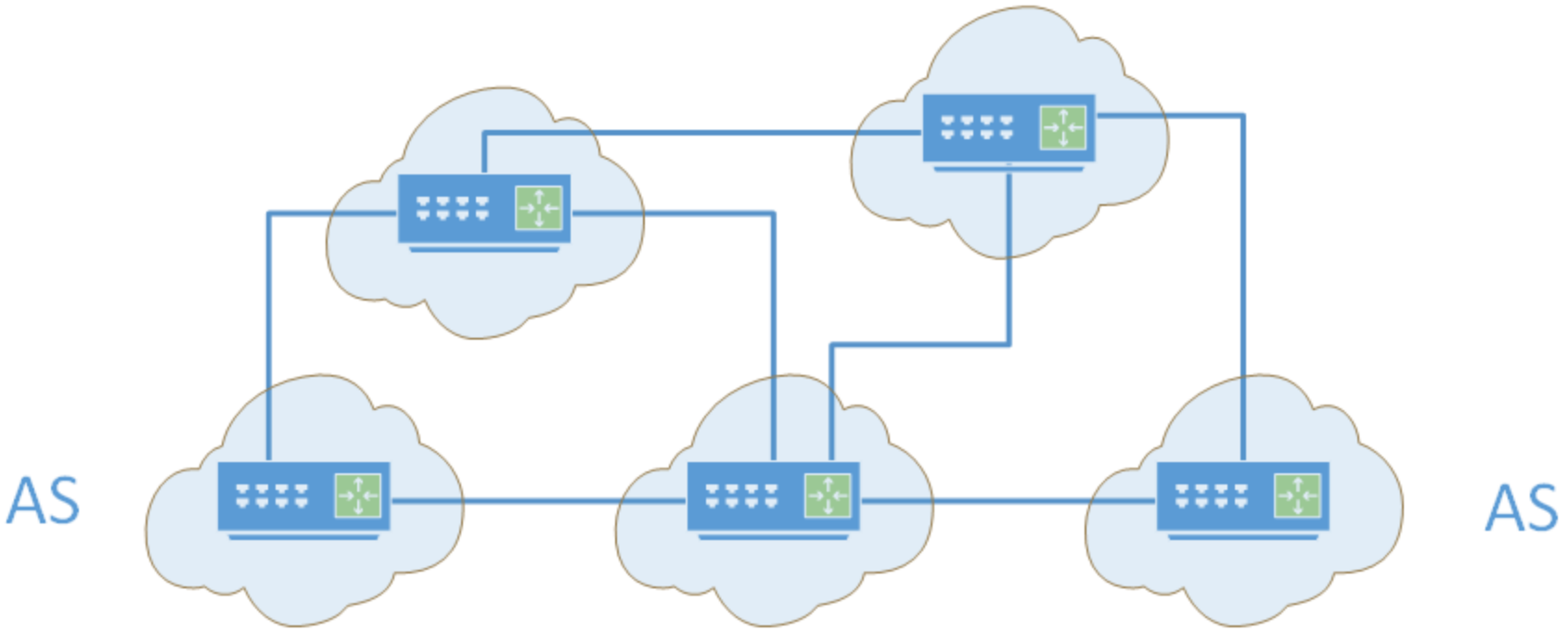
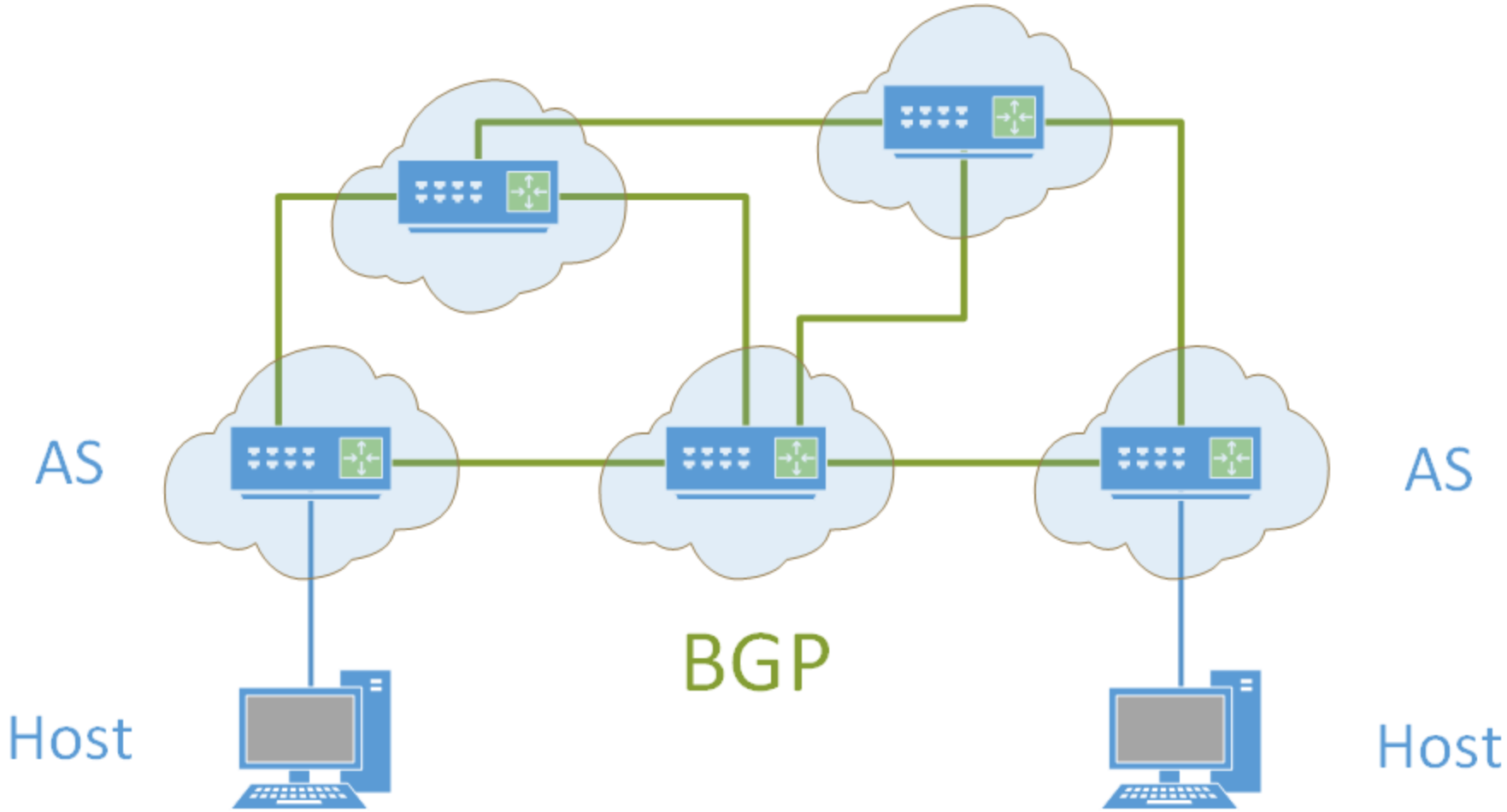
Stitching Inter-Domain Paths over IXPs

Vasileios Kotronis, Rowan Klöti, **Matthias Rost**, Panagiotis Georgopoulos, Bernhard Ager, Stefan Schmid, Xenofontas Dimitropoulos

Inter-Domain Routing: Status Quo

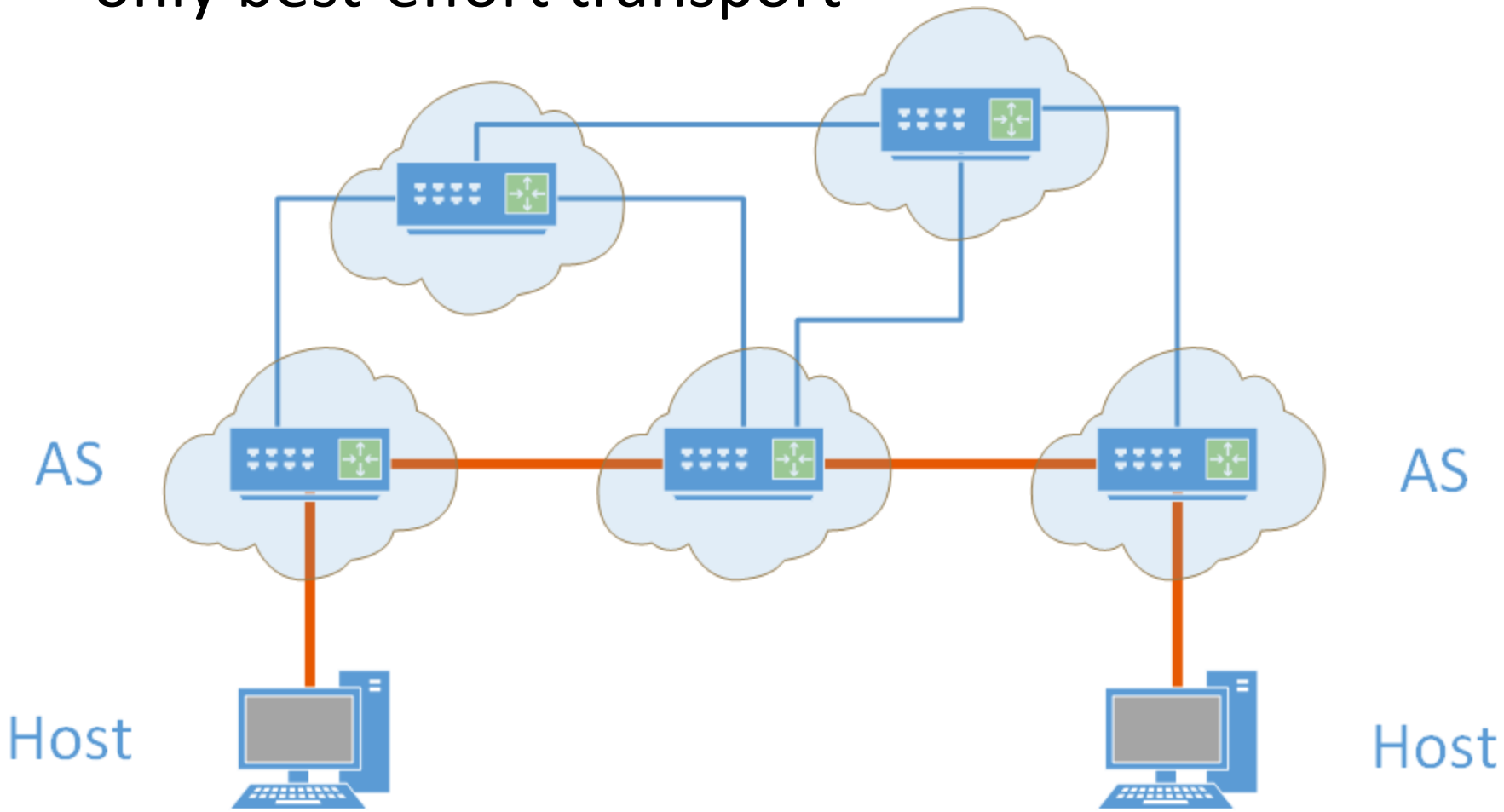


Inter-Domain Routing: Status Quo



Inter-Domain Routing: Status Quo

- BGP selects single policy-compliant path
- only best-effort transport



Motivation

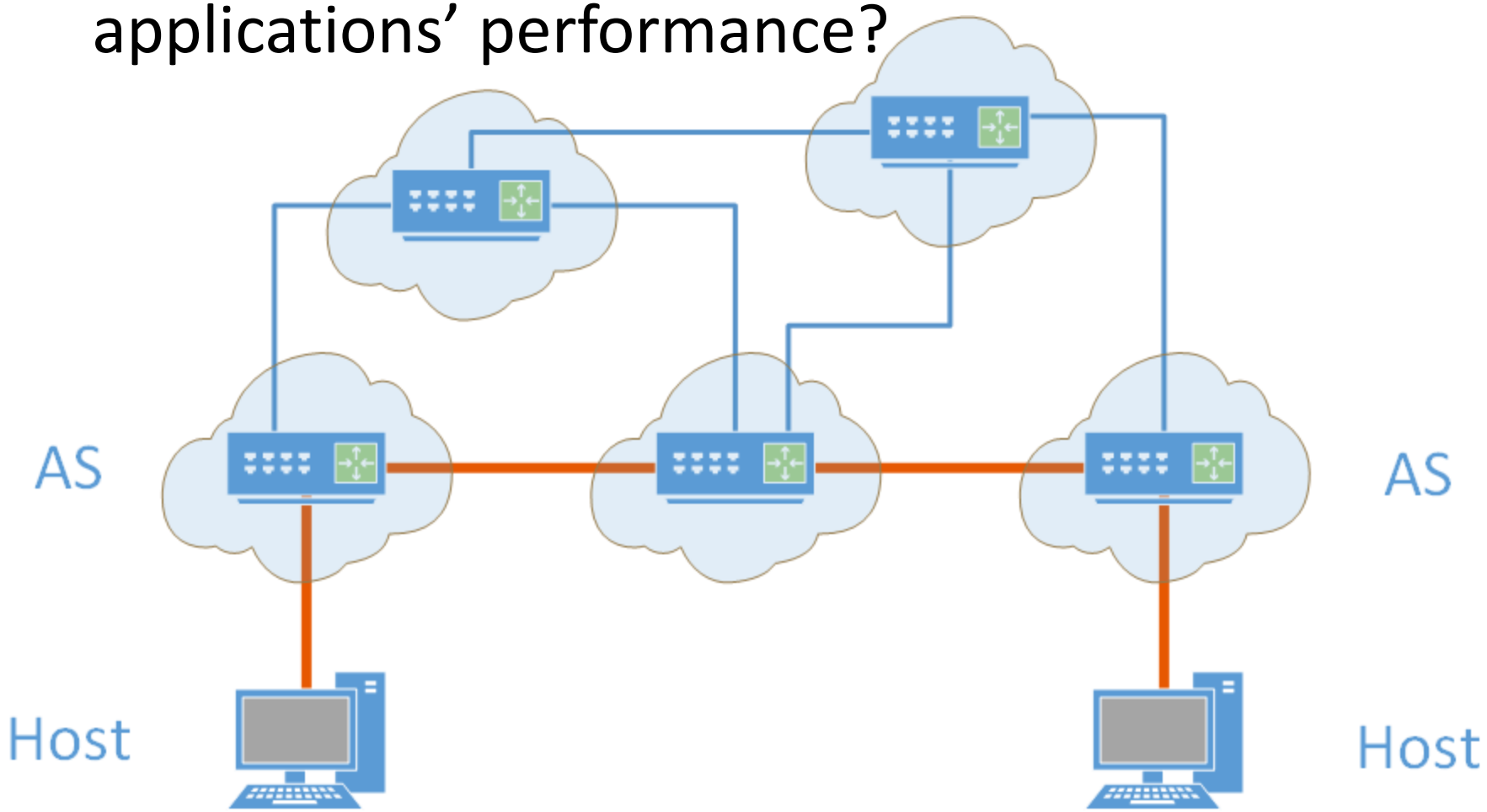
- Applications may require …
  - high bandwidth
  - low latency
  - enhanced reliability

usignite

mozilla IGNITE

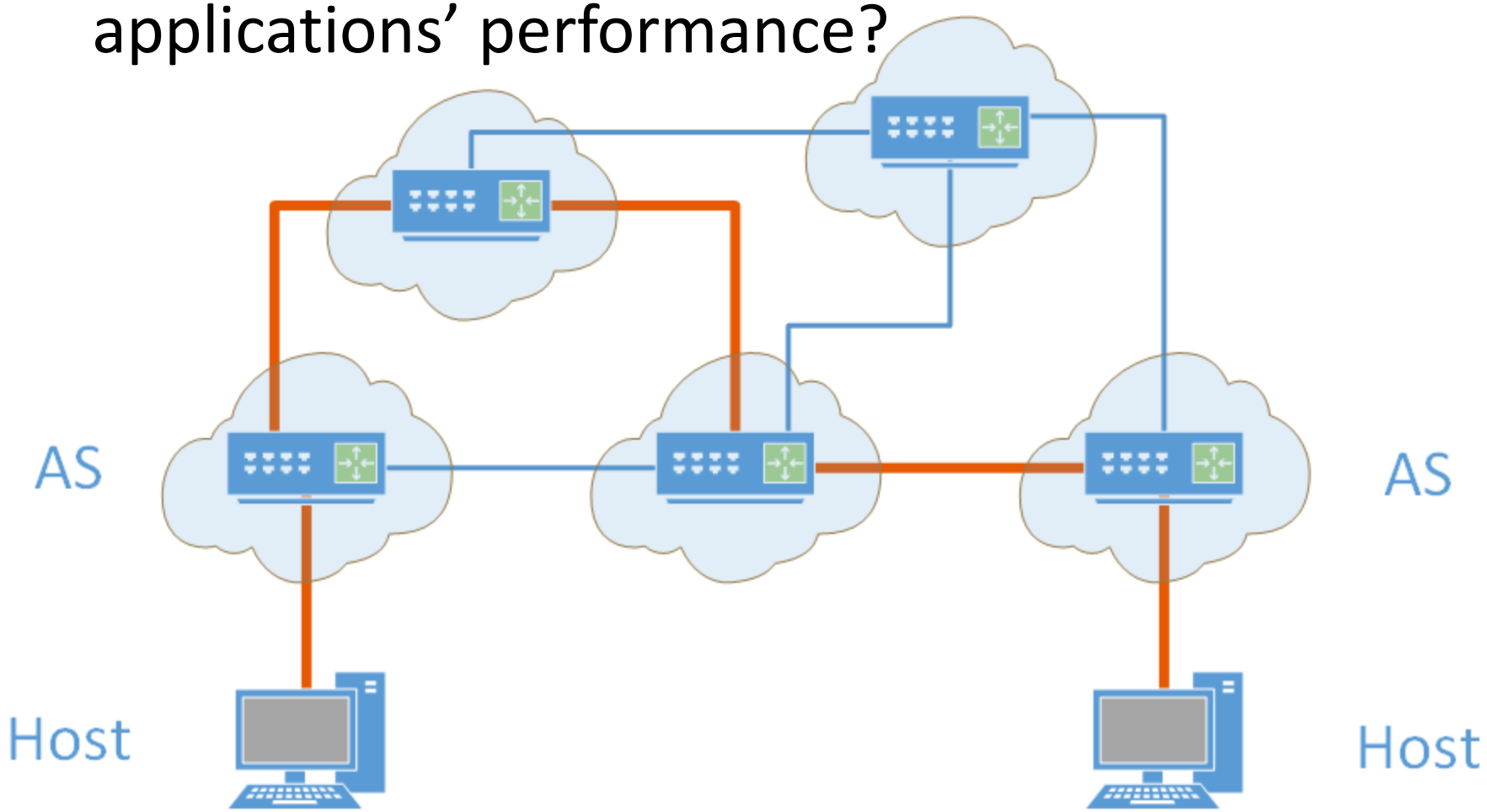
Inter-Domain Routing: Problem Statement

- How can we harness path diversity to improve applications' performance?



