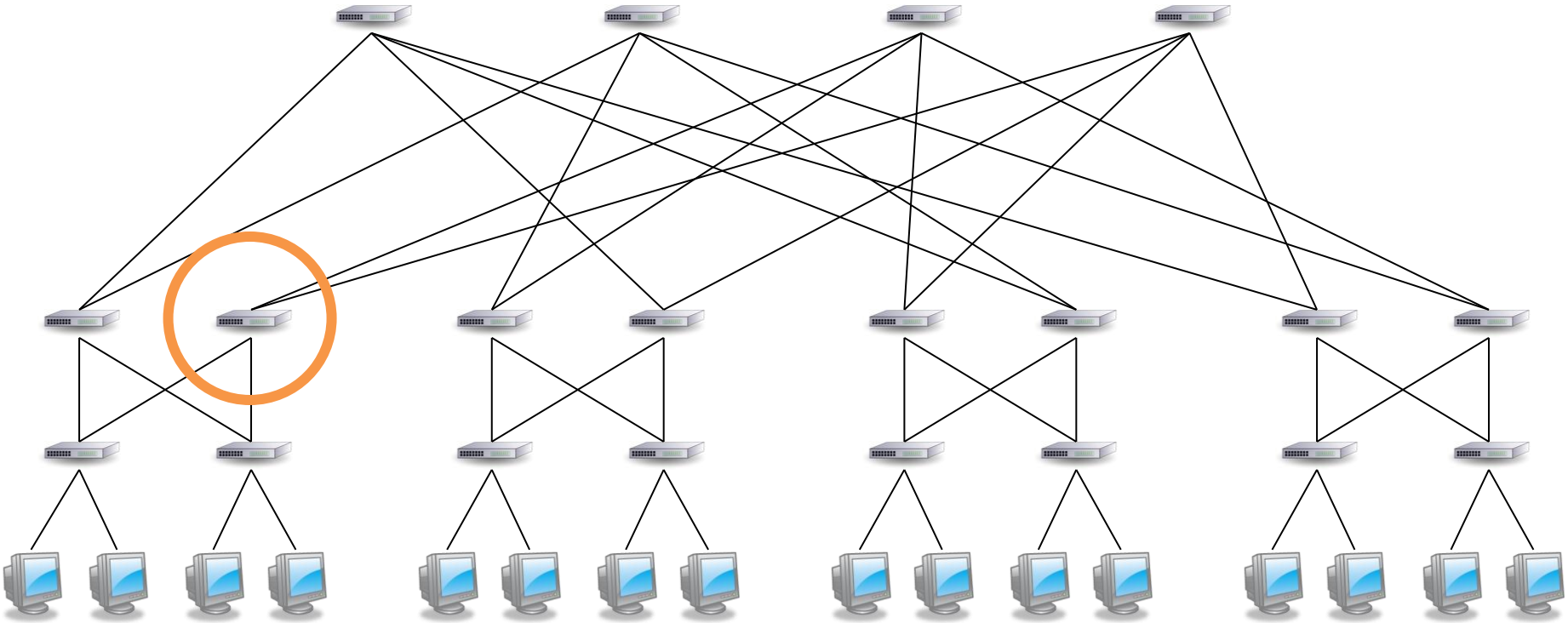
… and use path-based (exponential) representations



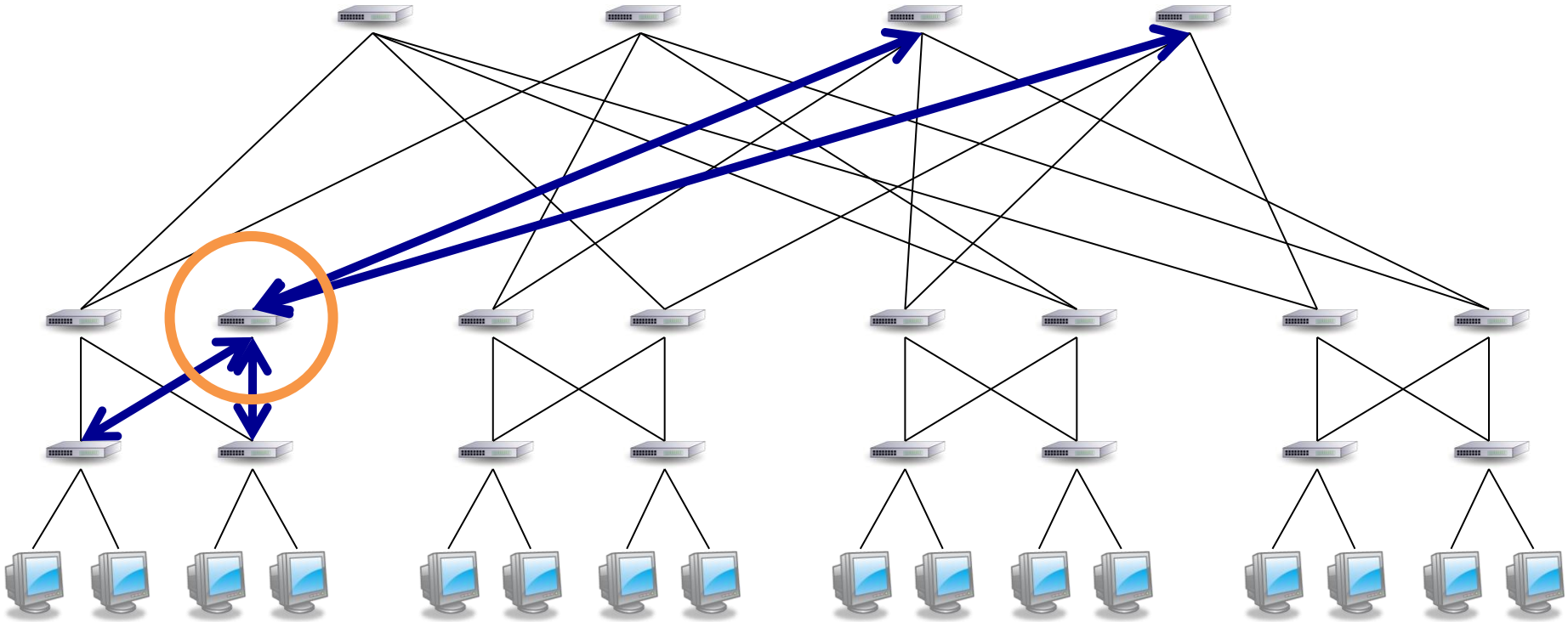
… and use path-based (exponential) representations



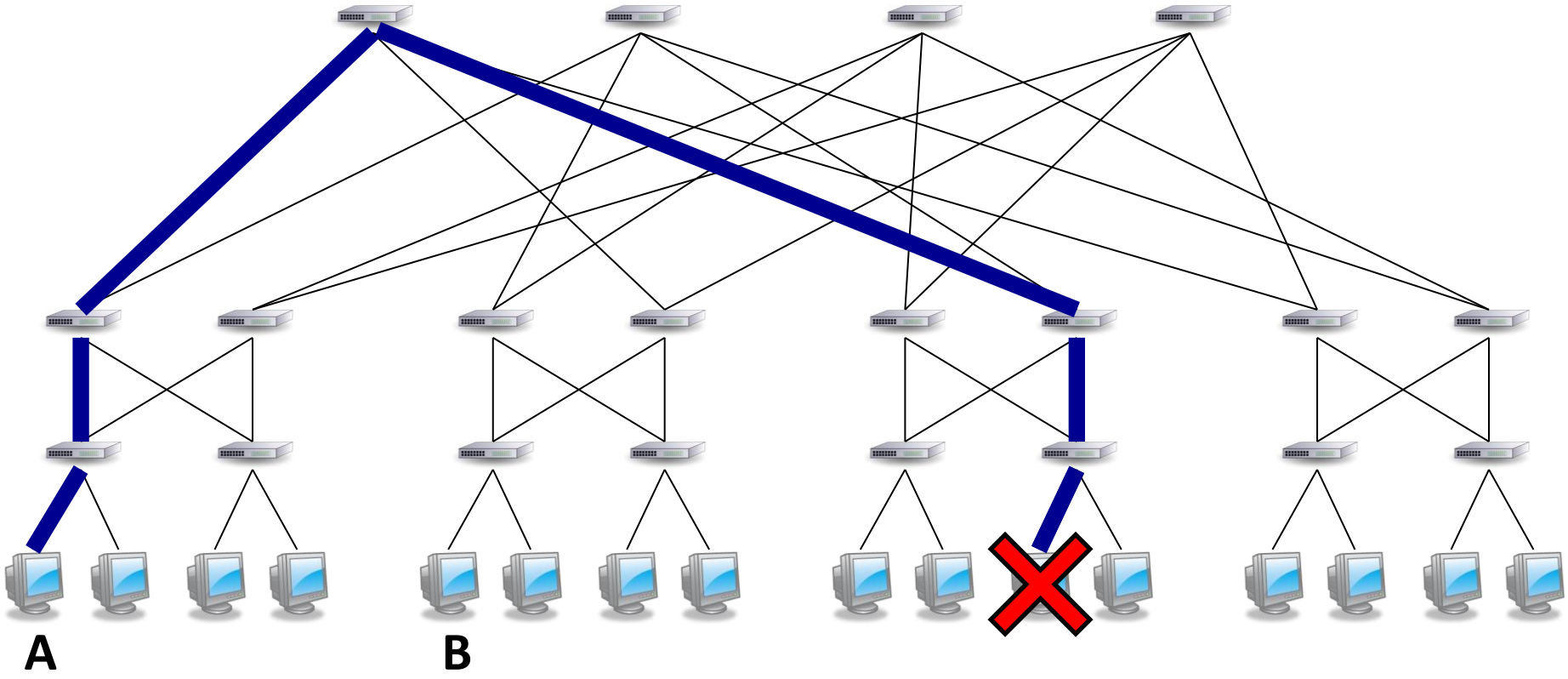
Routers solutions are local and hence scalable…



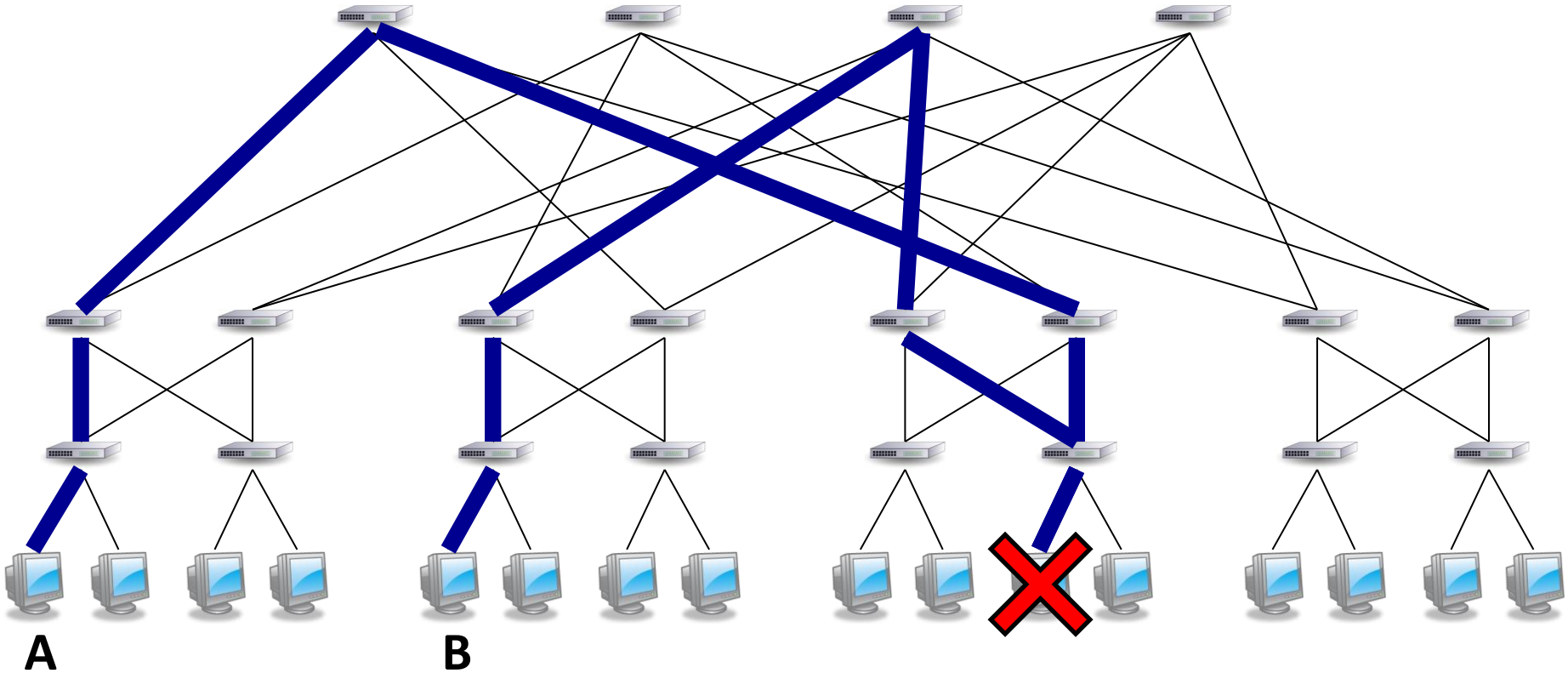
Routers solutions are local and hence scalable…