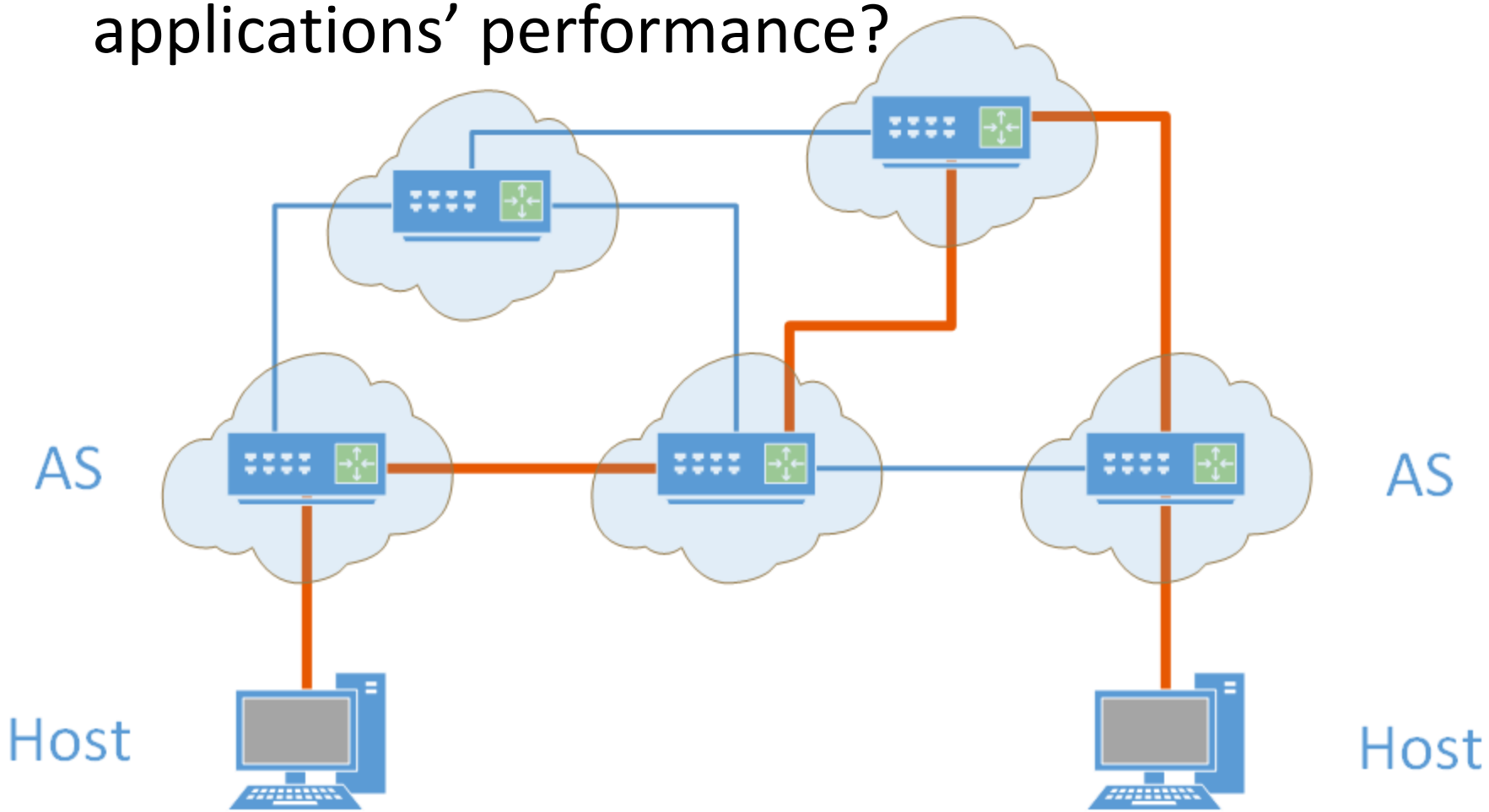
Inter-Domain Routing: Problem Statement

- How can we harness path diversity to improve applications' performance?



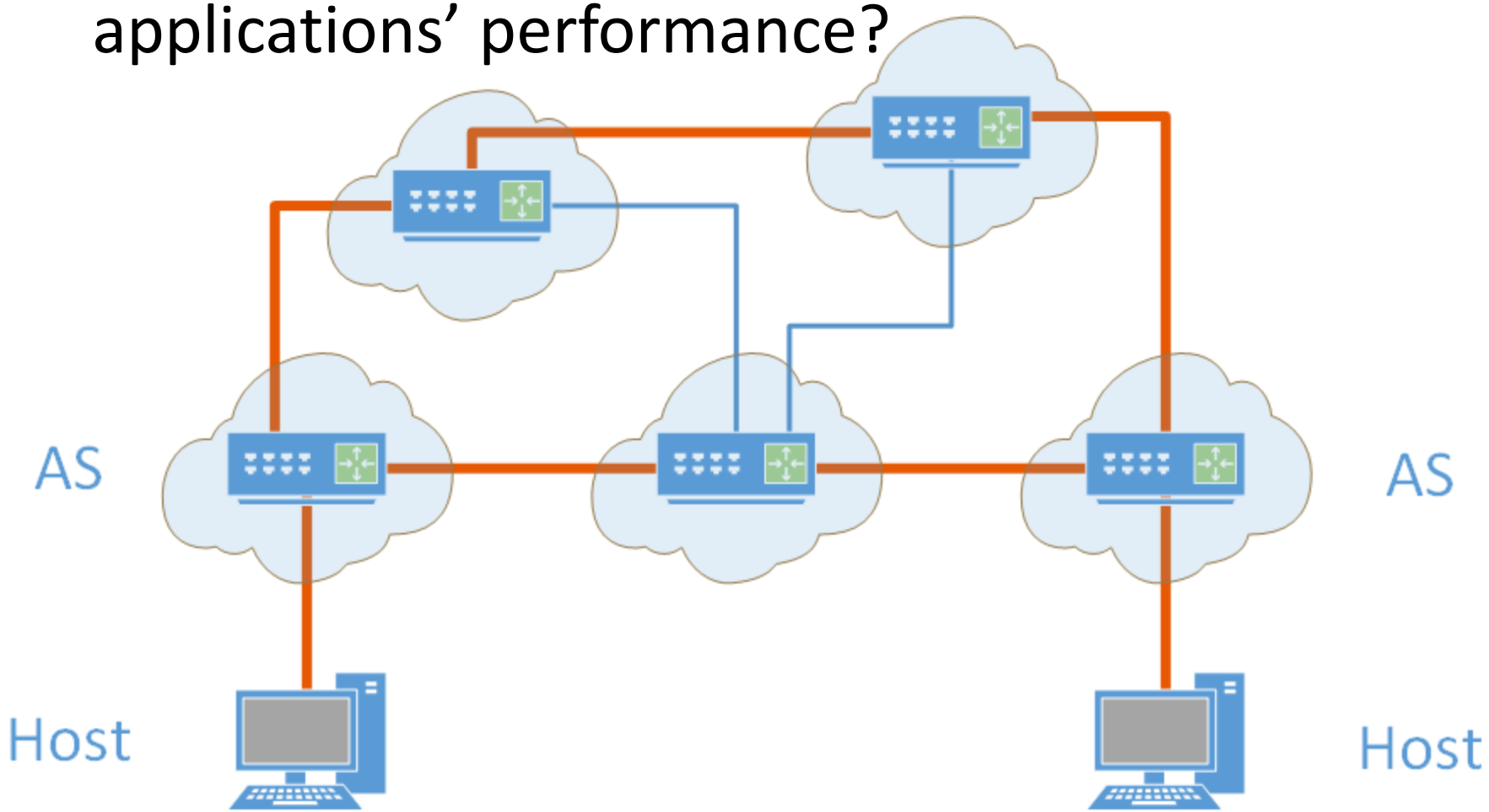
Inter-Domain Routing: Problem Statement

- How can we harness path diversity to improve applications' performance?



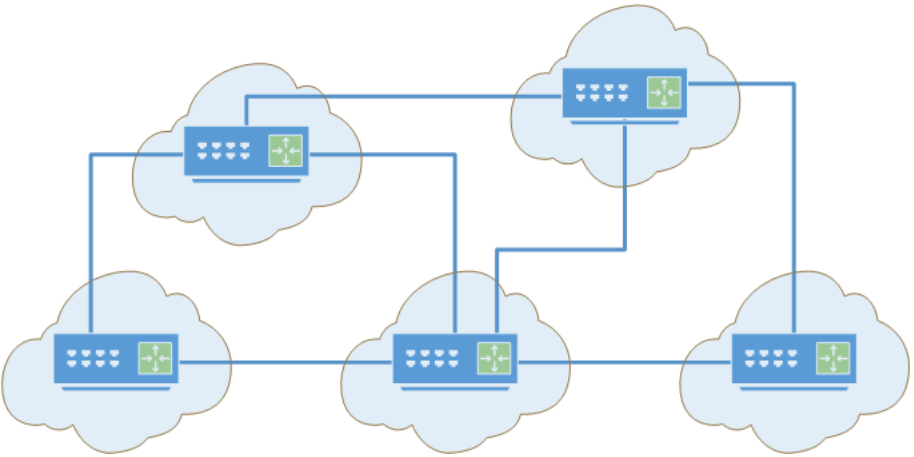
Inter-Domain Routing: Problem Statement

- How can we harness path diversity to improve applications' performance?

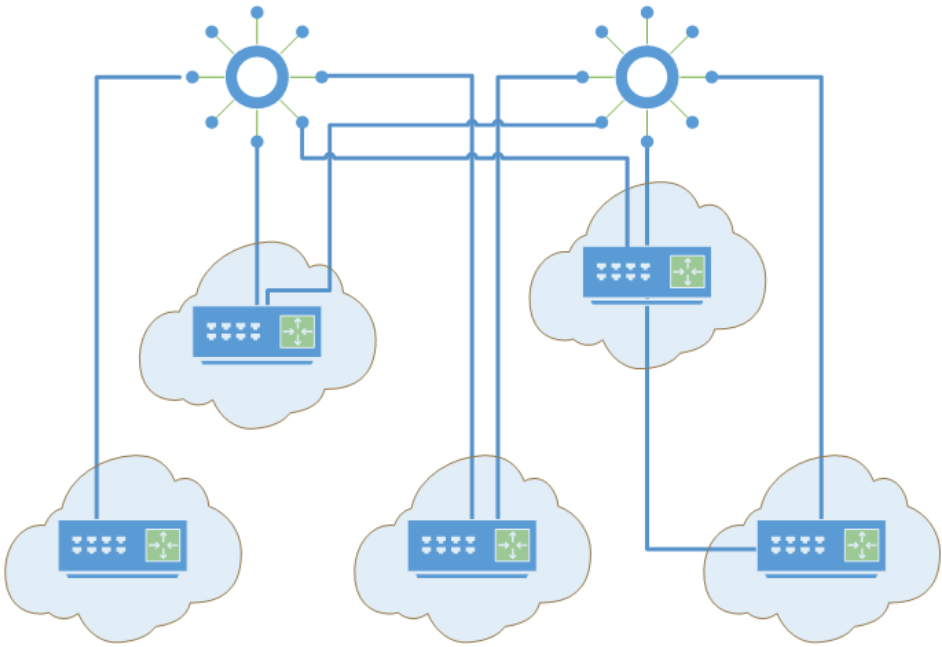


Our Proposal: Stitching Paths at IXPs

- ASes connect at Internet Exchange Points (IXPs)



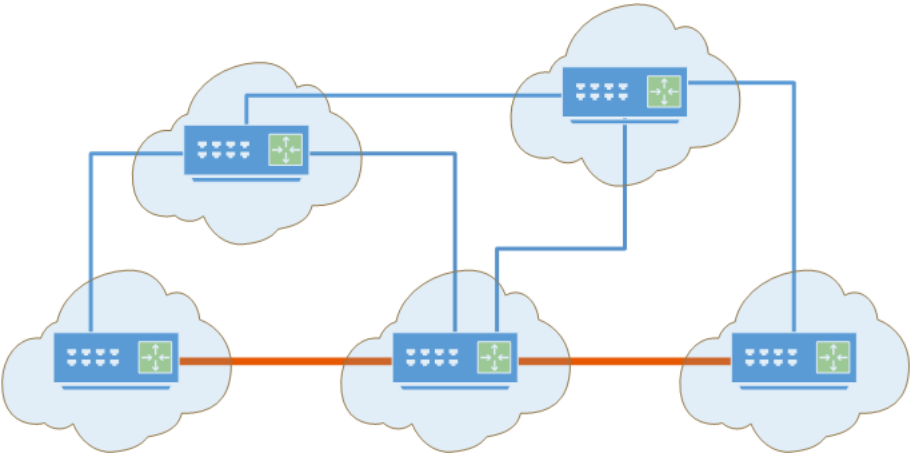
AS-centric view



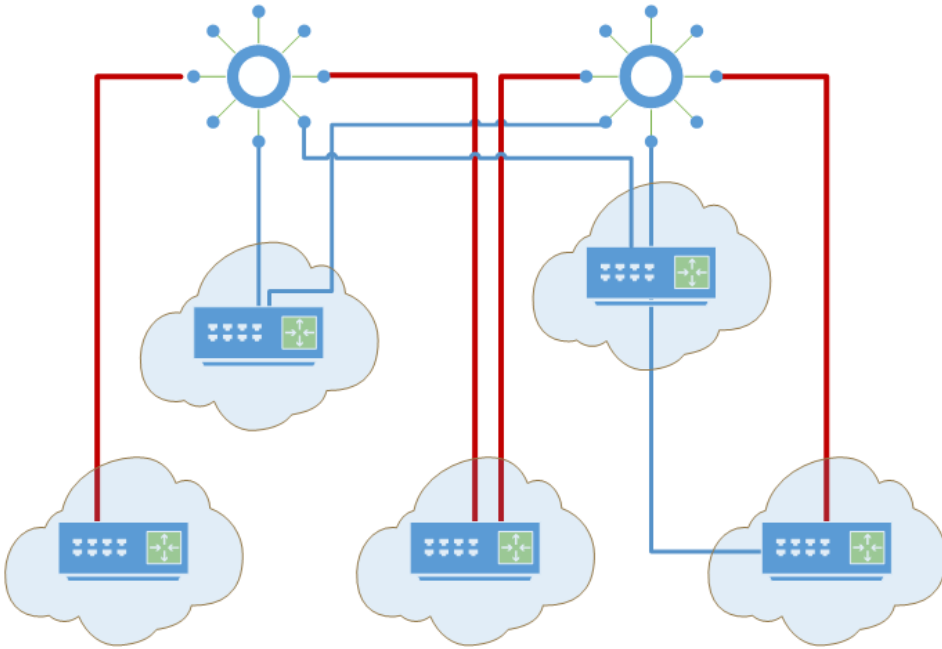
IXP-centric view

Our Proposal: Stitching Paths at IXPs

- ASes connect at Internet Exchange Points (IXPs)



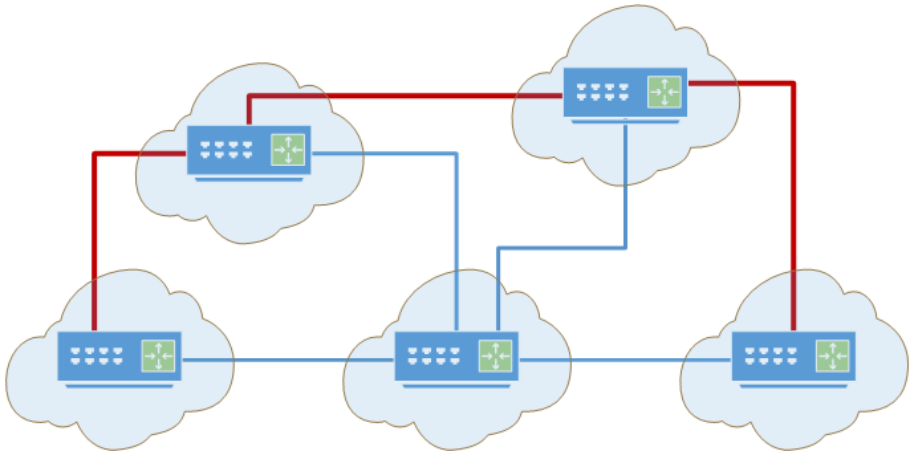
AS-centric view



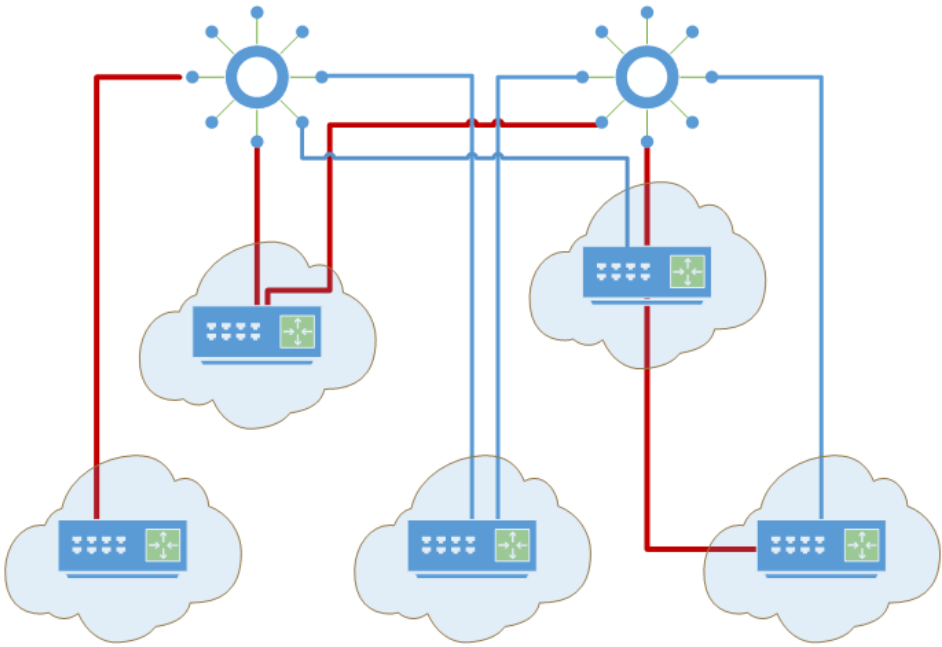
IXP-centric view

Our Proposal: Stitching Paths at IXPs

- ASes connect at Internet Exchange Points (IXPs)