… but lack knowledge of global routes



… but lack knowledge of global routes



Problems

1. Dynamic workloads

Solutions

1. Routers Plus Preprocessing (RPP) model
   - Poly-time preprocessing is free
   - In-band messages are free

Problems

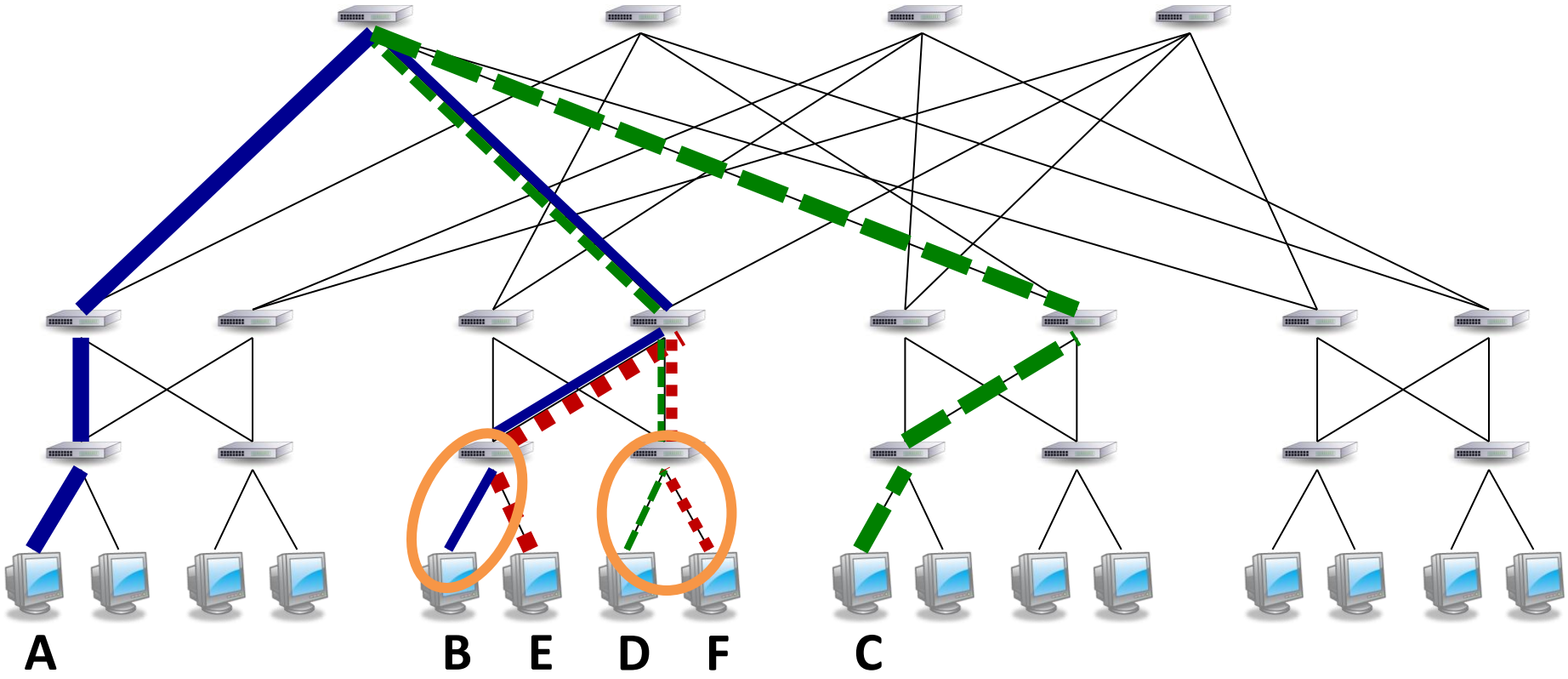
1. Dynamic workloads

2. Splitting flows

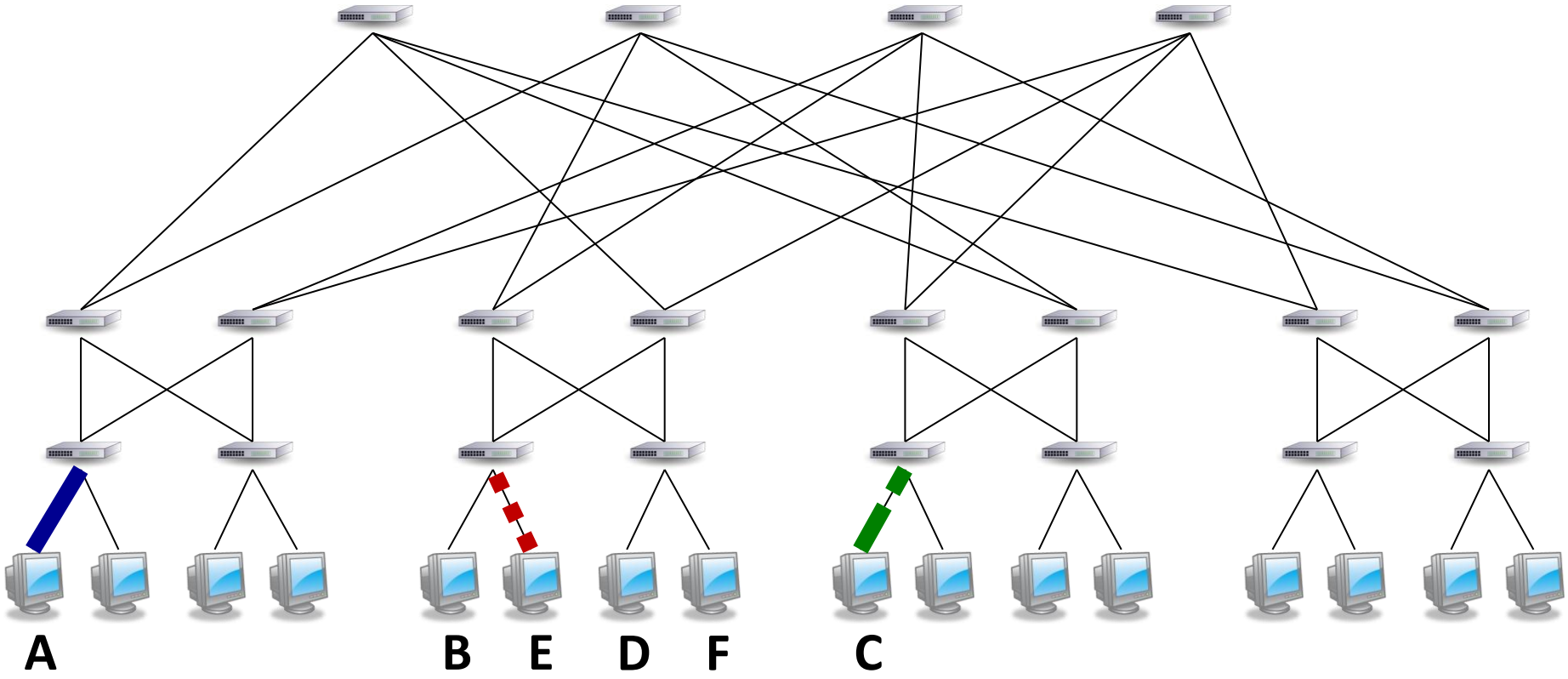
Solutions

1. Routers Plus Preprocessing (RPP) model
   - Poly-time preprocessing is free
   - In-band messages are free