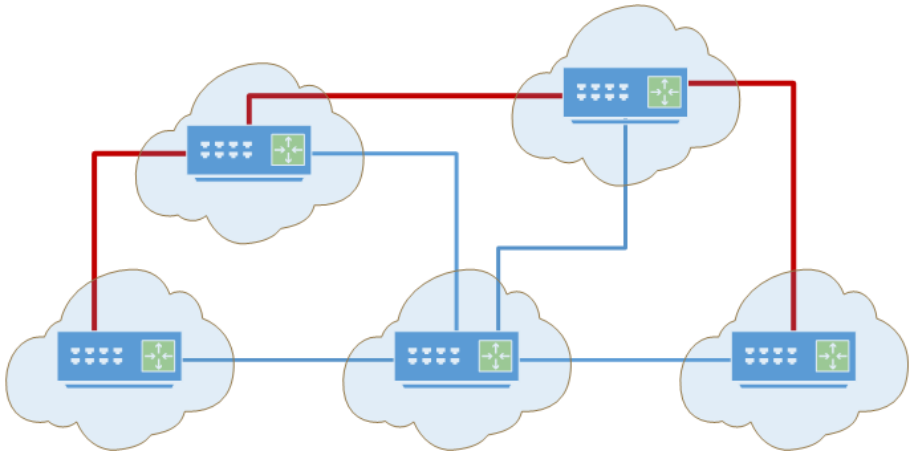
AS-centric view



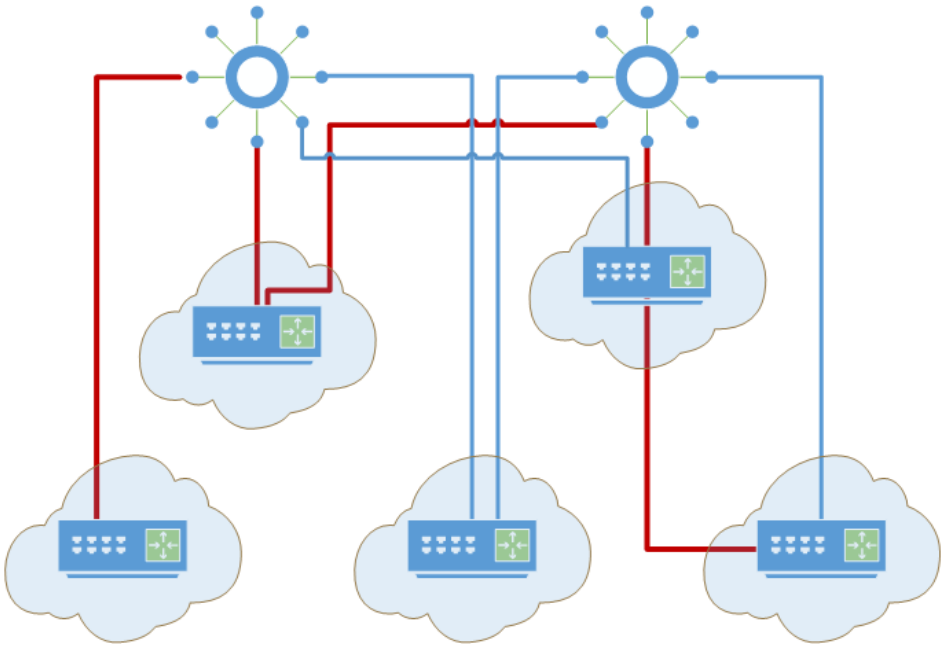
IXP-centric view

Our Proposal: Stitching Paths at IXPs

- ASes connect at Internet Exchange Points (IXPs)



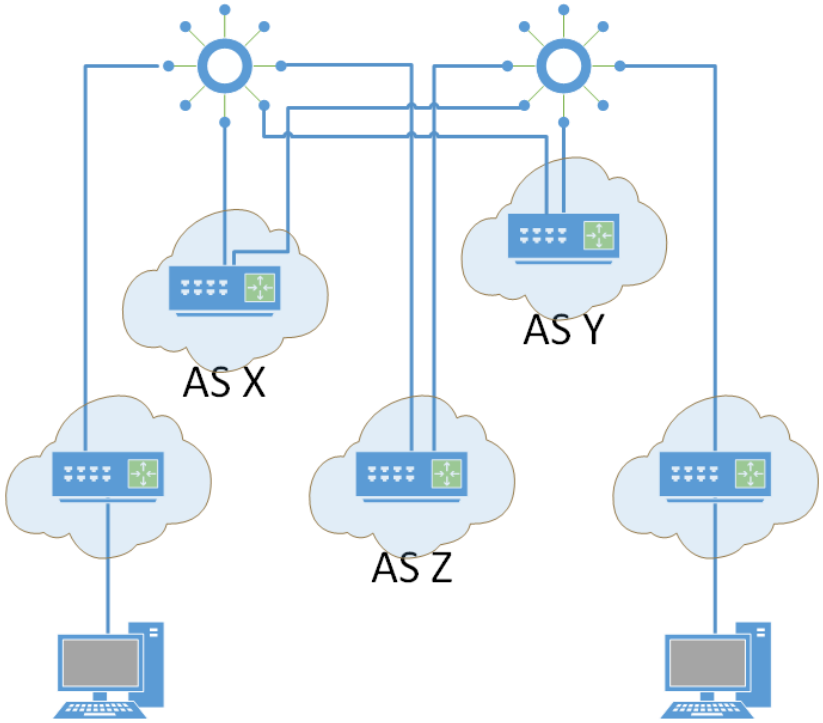
AS-centric view

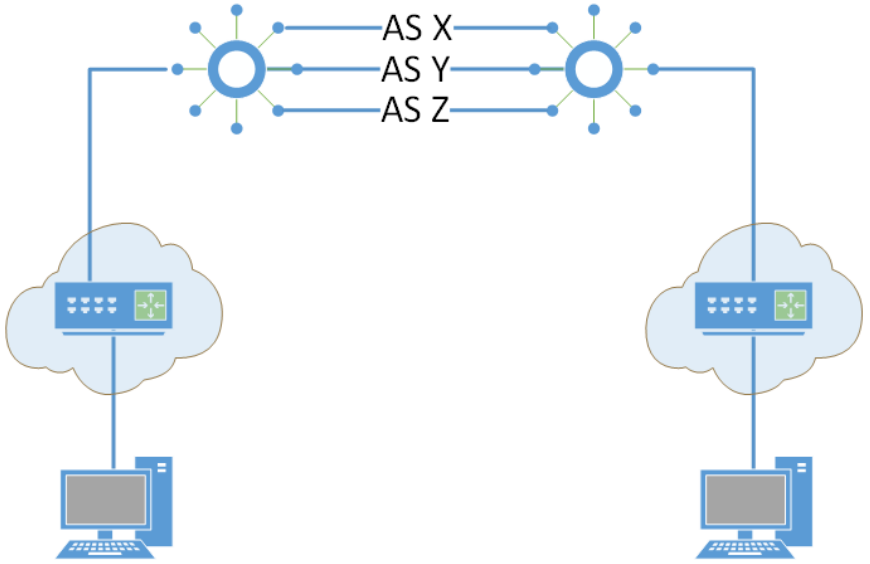


IXP-centric view

Our Proposal: Stitching Paths at IXPs

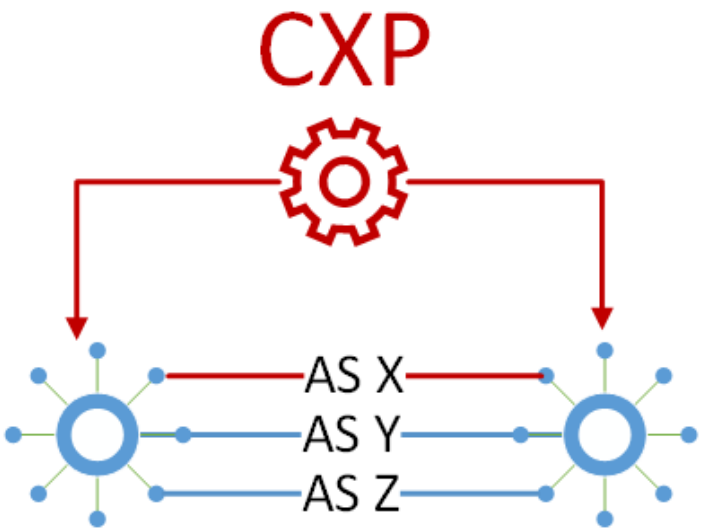
- ASes connect at Internet Exchange Points (IXPs)
- Idea: use ASes for providing inter-IXP paths





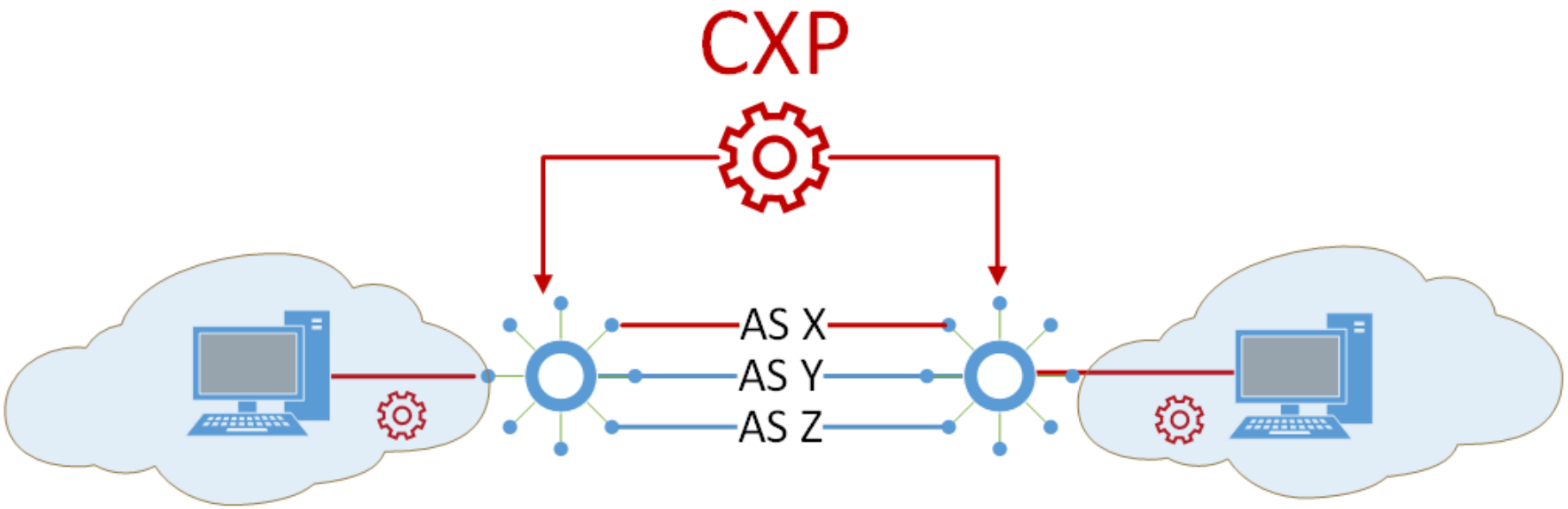
Our Proposal: Control Exchange Points

- **Centrally** stitch inter-IXP paths at IXPs using CXPs



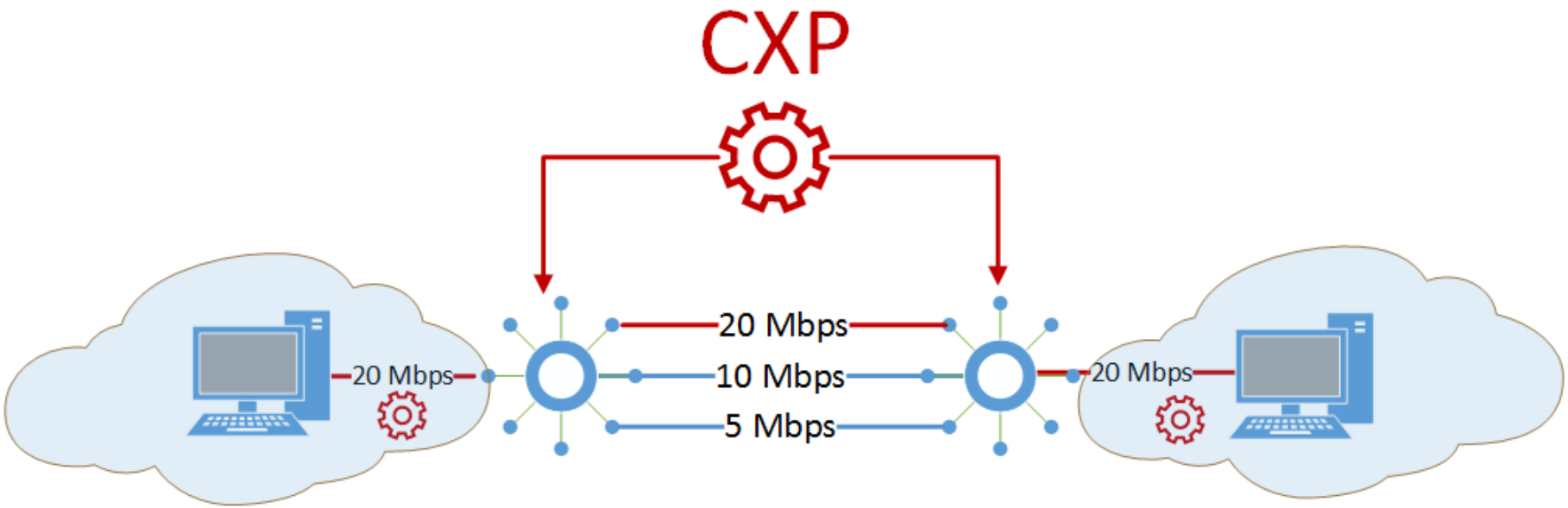
Our Proposal: Control Exchange Points

- **Centrally** stitch inter-IXP paths at IXPs using CXPs
- ASes are responsible to connect end hosts to IXPs



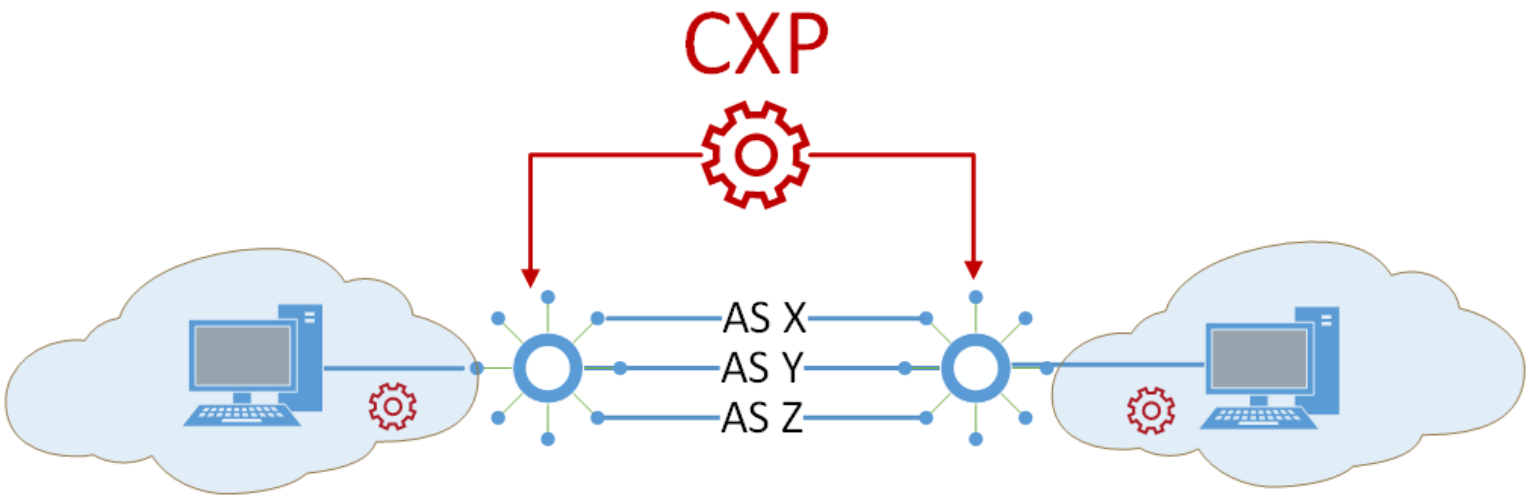
Our Proposal: Control Exchange Points

- **Centrally** stitch inter-IXP paths at IXPs using CXPs
- ASes are responsible to connect end hosts to IXPs
- ASes might provide guarantees on paths
  → end-to-end guarantees



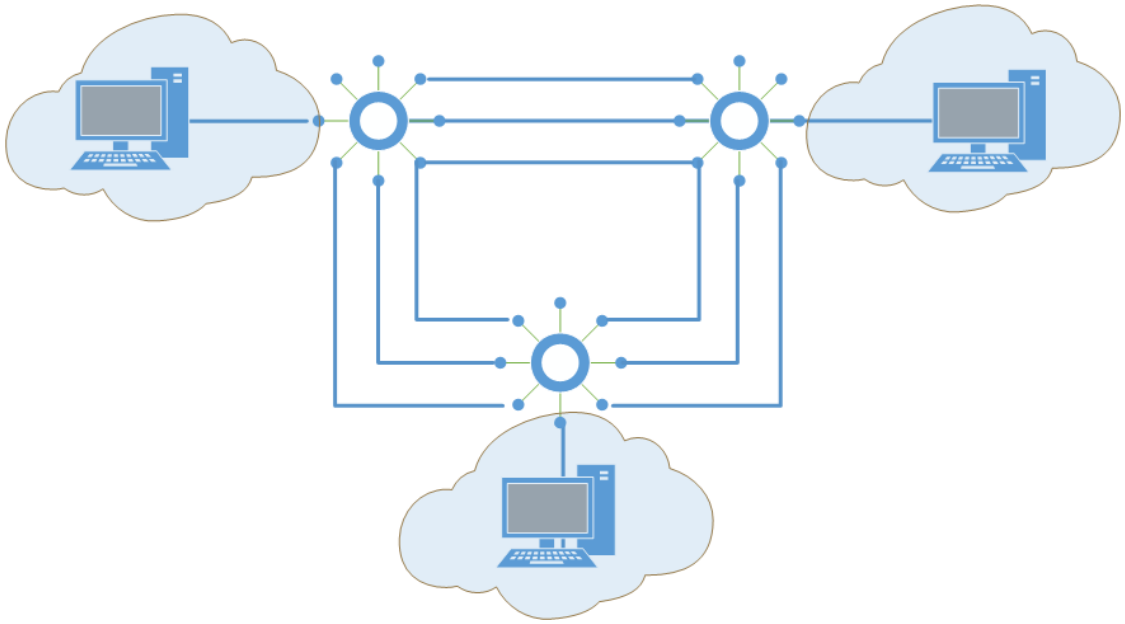
Main Questions

- Which IXPs should be controlled by CXPs?
- How many customers can we reach?
- What is the gain in path diversity?
- How to efficiently and centrally compute routes?
- What are the opportunities of centralized control?



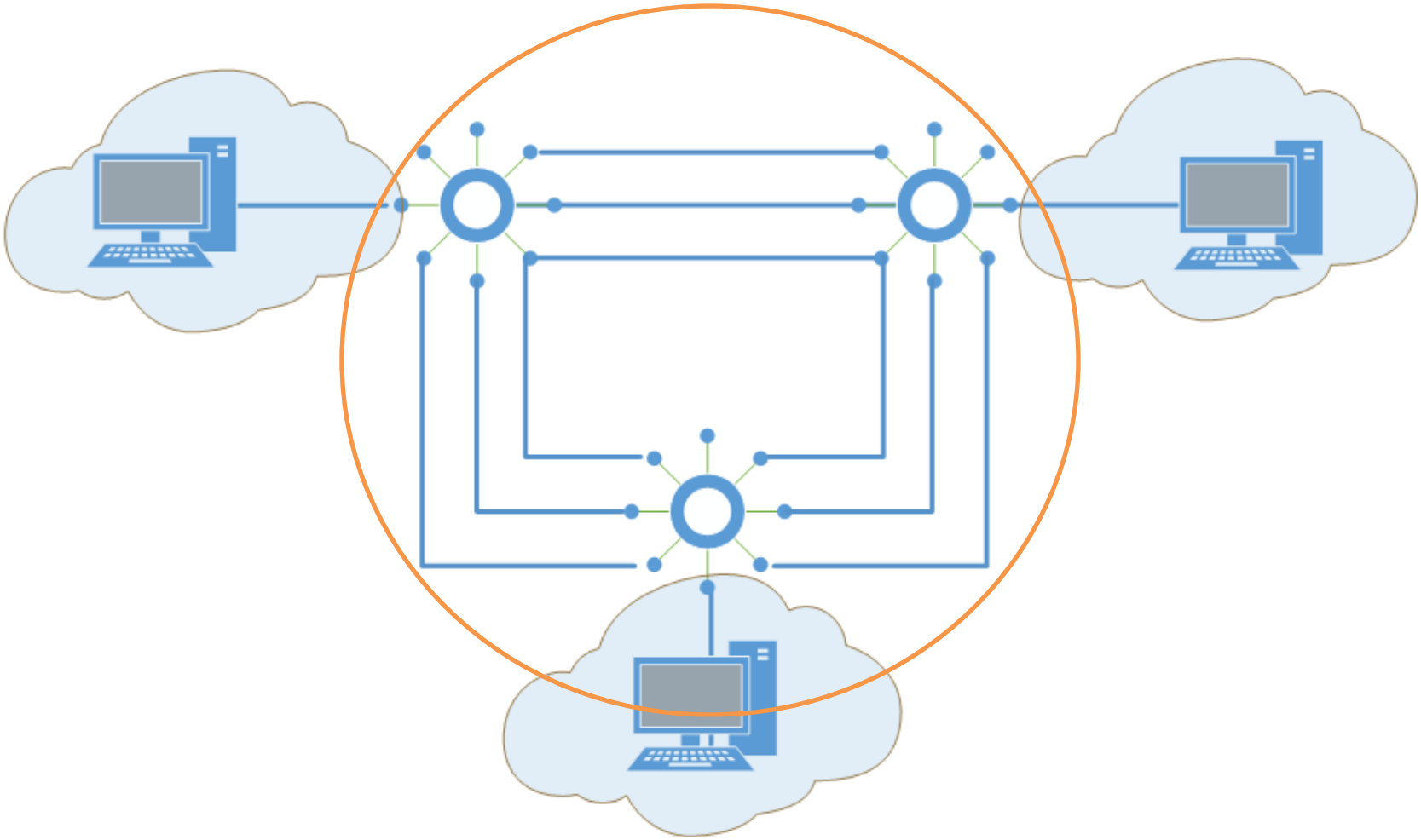
MEASURING THE IXP MULTIGRAPH

Methodology



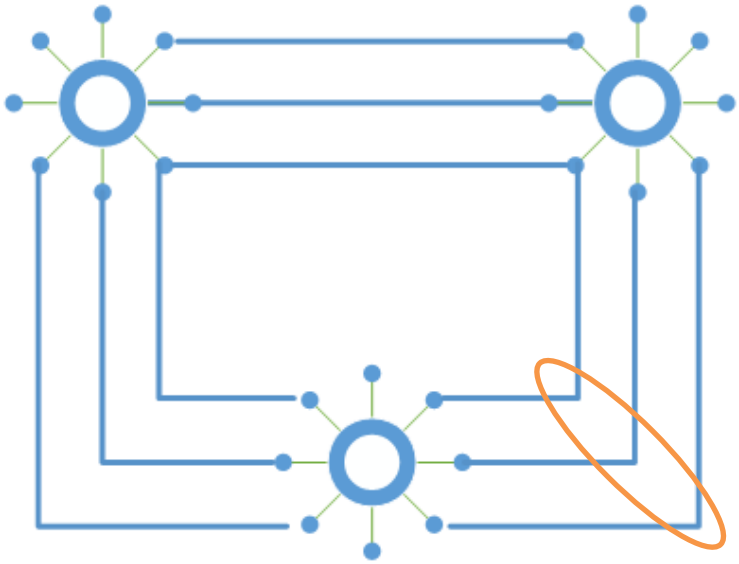
- Determine IXPs and the ASes connecting them
  - Euro-IX (and Peering-DB)
- Determine customer-cone of IXPs
  - CAIDA data