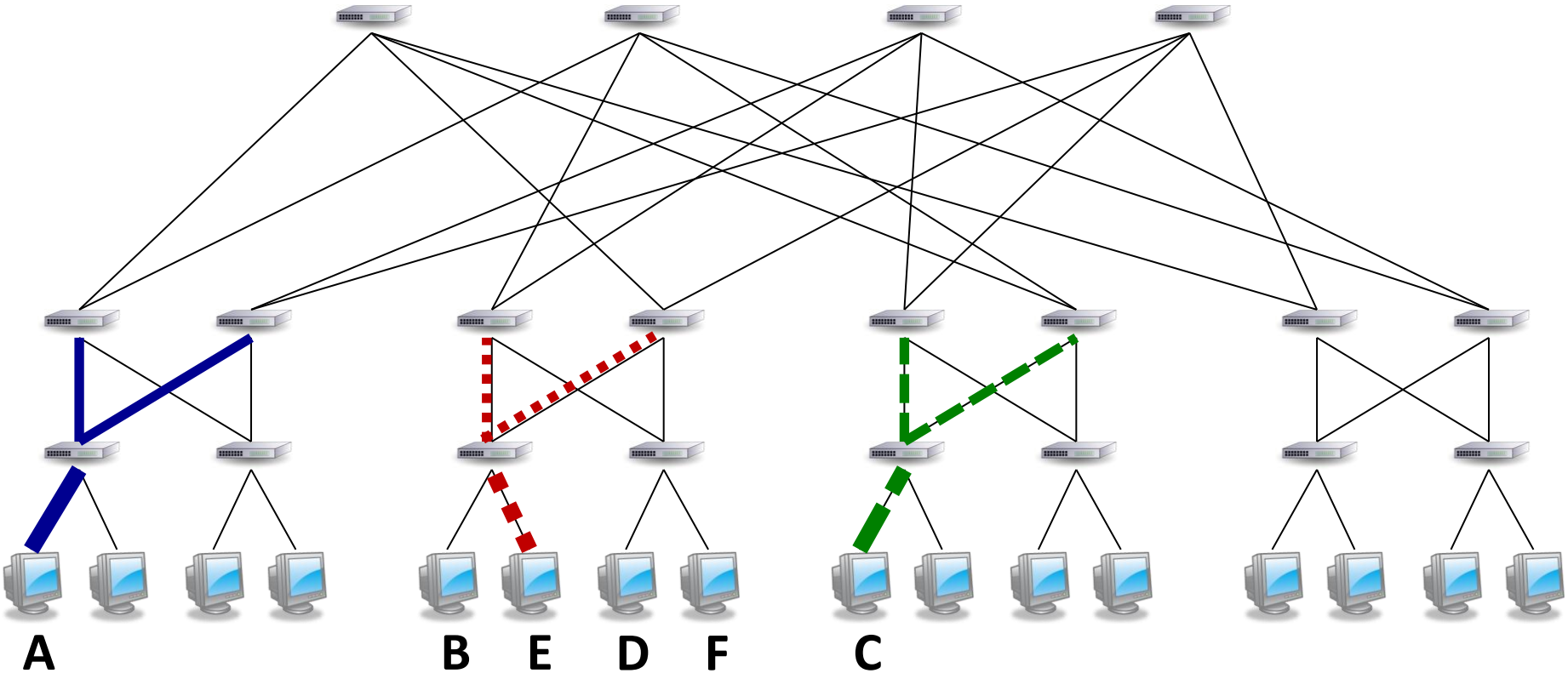
No splitting + collisions



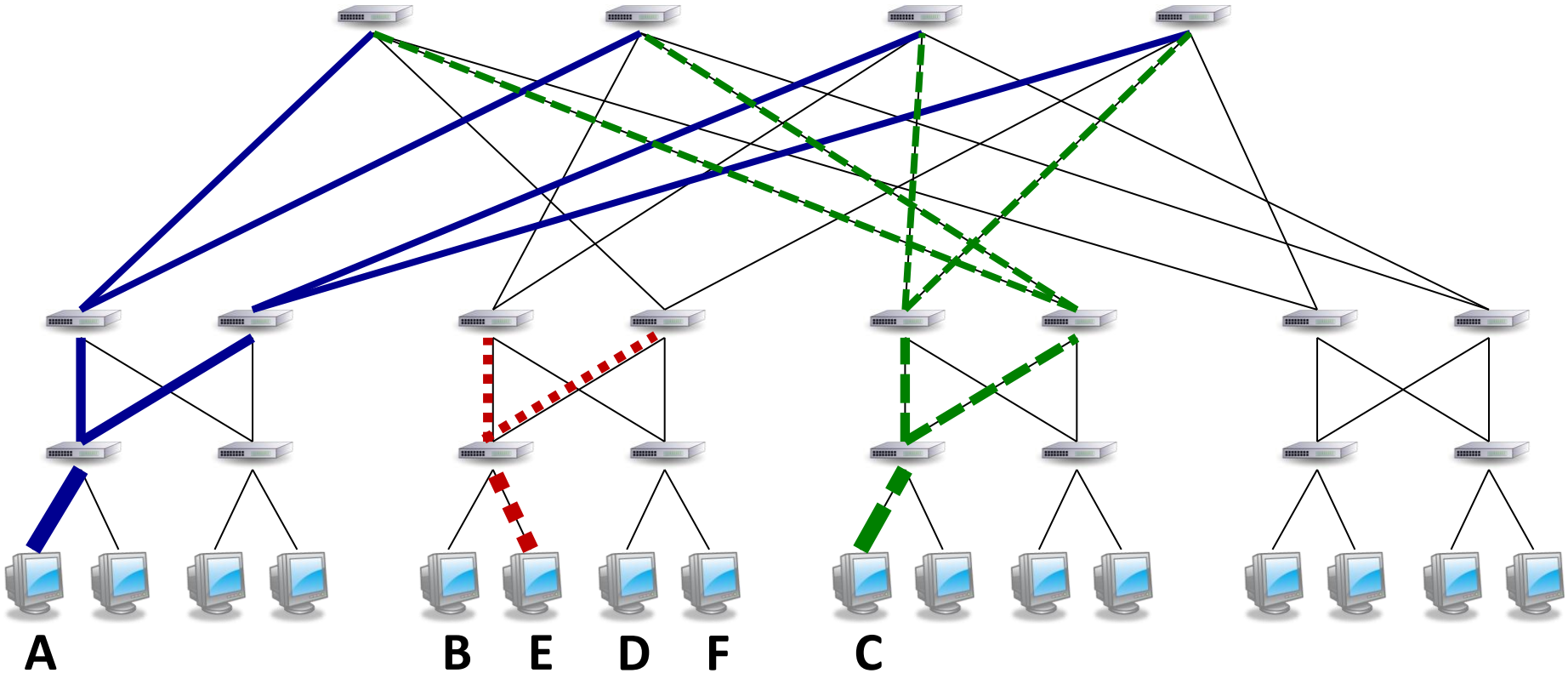
Proactive splitting



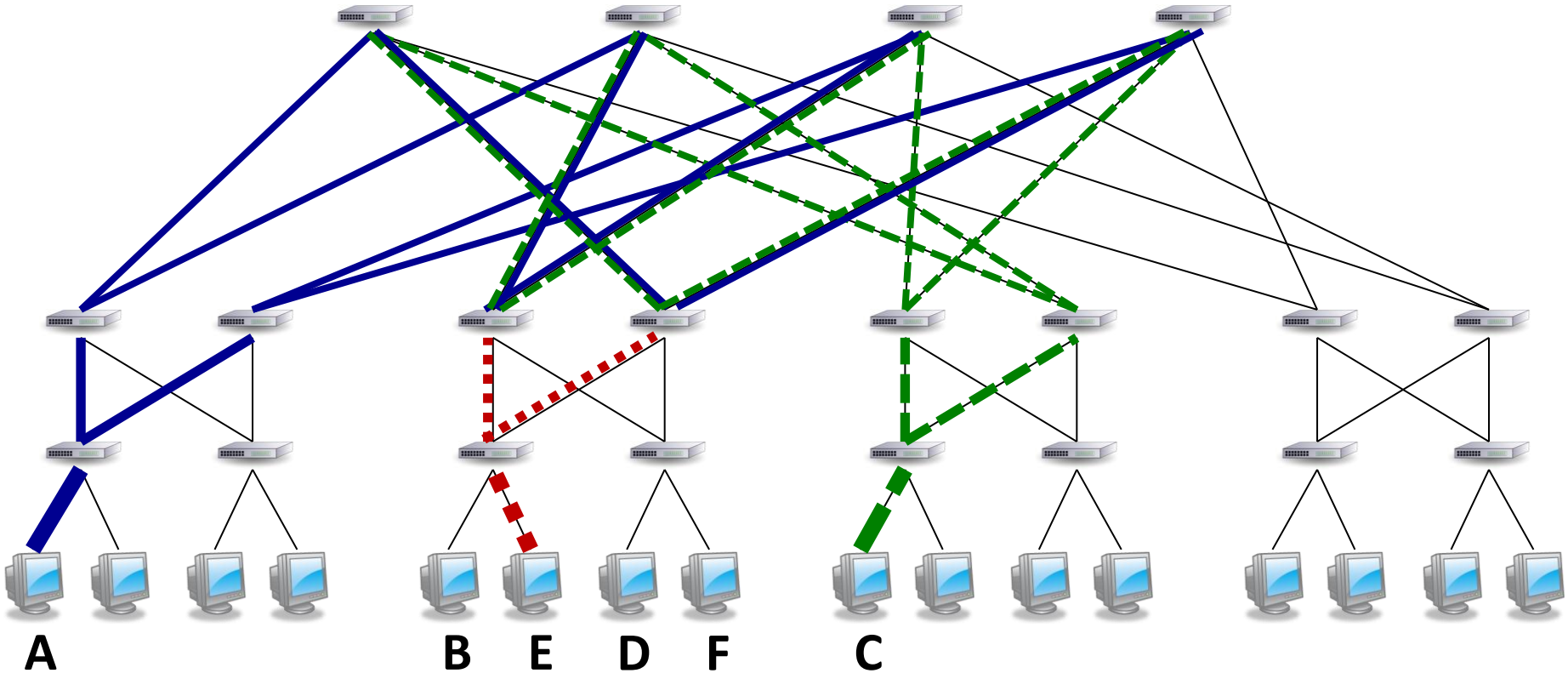
Proactive splitting



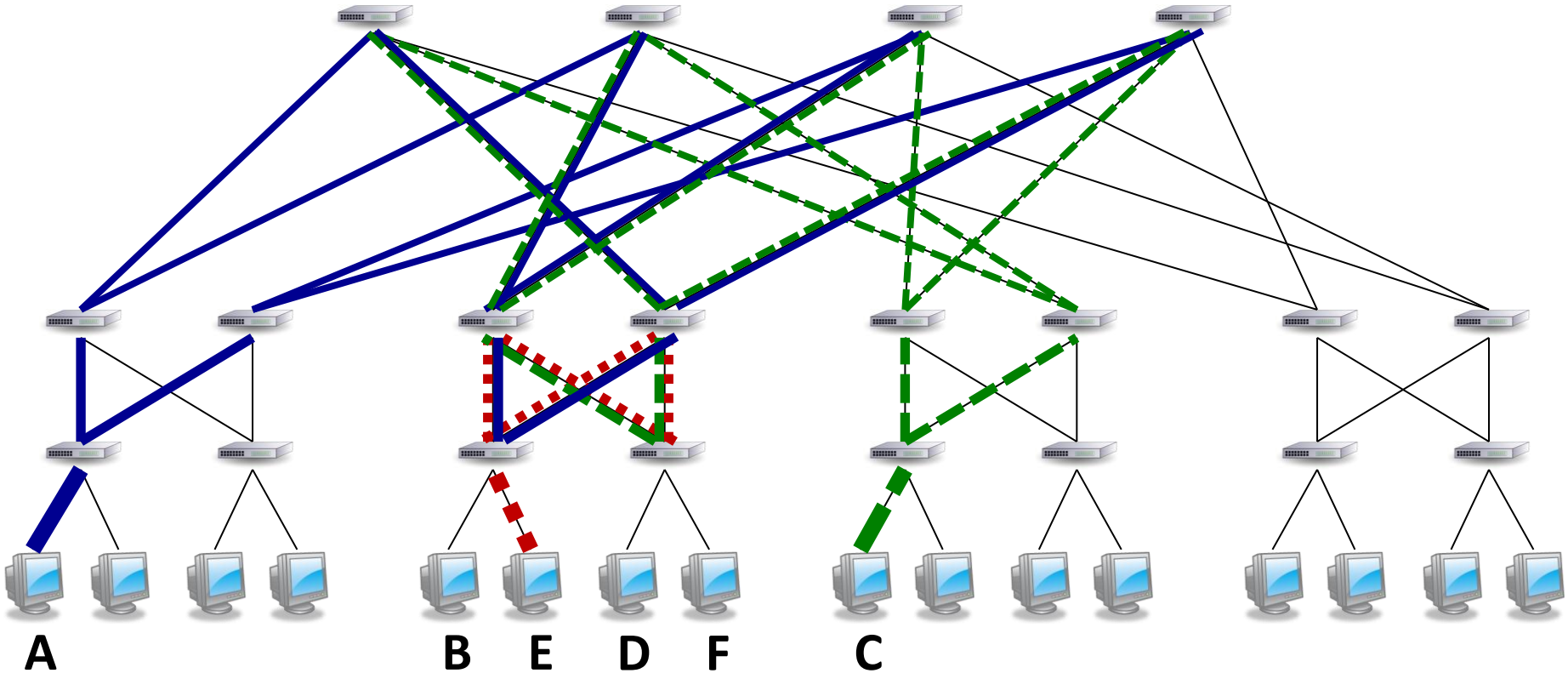
Proactive splitting



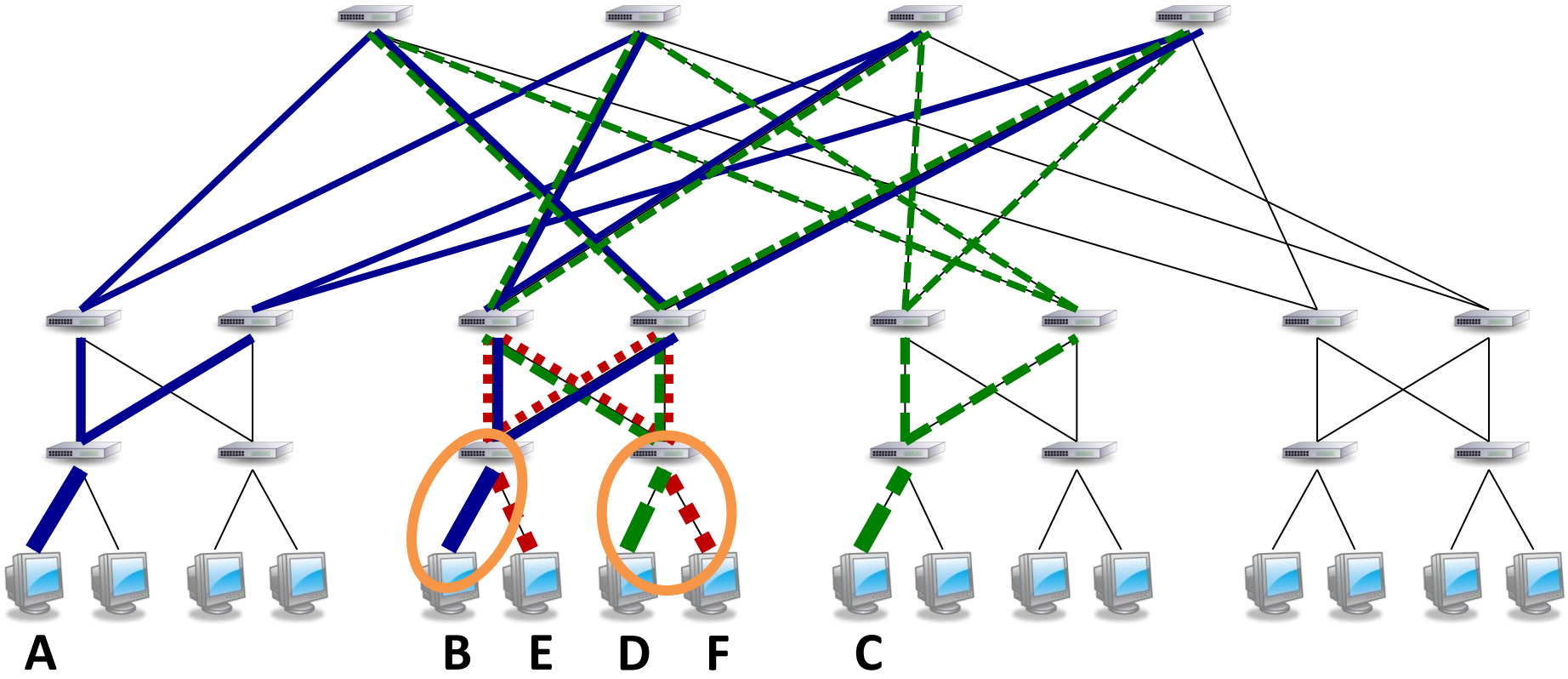
Proactive splitting



Proactive splitting



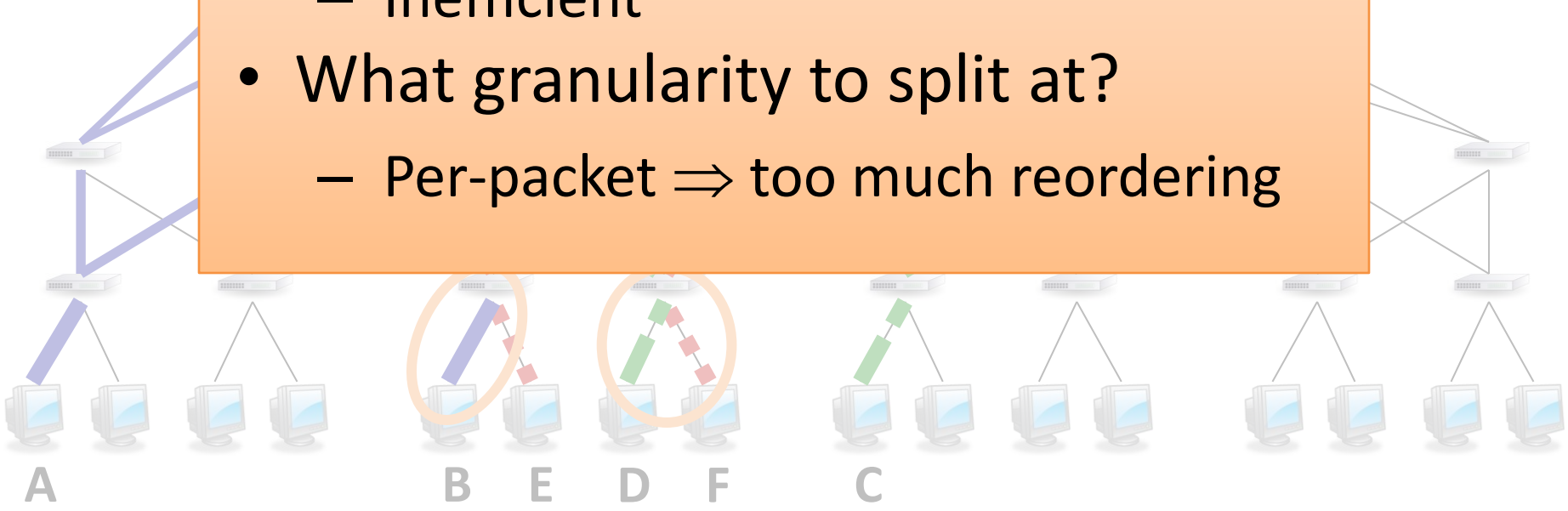
Proactive splitting



Proactive splitting

Problems:

- Split every flow?
  - Inefficient
- What granularity to split at?
  - Per-packet ⇒ too much reordering



Problems

1. Dynamic workloads

2. Splitting flows

Solutions

1. Routers Plus Preprocessing (RPP) model
   - Poly-time preprocessing is free
   - In-band messages are free

2. Splitting technique
   - Group flows by target, split aggregate flow
   - Group contiguous packets into ***flowlets*** to reduce reordering

Problems

1. Dynamic workloads

2. Splitting flows

3. Switch ≠ end host
   - Limited processing, high-speed matching on packet headers

Solutions

1. Routers Plus Preprocessing (RPP) model
   - Poly-time preprocessing is free
   - In-band messages are free

2. Splitting technique
   - Group flows by target, split aggregate flow
   - Group contiguous packets into ***flowlets*** to reduce reordering

3. Add forwarding table rules to programmable switches
   - Match TCP seq num header, use bit tricks to create flowlets

Conclusion

- Both theoretical and practical innovations needed to bridge theory-practice gap