Results at a Glance



Results at a Glance: IXP Multigraph

229 IXPs

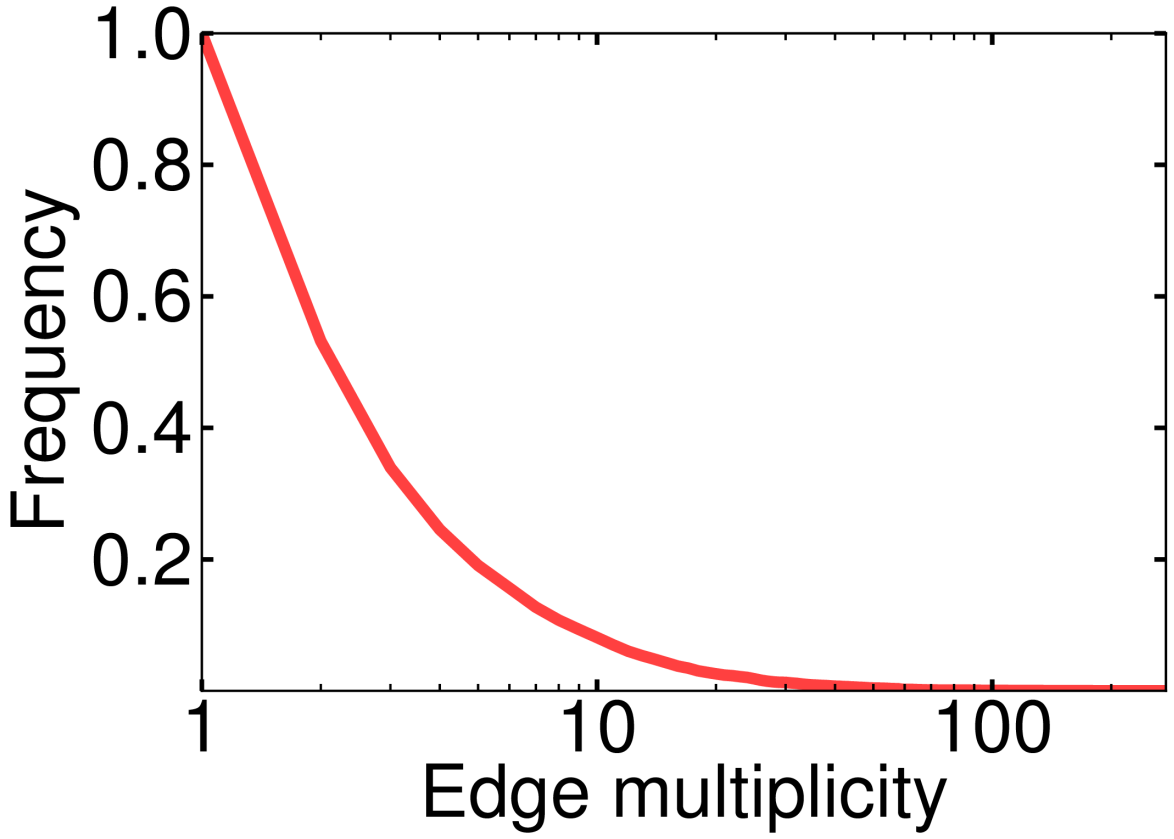


49k edges

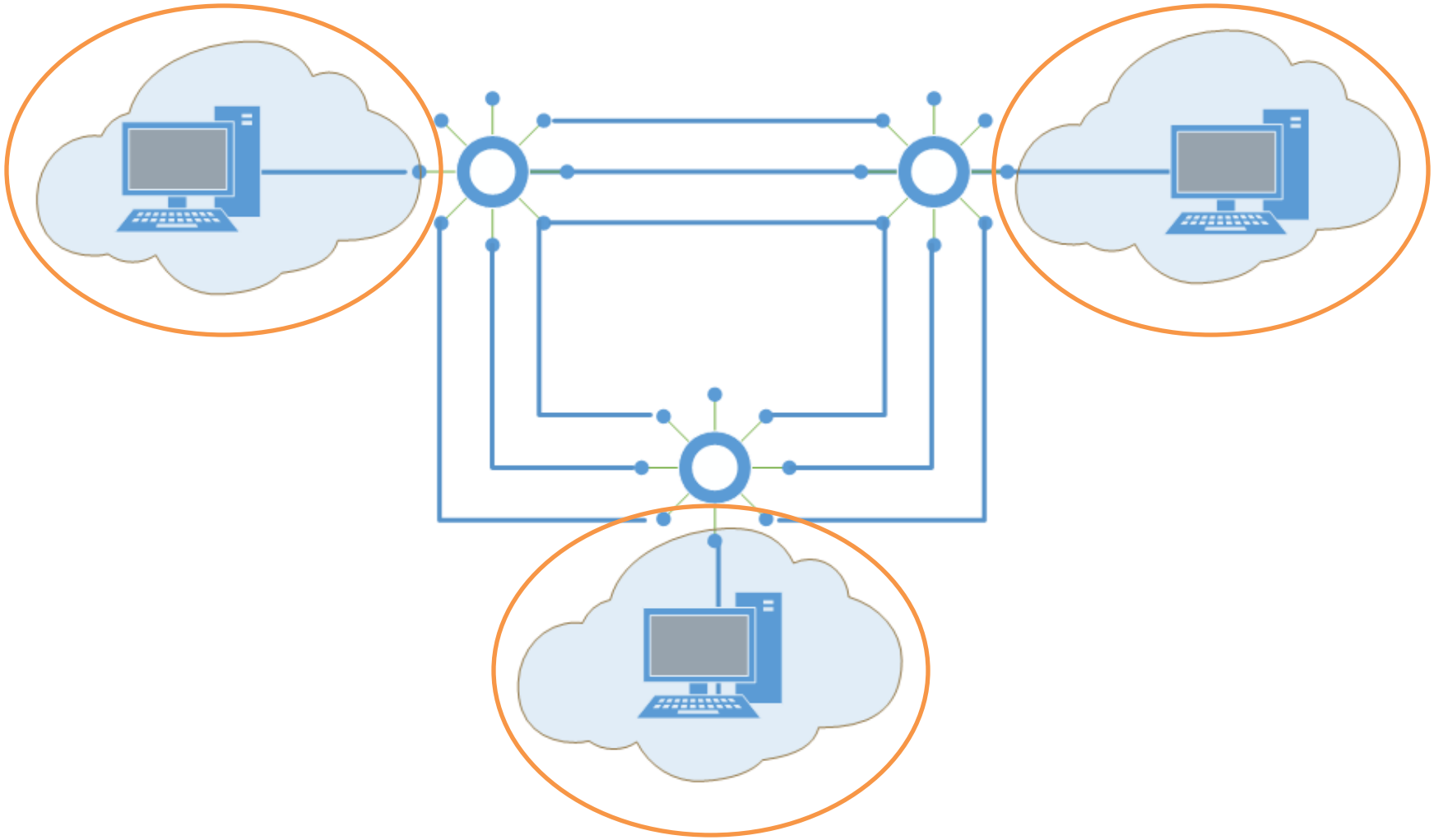
Avg. degree: 220

Avg. edge multiplicity: 4.3

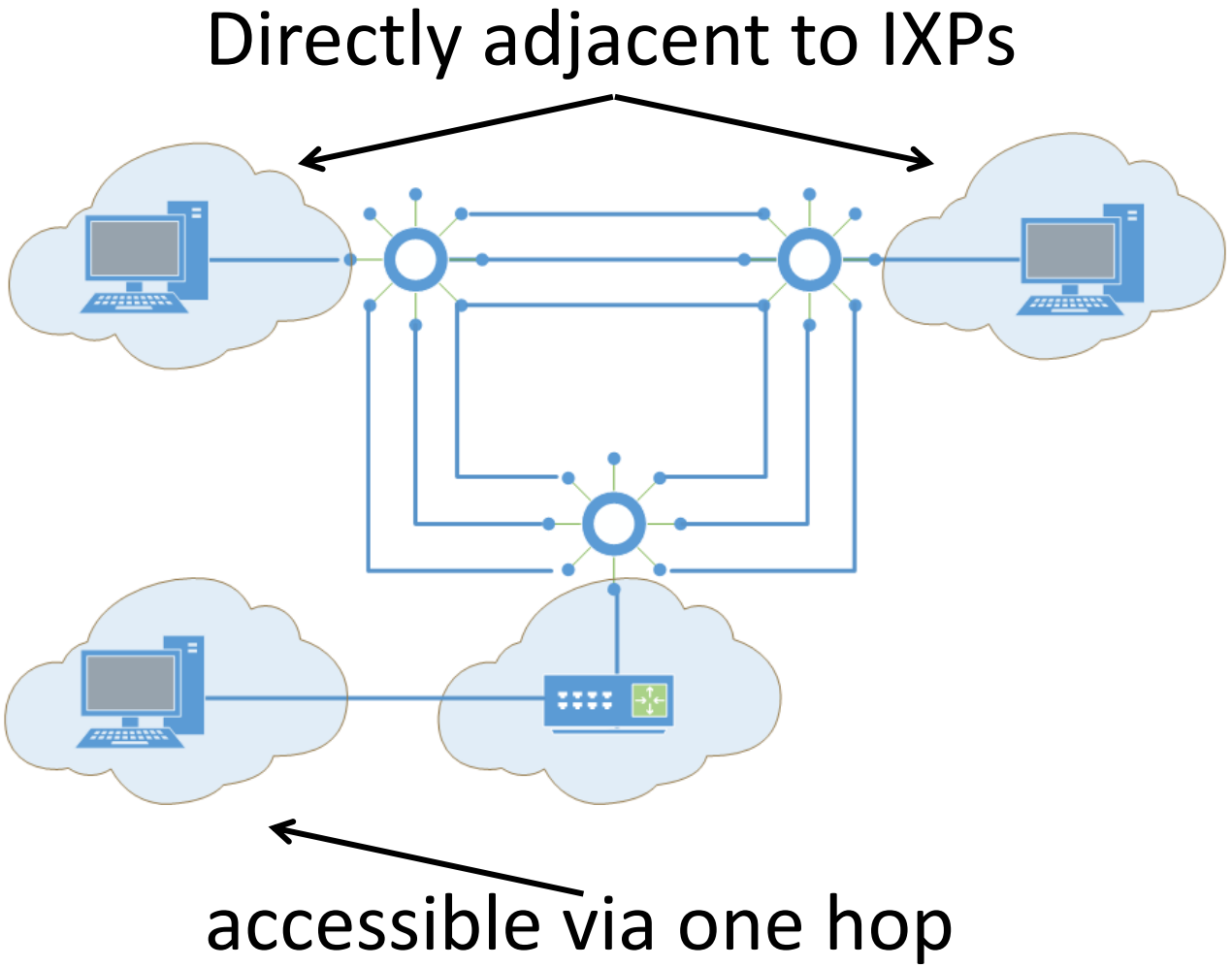
Results at a Glance: IXP Multigraph



Results at a Glance: Customer Reach

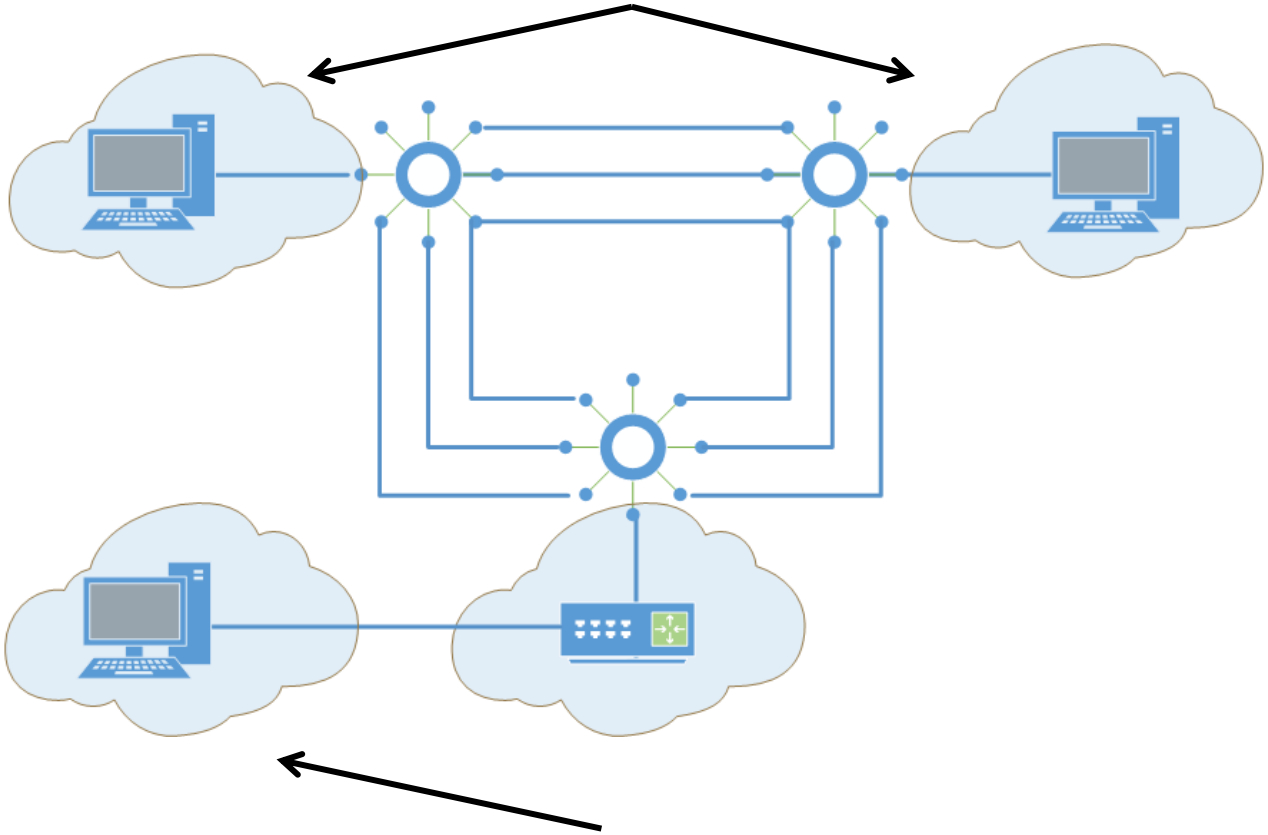


Results at a Glance: Customer Reach



Results at a Glance: Customer Reach

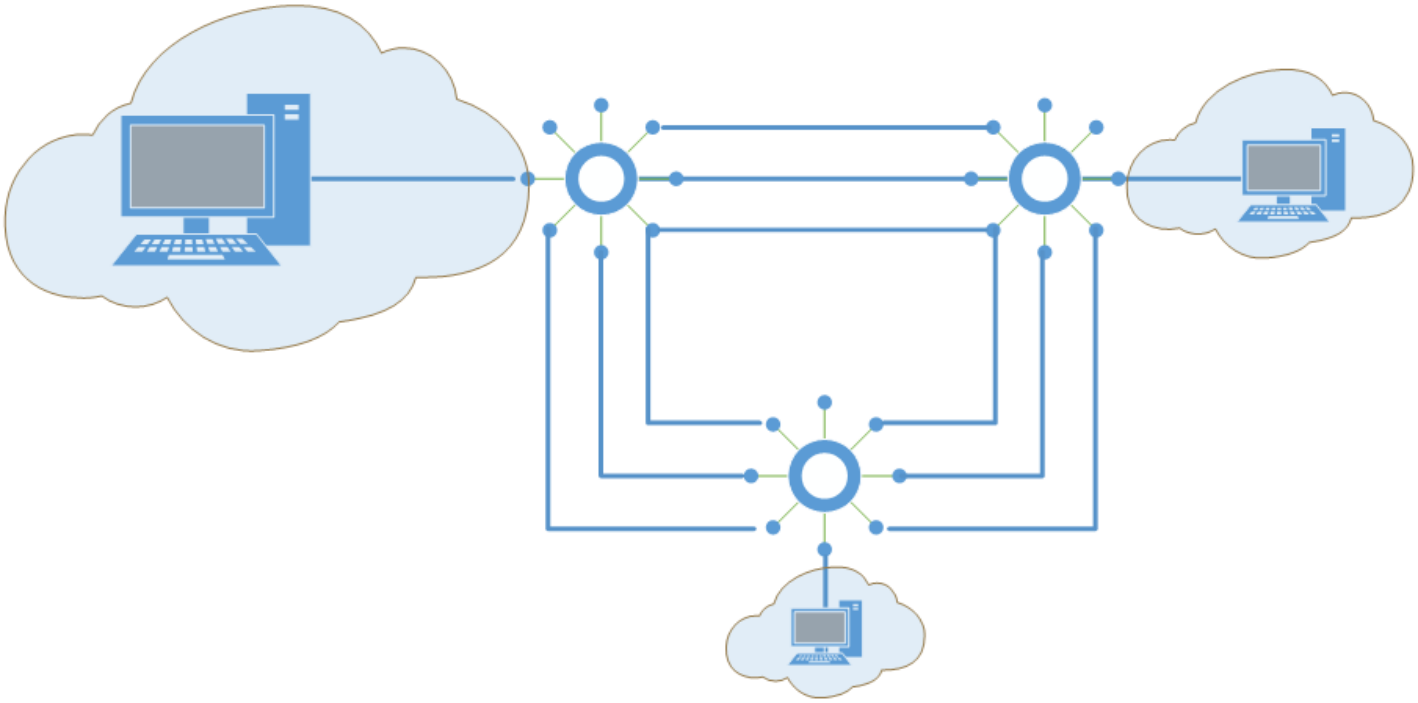
Approx. 61% of IPv4 adresses



Additional 30 % of IPv4 adresses

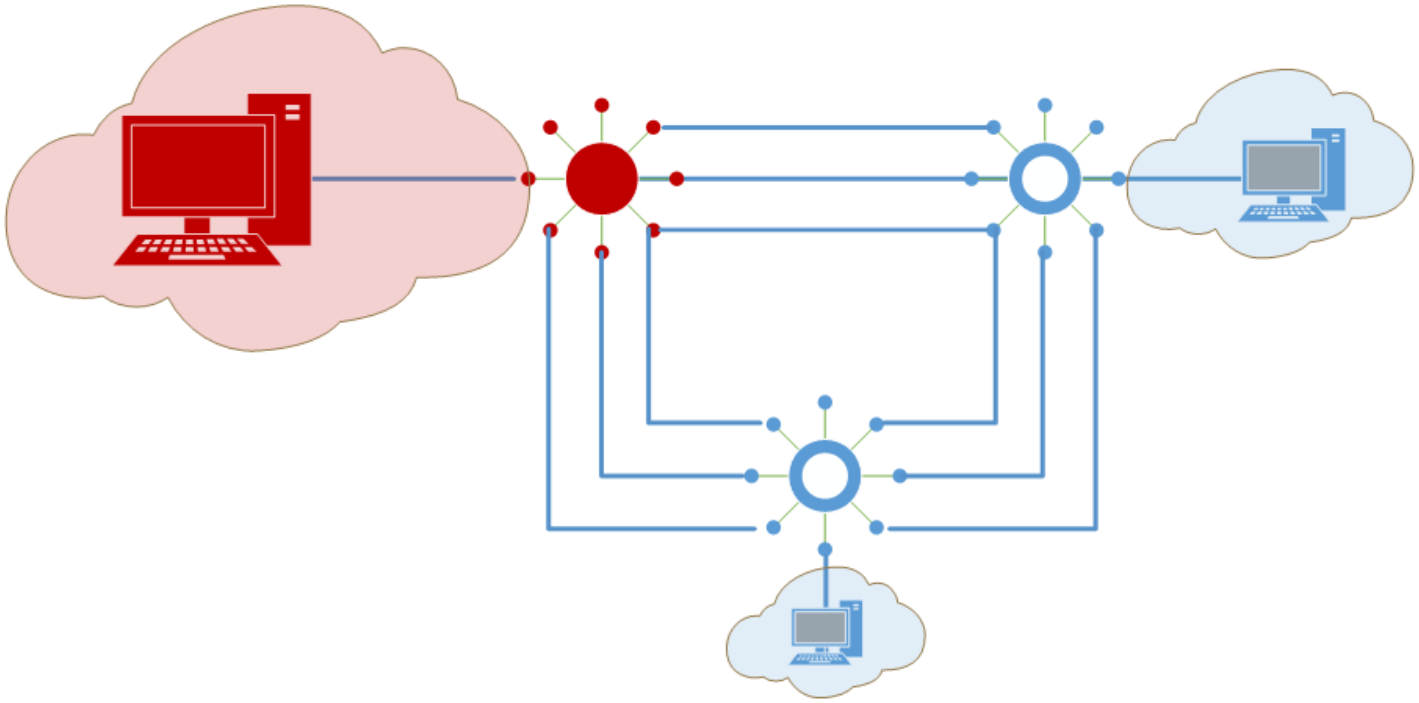
Incremental Deployment of CXPs

- Do we really need all of the 229 IXPs to offer end-to-end paths?
- Greedily select IXPs maximizing customer cone.



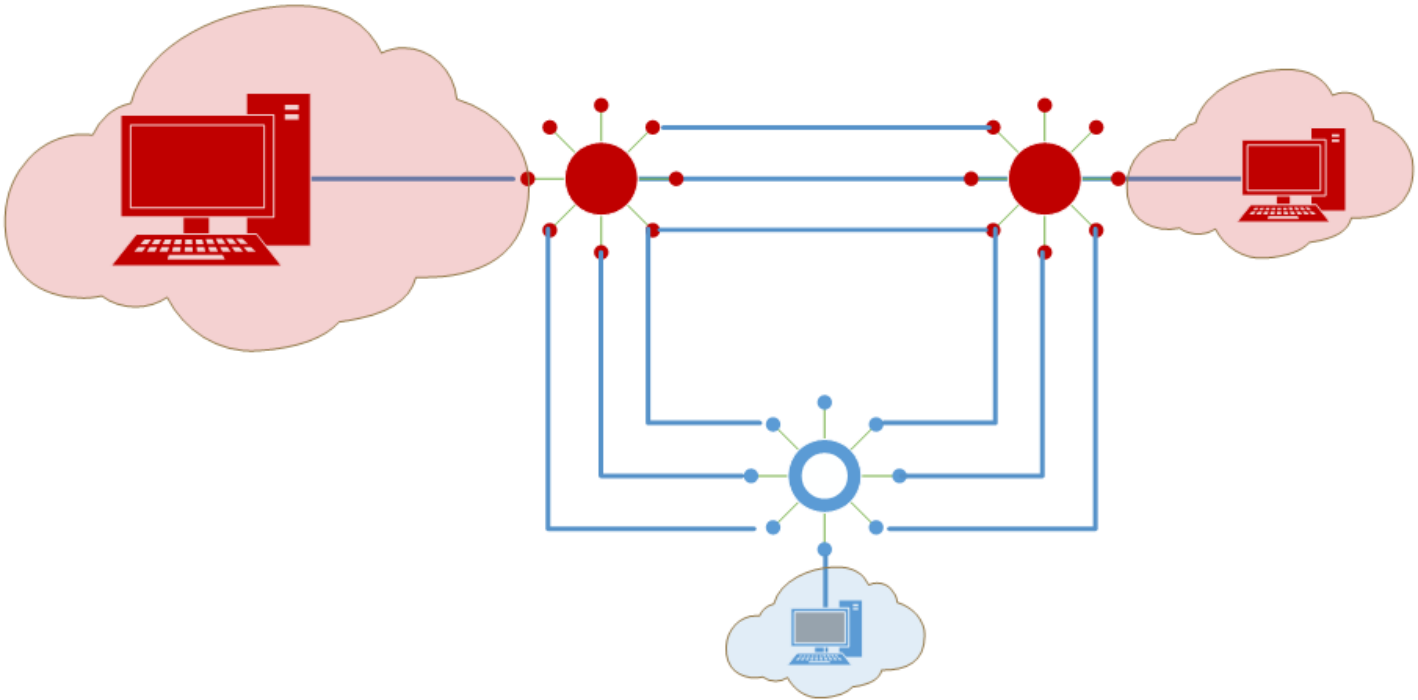
Incremental Deployment of CXPs

- Do we really need all of the 229 IXPs to offer end-to-end paths?
- Greedily select IXPs maximizing customer cone.



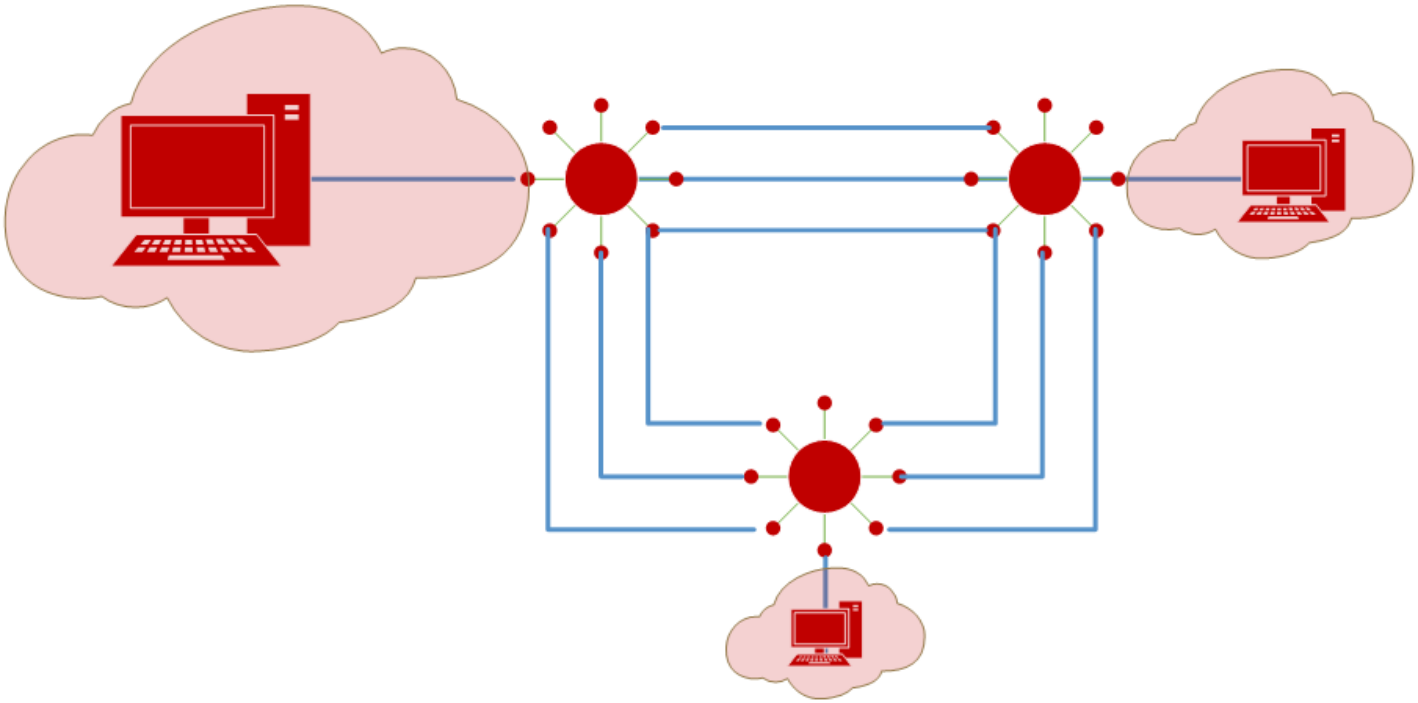
Incremental Deployment of CXPs

- Do we really need all of the 229 IXPs to offer end-to-end paths?
- Greedily select IXPs maximizing customer cone.



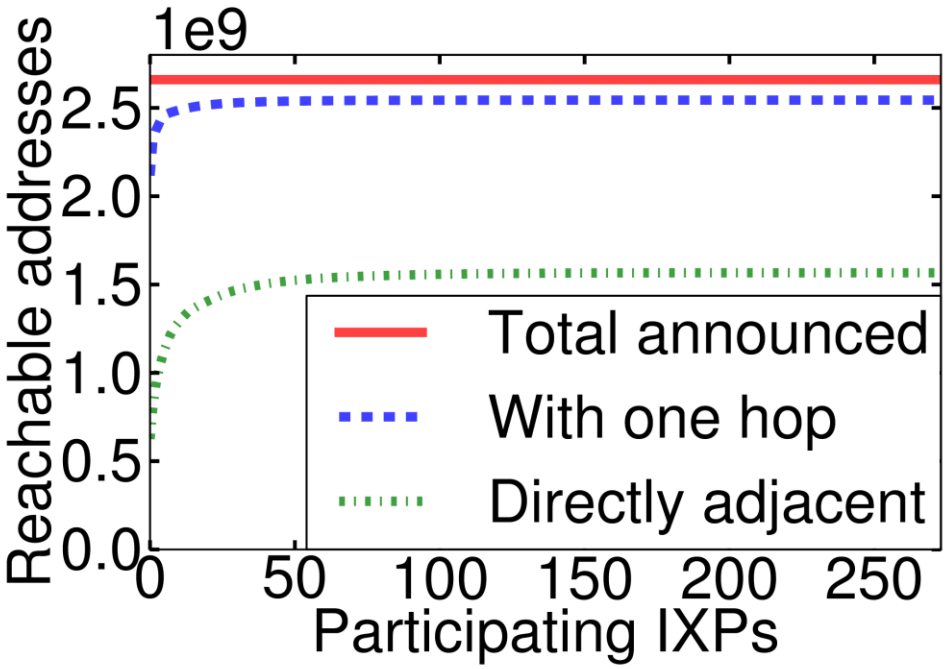
Incremental Deployment of CXPs

- Do we really need all of the 229 IXPs to offer end-to-end paths?
- Greedily select IXPs maximizing customer cone.