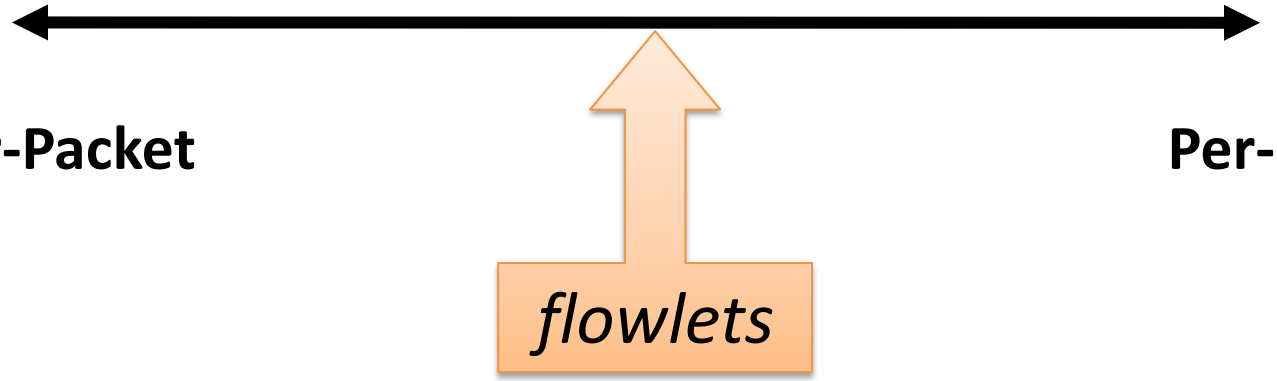
- LOCALFLOW: optimal algorithm in new framework for data center networks