Incremental Deployment of CXPs

- Do we really need all of the 229 IXPs to offer end-to-end paths?
- Greedily select IXPs maximizing customer cone.



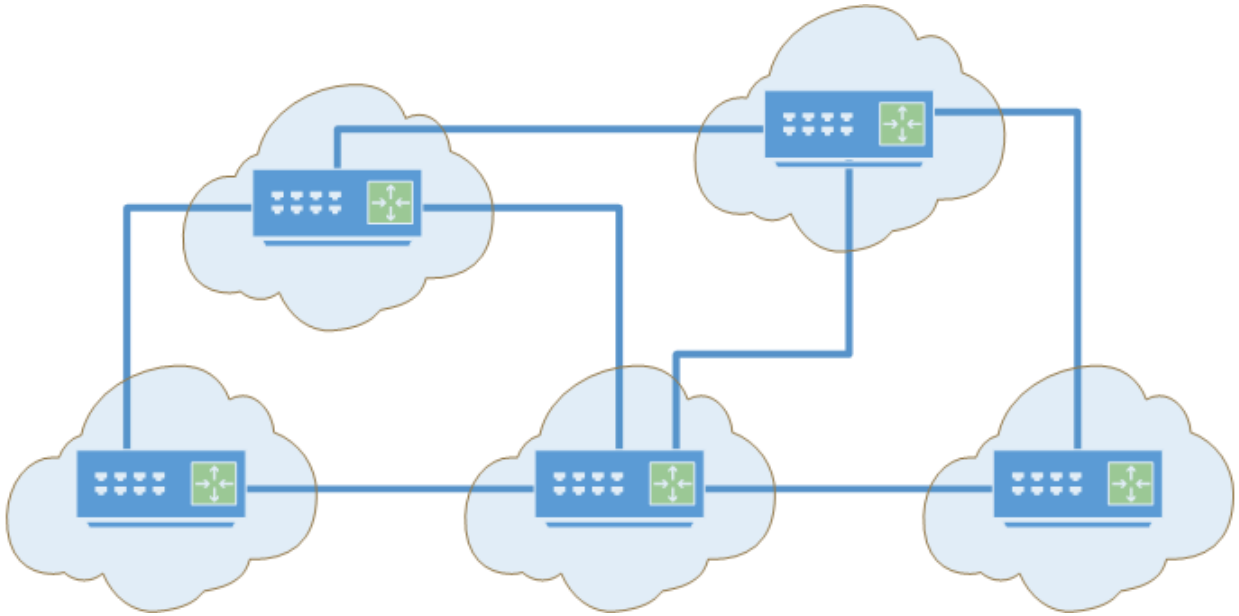
Incremental Deployment of CXPs

- Do we really need all of the 229 IXPs to offer end-to-end paths?
- Greedily select IXPs maximizing customer cone.

number of IXPs	reachable directly	with 1-hop
5	approx. 40%	approx. 91%
20	approx. 55%	approx. 92%

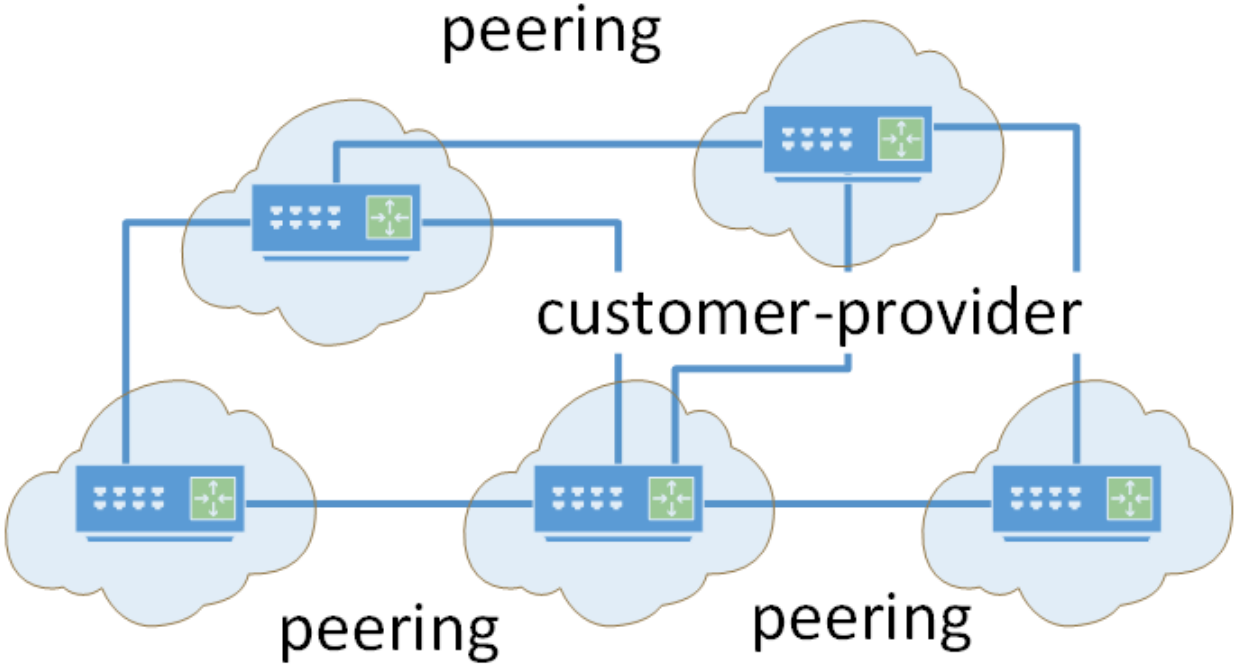
Path Diversity

- What is the gain in path diversity over BGP?



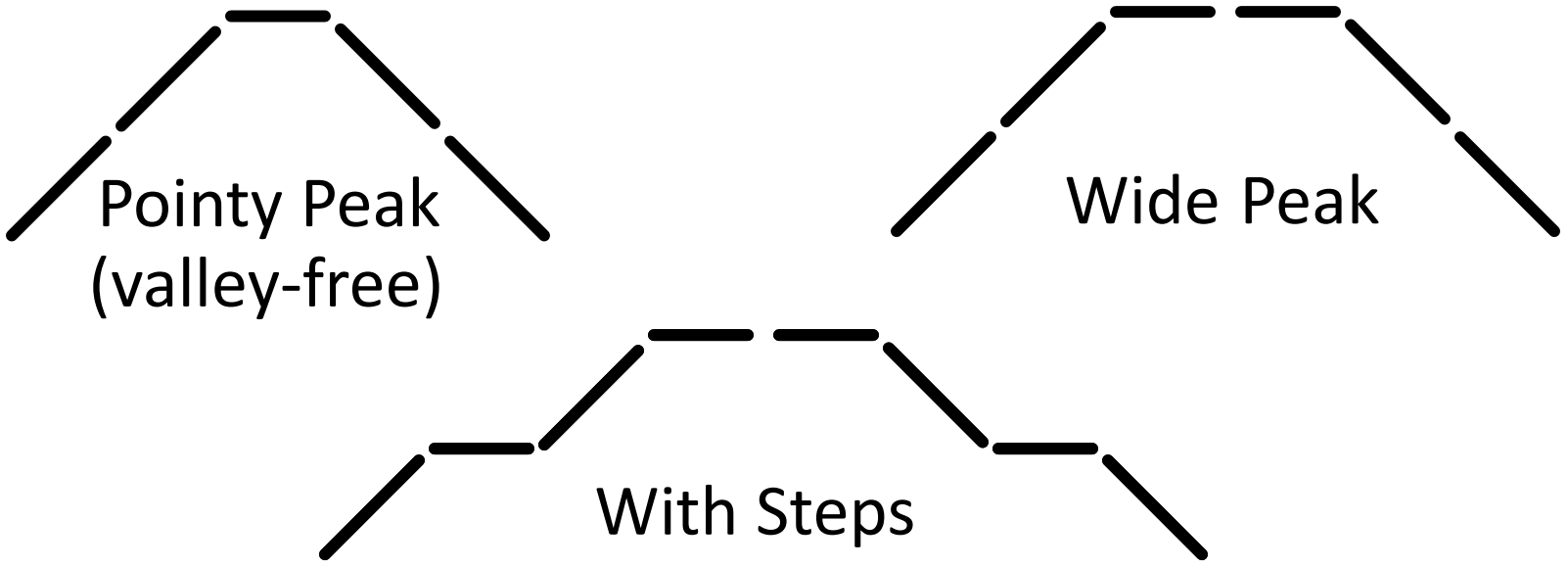
Path Diversity

- What is the gain in path diversity over BGP?
- BGP: valley-free (at most one peering link)

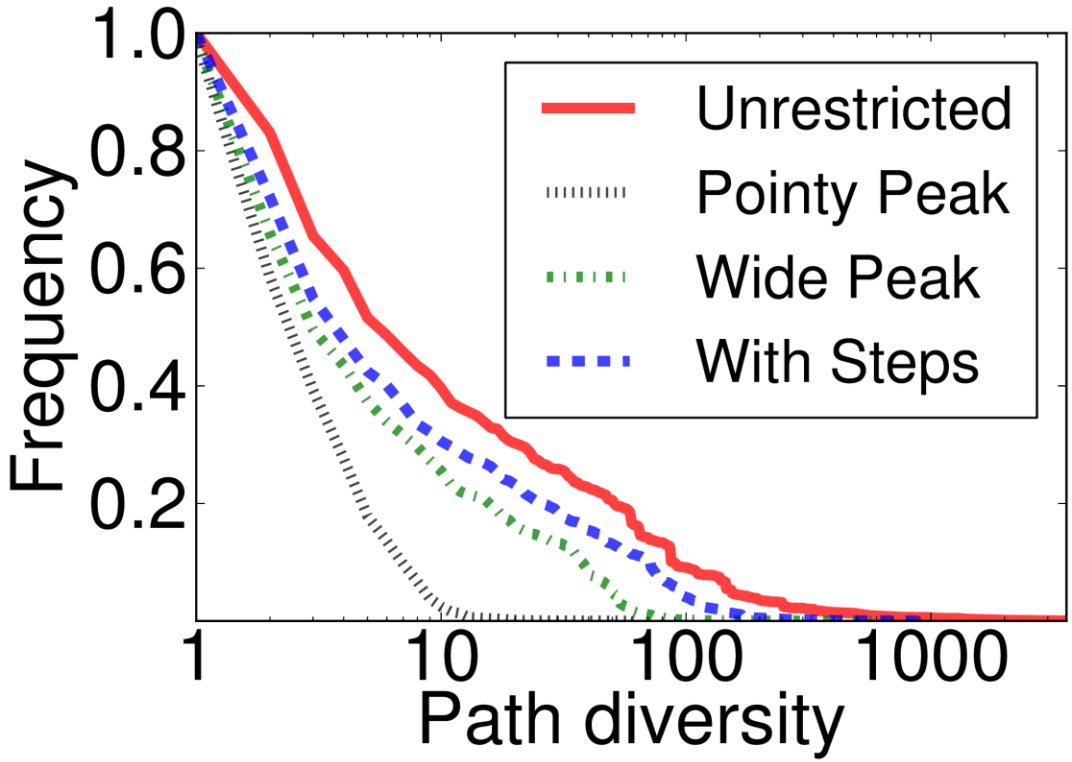


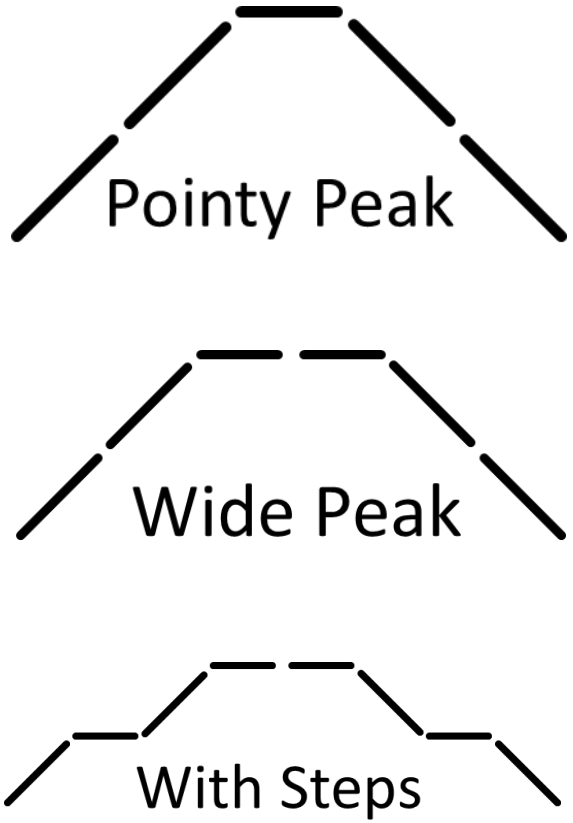
Path Diversity

- What is the gain in path diversity over BGP?
- BGP: valley-free (at most one peering link)

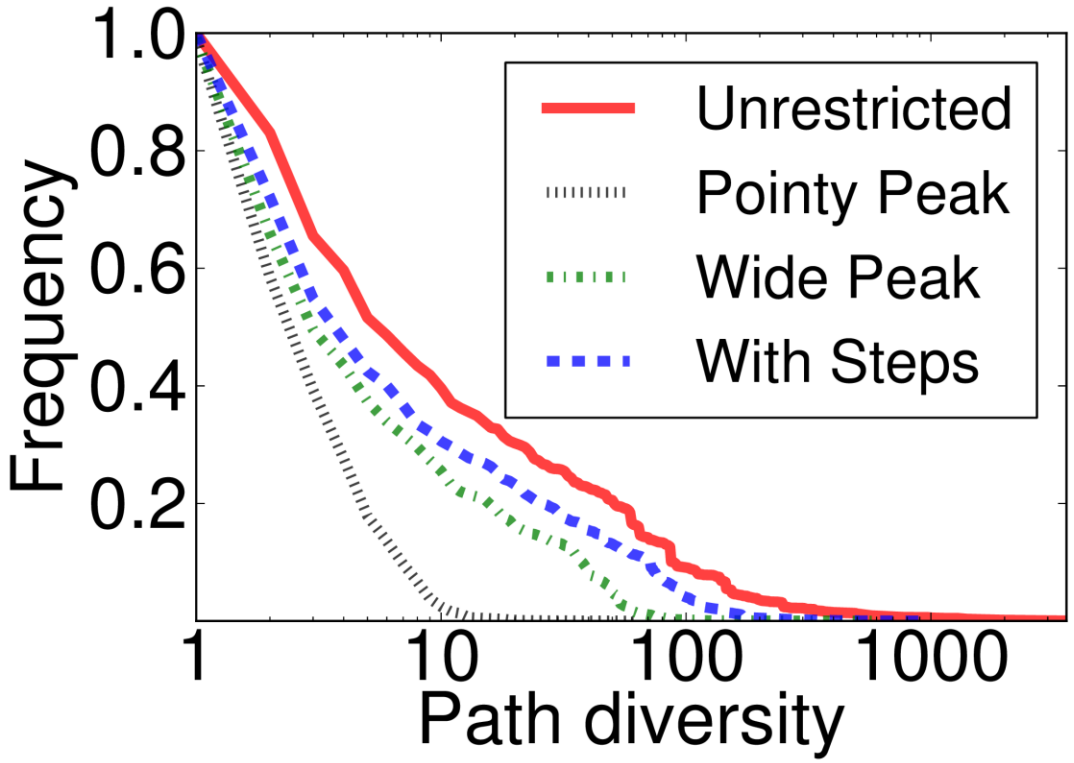


Path Diversity: Results





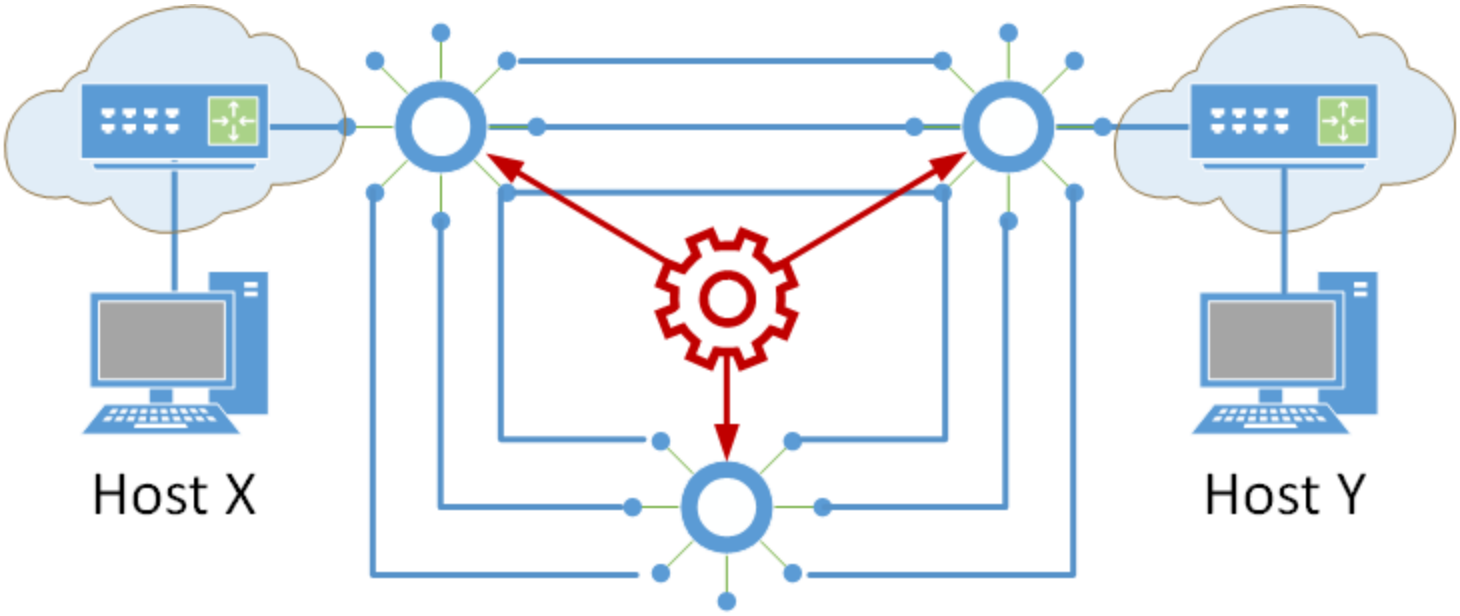
Path Diversity: Results



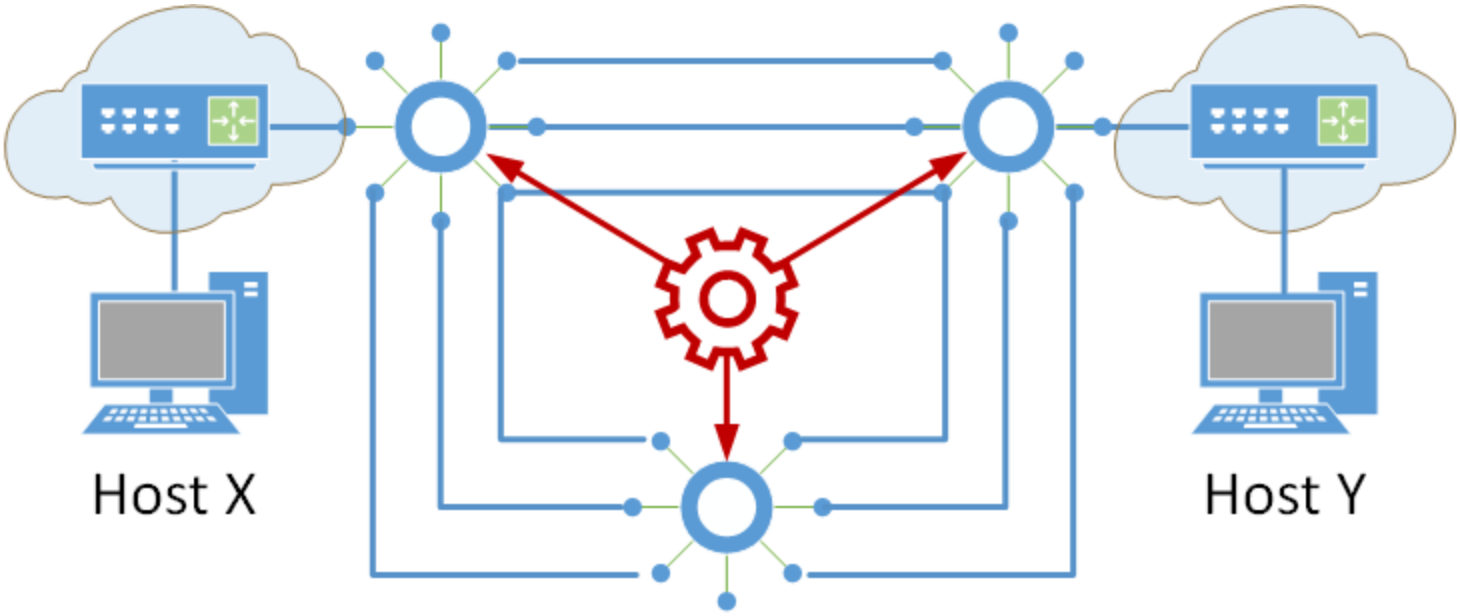
- Significant gains even when only stitching peering links
- Up to 29x times the path diversity

HOW TO EFFICIENTLY COMPUTE END-TO-END PATHS AT CXPS

Model

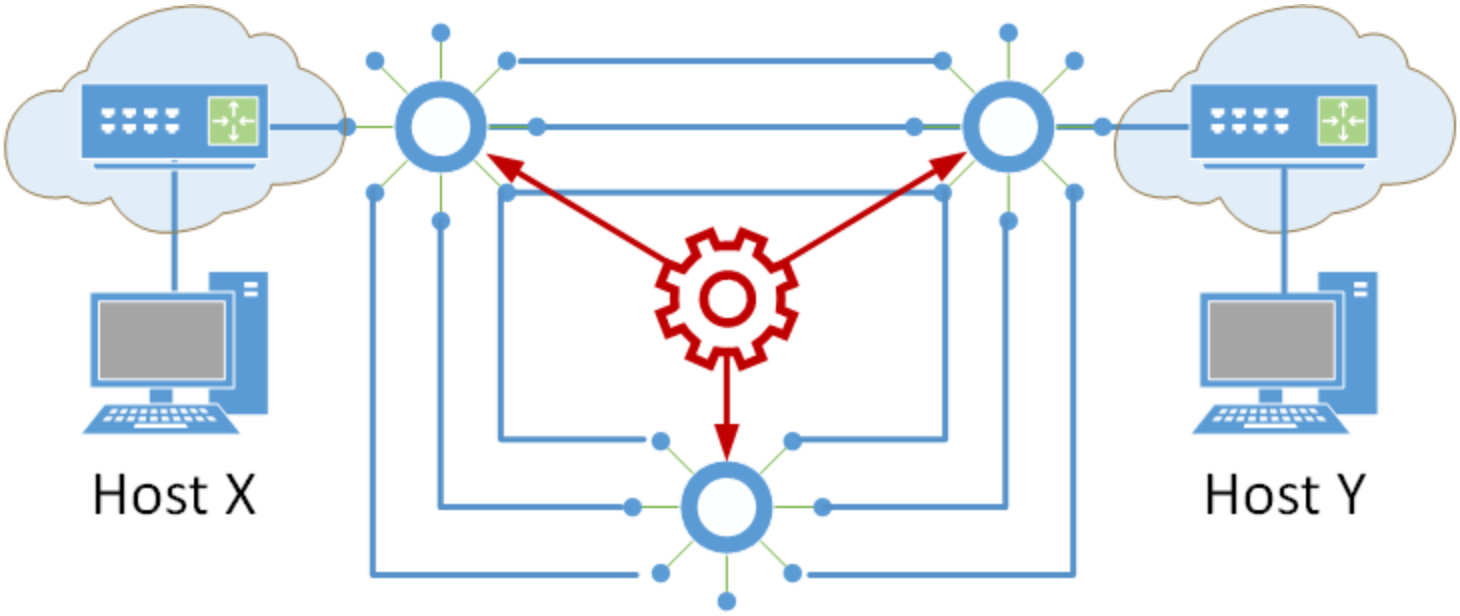


Model



Hosts:	X,Y
Bandwidth:	50 Mbps
Max. Latency:	100 ms

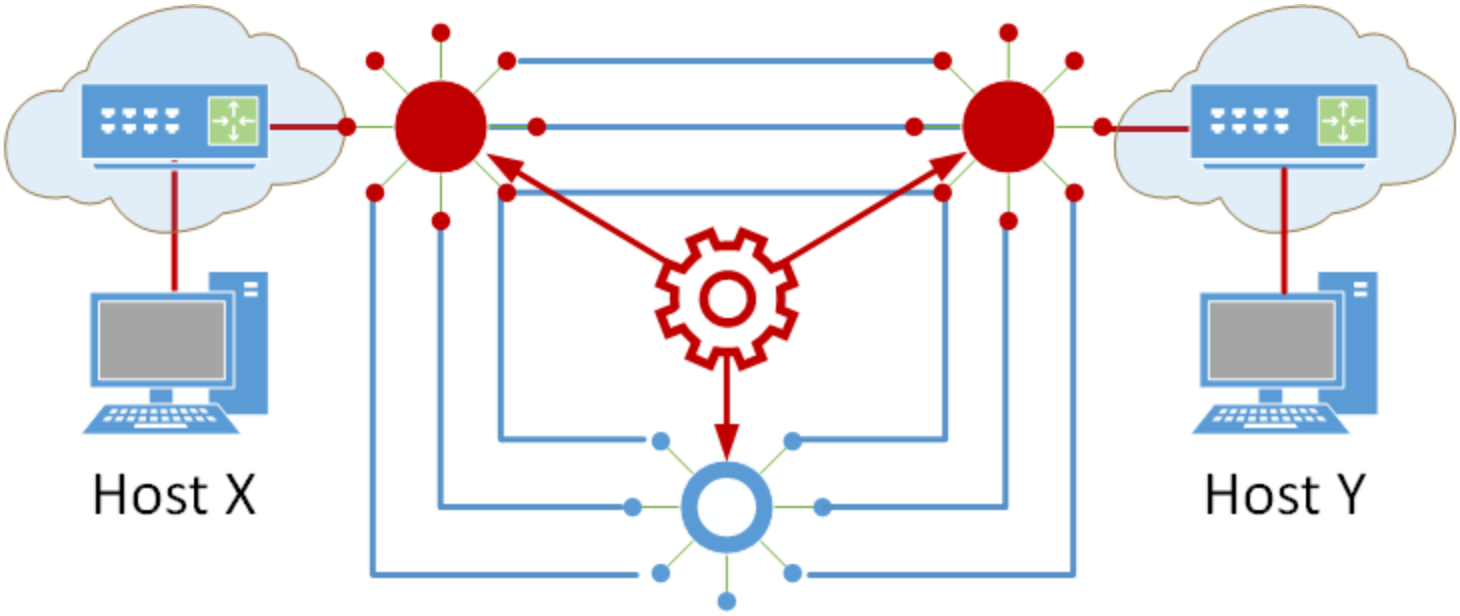
Model



Hosts: X,Y
Bandwidth: 50 Mbps
Max. Latency: 100 ms

Objective: embed requests (as many as possible)

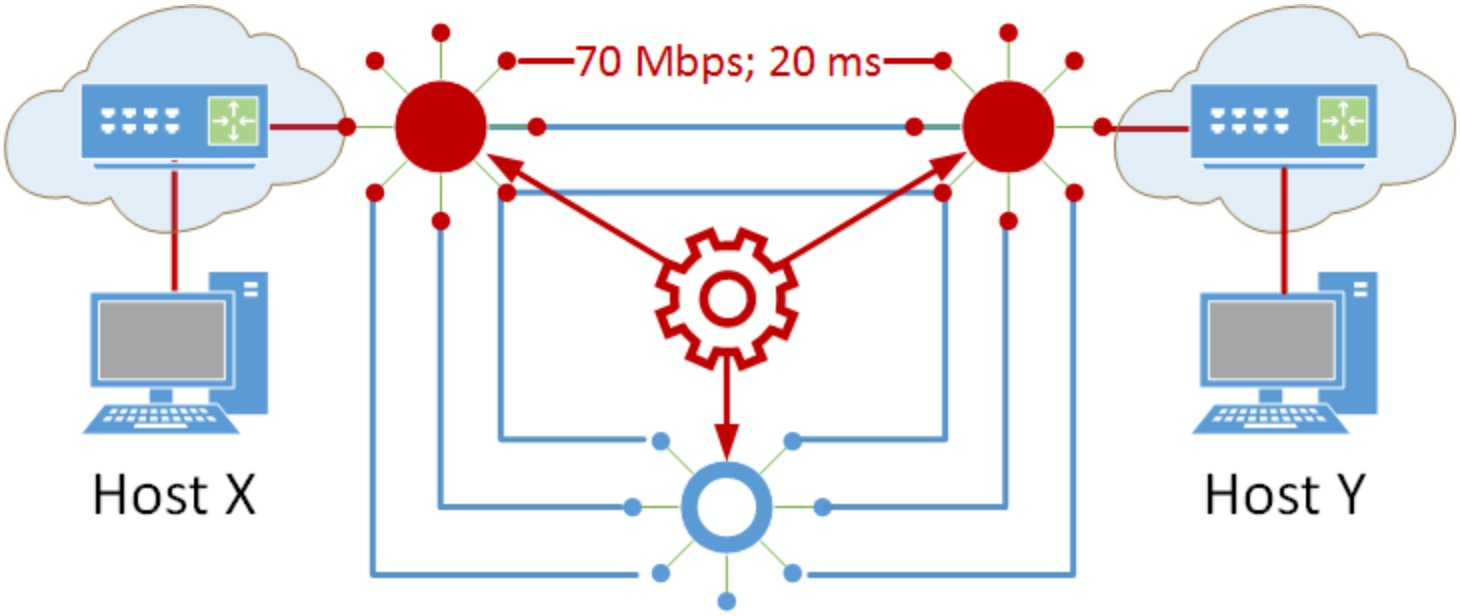
Model



Hosts: X,Y
Bandwidth: 50 Mbps
Max. Latency: 70 ms

ASes of X and Y provide connectivity to the IXPs

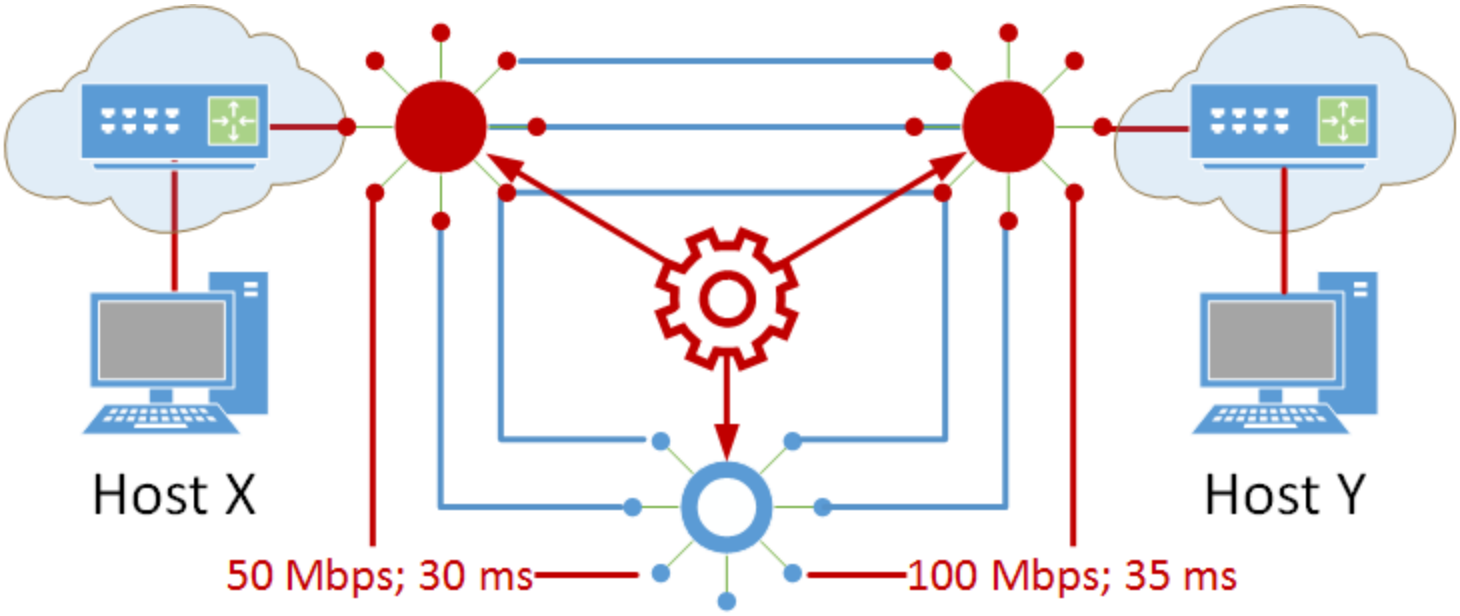
Model



Hosts:	X,Y
Bandwidth:	50 Mbps
Max. Latency:	70 ms

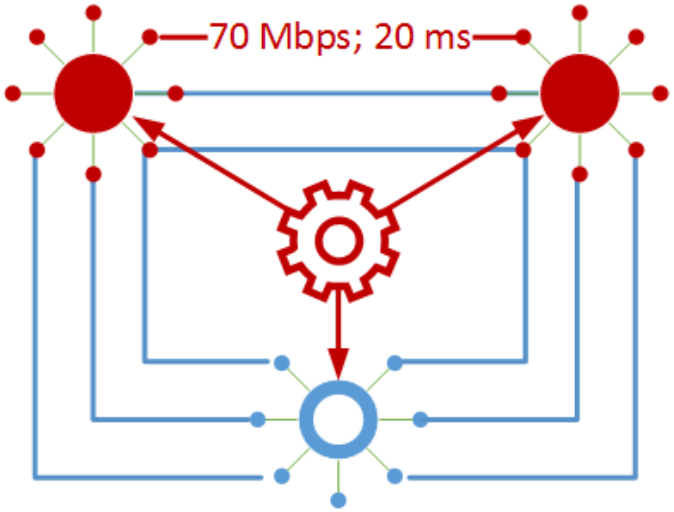
Task: find appropriate path for connecting the IXPs