Additional slides

Granularity of splitting

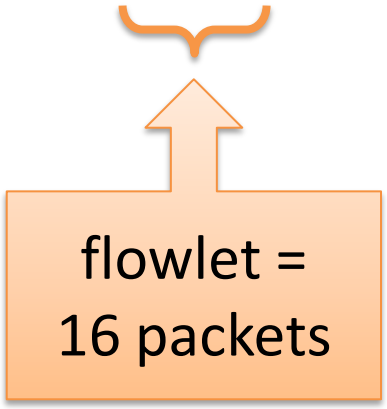
Optimal routing
High reordering

Suboptimal routing
Low reordering

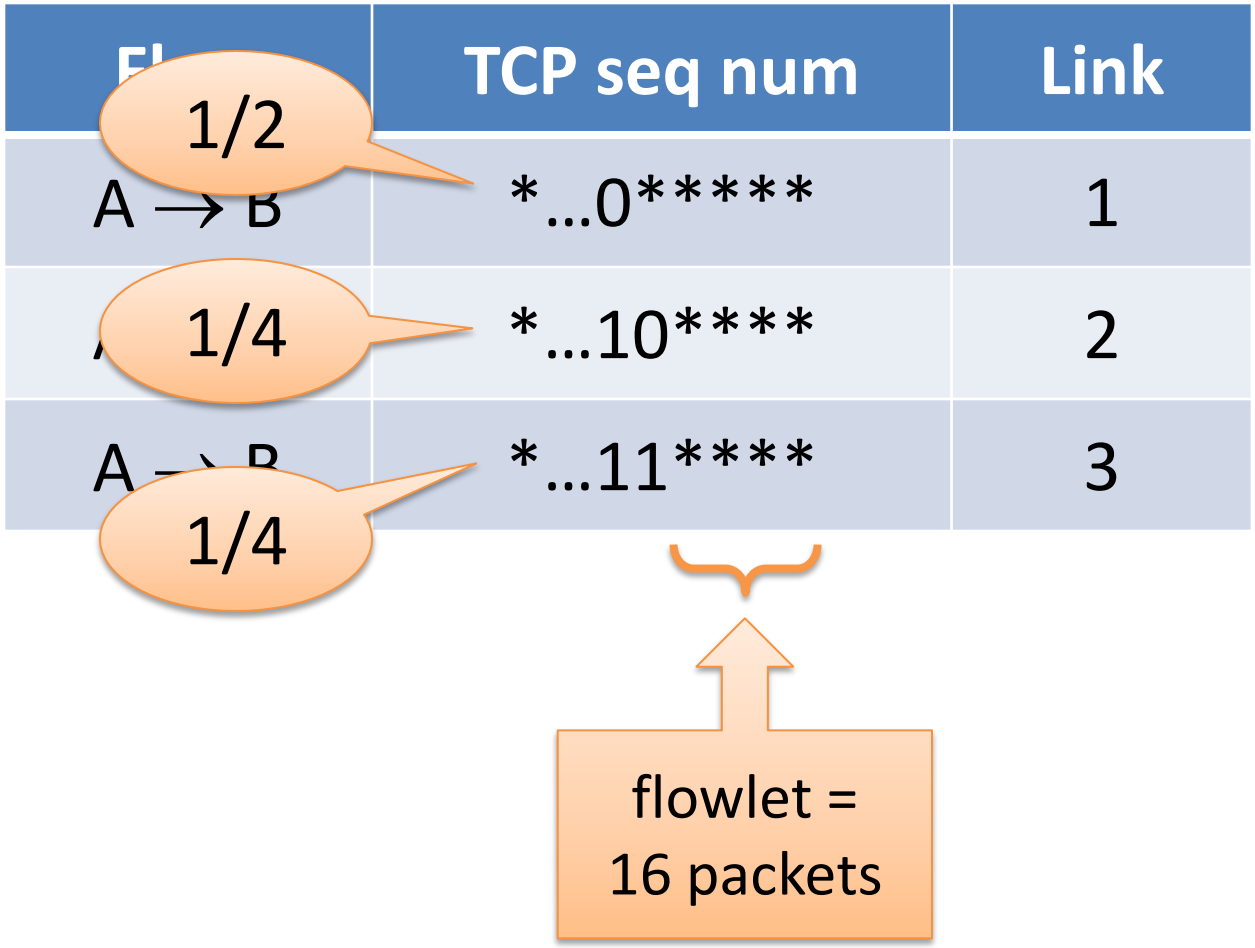


Line rate splitting

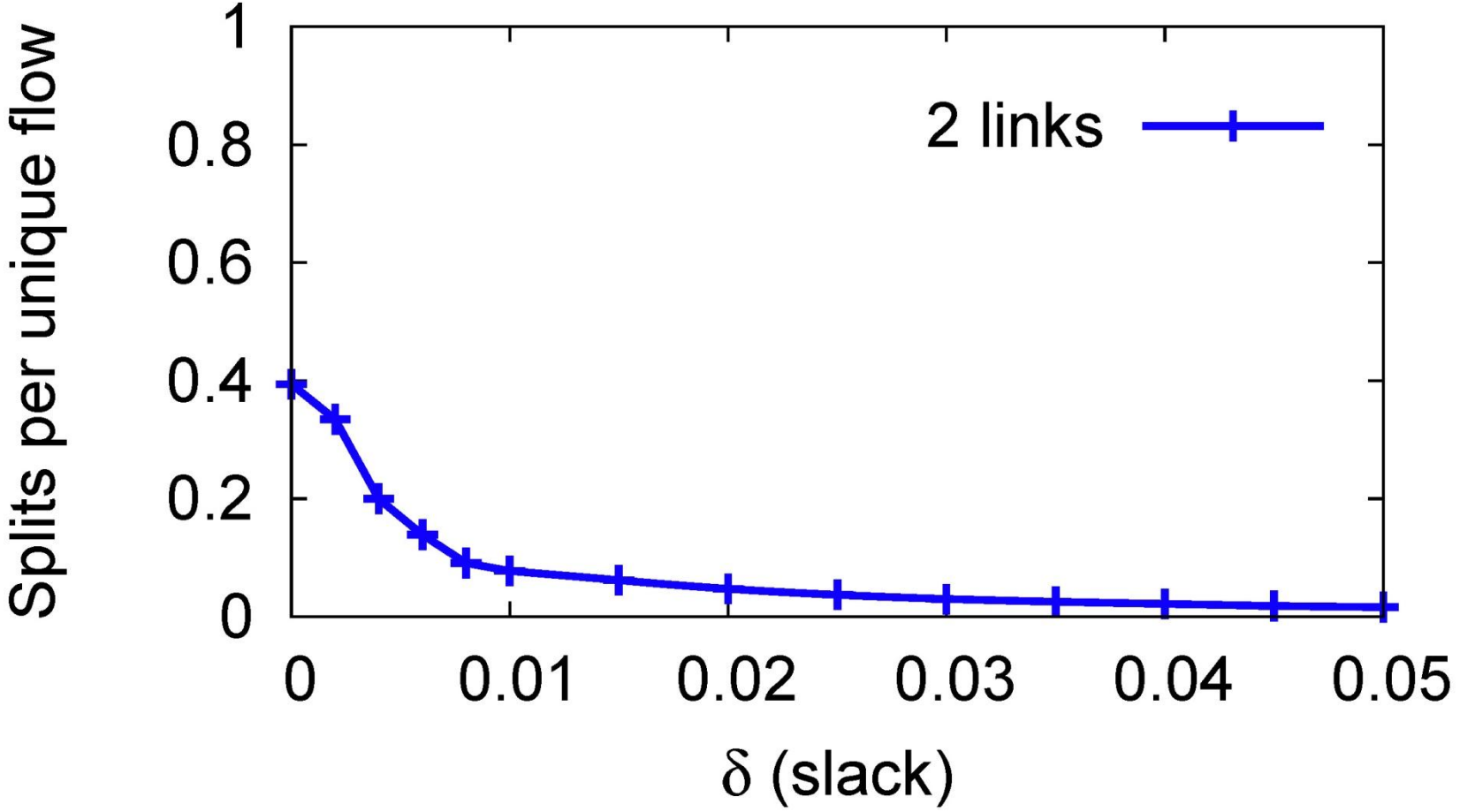
Flow	TCP seq num	Link
A → B	*…0*****	1
A → B	*…10****	2
A → B	*…11****	3



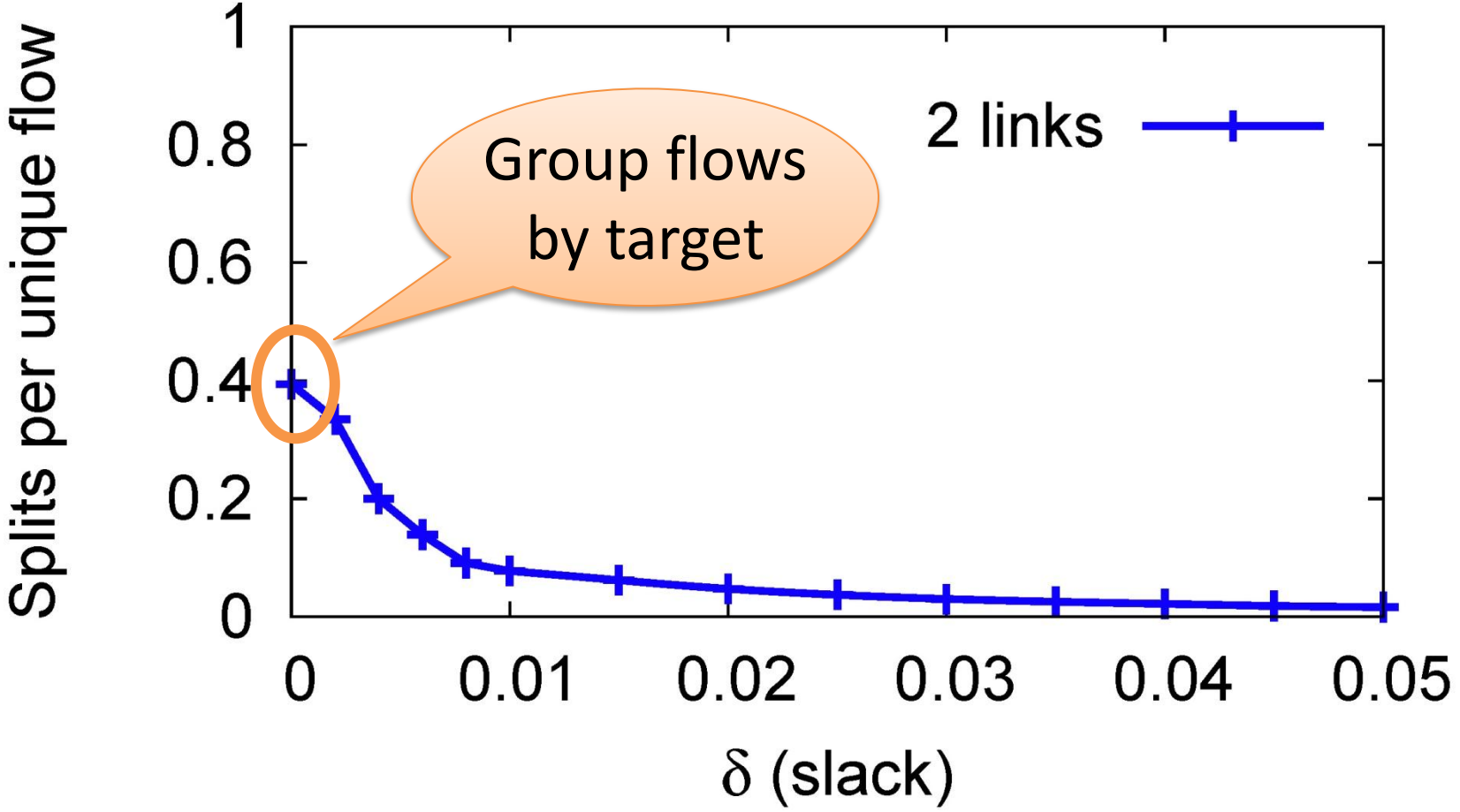
Line rate splitting



LOCALFLOW: Frequency of splitting



LOCALFLOW: Frequency of splitting



LOCALFLOW: Frequency of splitting