Model

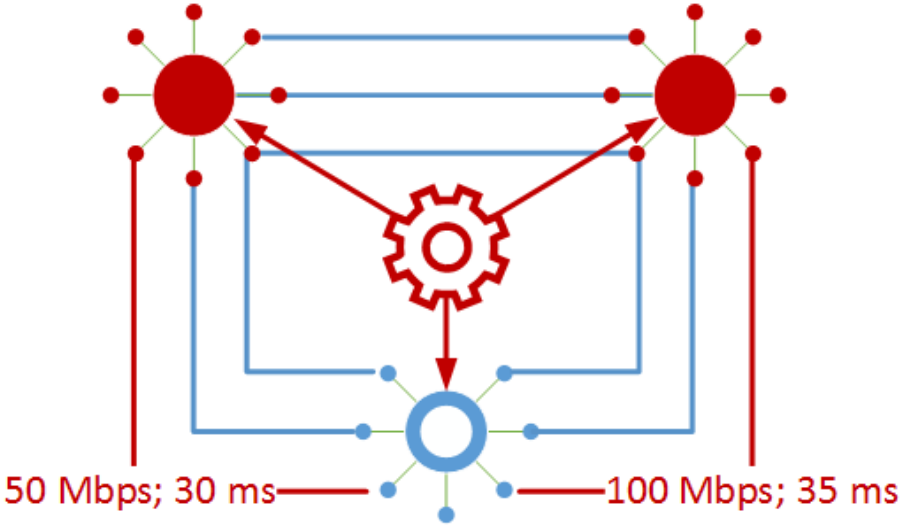


Hosts:	X,Y
Bandwidth:	50 Mbps
Max. Latency:	70 ms

Trading Off Objectives

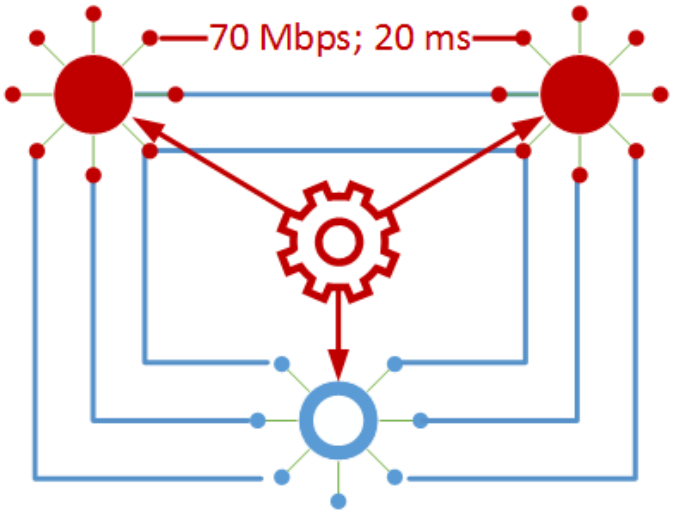


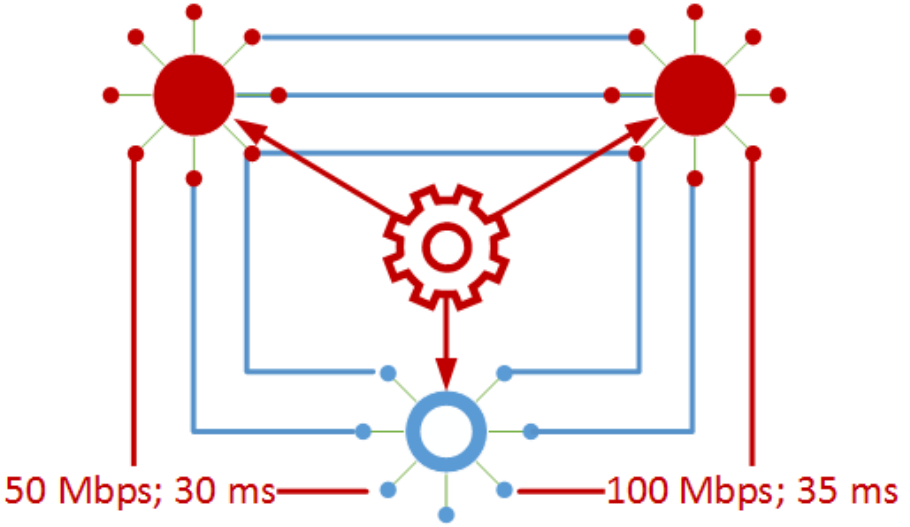
- single link
- resource fragmentation
- uses low-latency link



- two links
- no resource fragmentation

Trading Off Objectives





CXP should consider

- resource utilization (hop count)
- resource fragmentation
- utilization of scarce resources

Finding Good Paths is Challenging

Theory

- Finding optimal paths is NP-hard when considering latency etc.!
- Feasible paths can be found in polynomial time.

Practice

- Even when only considering 14 IXPs, the IXP multigraph contains around 4k edges.

Finding Good Paths is Challenging

Theory

- Finding optimal paths is NP-hard when considering latency etc.!
- Feasible paths can be found in polynomial time.

Practice

- Even when only considering 14 IXPs, the IXP multigraph contains around 4k edges.

We develop an algorithmic framework.

- Efficiently computing paths.
- Harnessing centralized control.

Finding Good Paths is Challenging

Theory

- Finding optimal paths is NP-hard when considering latency etc.!
- Feasible paths can be found in polynomial time.

Practice

- Even when only considering 14 IXPs, the IXP multigraph contains around 4k edges.

Sample-Selection Framework

- Sample set of feasible paths.
- Select “best” one found.
- Reconfigure later if necessary.

Path Sampling Strategies

- Perturbed Dijkstra (PD)
  - project inter-IXP links on the lowest latency one and apply Dijkstra
  - iterate without the links found
- Guided Dijkstra (GD)
  - Dijkstra choosing a single inter-IXP link at random during neighborhood exploration
- Guided Walk (GW)
  - Choose next IXP node and the respective edge uniformly at random

Path Selection Strategy

- Strictly prefer paths with smaller hop count
- Break ties by …
  - trying to avoid using scarce low latency links
  - trying to avoid depleting bandwidth between adjacent IXPs

Reconfiguration Support

- We propose Integer Program HeurPaths for background reconfigurations.
- Given the previously sampled paths, it selects any of them for enabling the embedding of additional requests.

Integer Program 2: Heuristic Path Formulation (HeurPaths)

$$\max \sum_{R \in \mathcal{R}} x_R \qquad \text{(OBJ)}$$

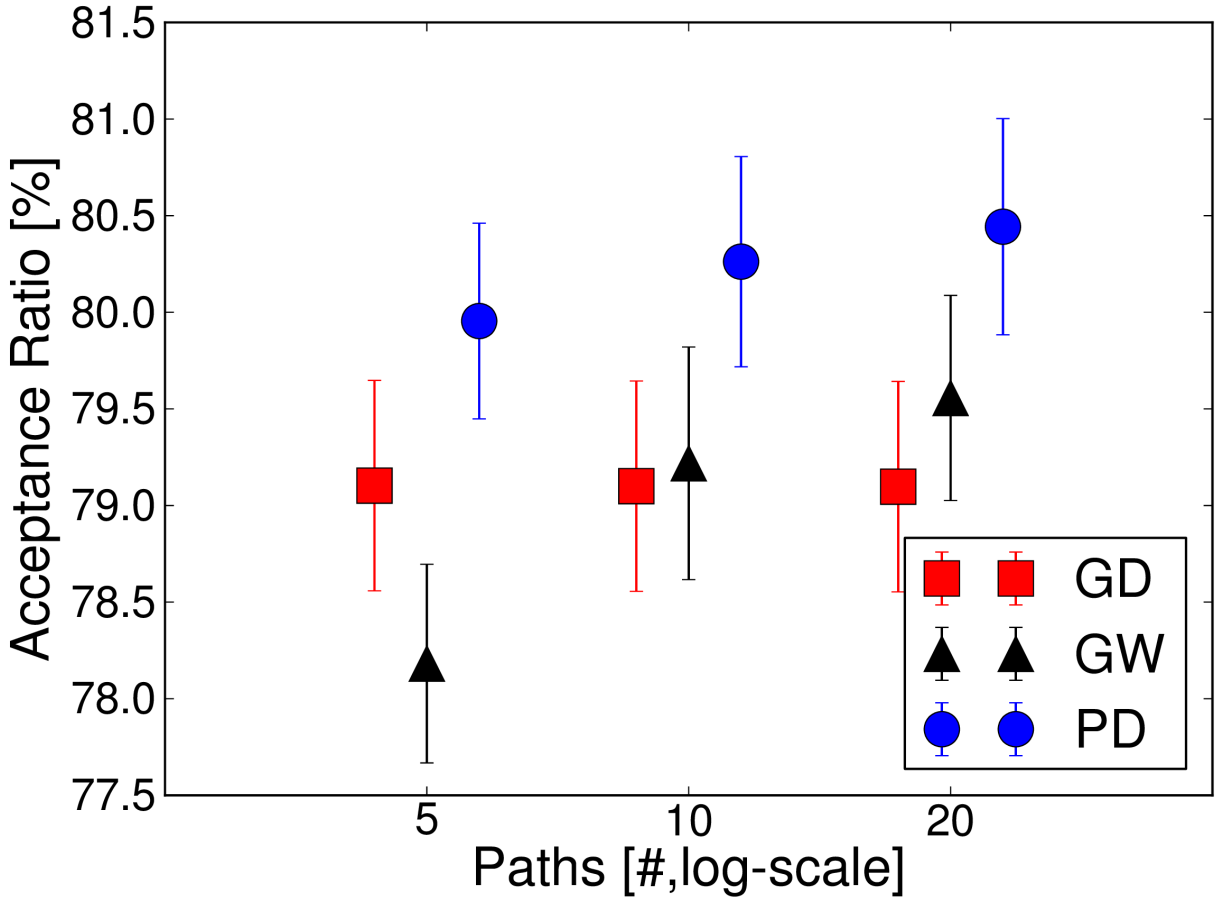
$$x_R = \sum_{P_R \in \mathcal{P}_R} y_{P_R} \qquad \forall R \in \mathcal{R} \qquad \text{(HP-1)}$$

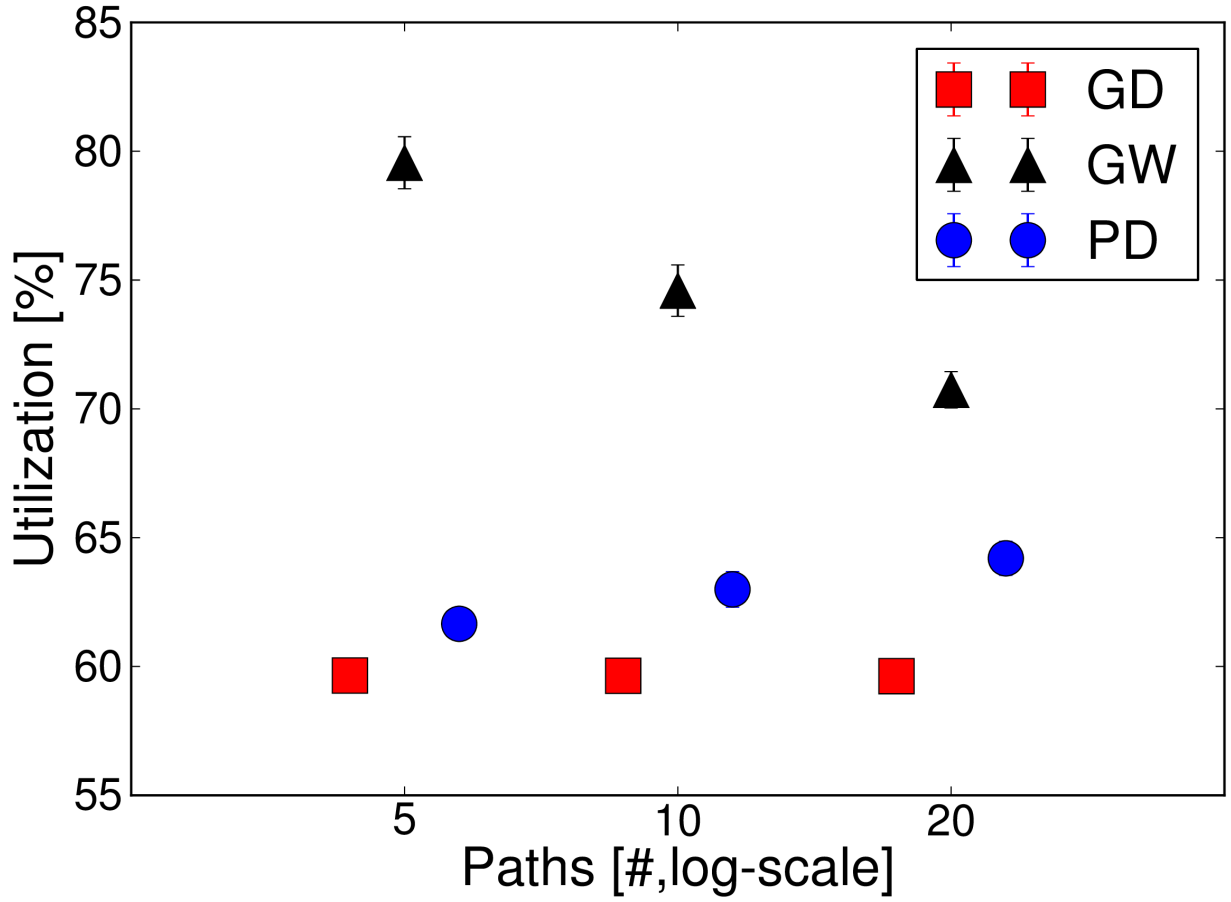
$$\mathsf{bw}_e \geq \sum_{R \in \mathcal{R}, e \in P_R} \mathsf{bw}_R \cdot y_R \qquad \forall\, e \in E_G \qquad \text{(HP-2)}$$

$$x_R \in \{0,1\} \qquad \forall\, R \in \mathcal{R} \qquad \text{(HP-3)}$$

$$y_{P_R} \in \{0,1\} \qquad \forall\, R \in \mathcal{R}, P_R \in \mathcal{P}_R \qquad \text{(HP-4)}$$

EVALUATION


81.5
81.0
80.5
80.0
79.5
79.0
78.5
78.0
77.5
Acceptance Ratio [%]
5
10
20
Paths [#,log-scale]
GD
GW
PD


85
80
75
70
65
60
55
Utilization [%]
5
10
20
Paths [#,log-scale]
GD
GW
PD

Challenges

- CXP needs to embed requests such that latency and bandwidth requirements are satisfied
- CXP would like to …
  - embed as many requests as possible (profit!)
  - minimize resource utilization
  - avoid resource fragmentation
  - avoid unnecessary usage of low-latency links
- Shortest Paths Problem with multiple objectives: NP-hard

Algorithmic Framework: Sample&Select

- Generate a set of feasible paths (w.r.t. bandwidth and latency) and select one of them using a high-level objective function
  - Sample paths: Variants of Dijkstra / Randomized Walks (5, 10, 20, 100 paths..)
  - Select:
    - Minimize hop count (resource)
    - Try to avoid scarce low latency links / scarce bandwidth links

Path Sampling: Considerations

- Multigraph
  - Given a request we can check whether a solution can exist:
    - Remove all links not supporting the bandwidth
    - Project the graph onto a simple one using lowest-latency edges
    - Peform any shortest-paths algorithm
    - Gives us two types of information:
      - Is it feasible?
      - Shortest paths distances from any node towards the receiver

Path Sampling: Algorithms

- Perturbed Dijkstra
  - Project multigraph onto simple graph and apply Dijkstra
  - Compute shortest path → remove used edges → recompute paths (simple!) will always use low-latency links
- Guided Dijkstra
  - Given any state in the path computation (that means we have a distance for a node from the start), consider the next hop
- Guided Walk
  - Similar to Guided Dijkstra but explore nodes randomly, not using a distance Queue (such that we can always extend the path)

Algorithmic Opportunity: Centralized Batch Embeddings

- Given an existing set of requests we may reconfigure embeddings!
- We provide a simple Integer Program such that provided a set of paths for each request any of the path can be chosen, trying to maximize the number of embedded requests
- Even for 10,000 requests with 20 paths each, computation times for optimal solutions lies within 5 seconds to 1 minute
- Optimal IP will take up to hours!

Simulation Parameter Space

- Number of requests
- Arrival Process
- Topology used
- Latency distribution
- End-host distribution
- Paths per request
- Capacities of requests and substrate network

Simulation Parameter Space

- Number of requests
- Arrival Process / Leaving Process
- Topology used
- Latency distribution (requests and substrate)
- End-host distribution
- Paths per request
- Capacities of requests and substrate network

RESULTS

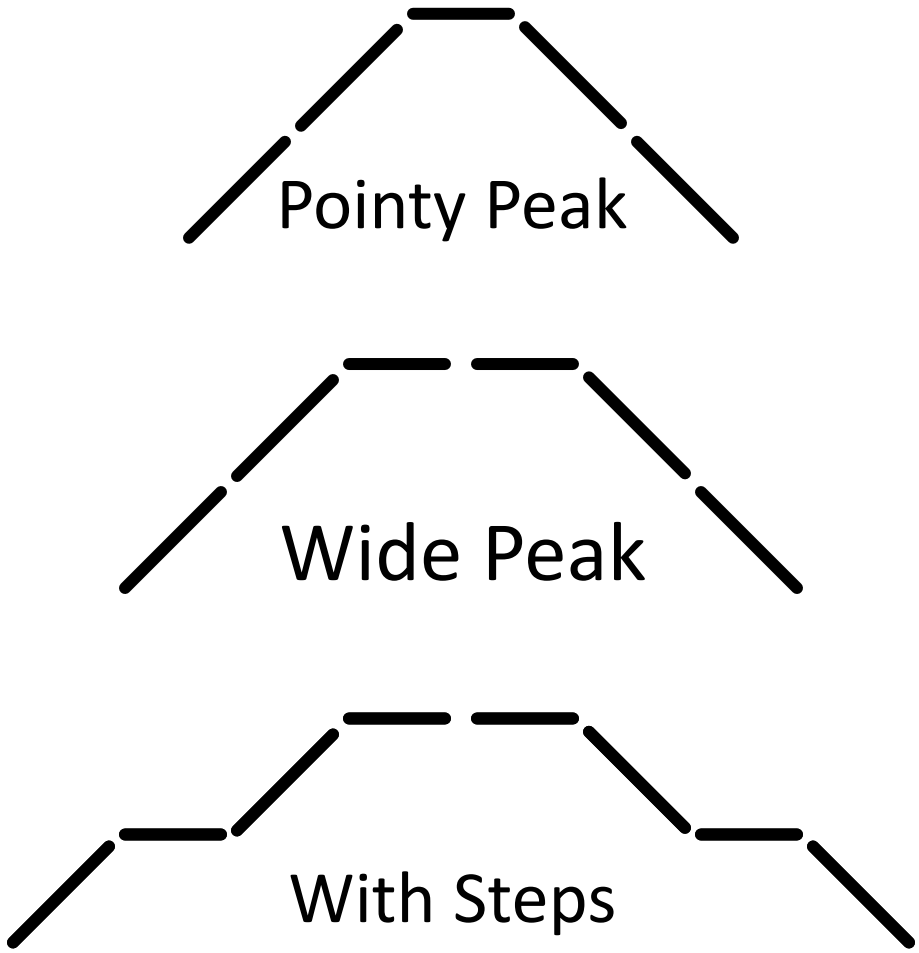
Acceptance Ratio

Code Availability

- Github link

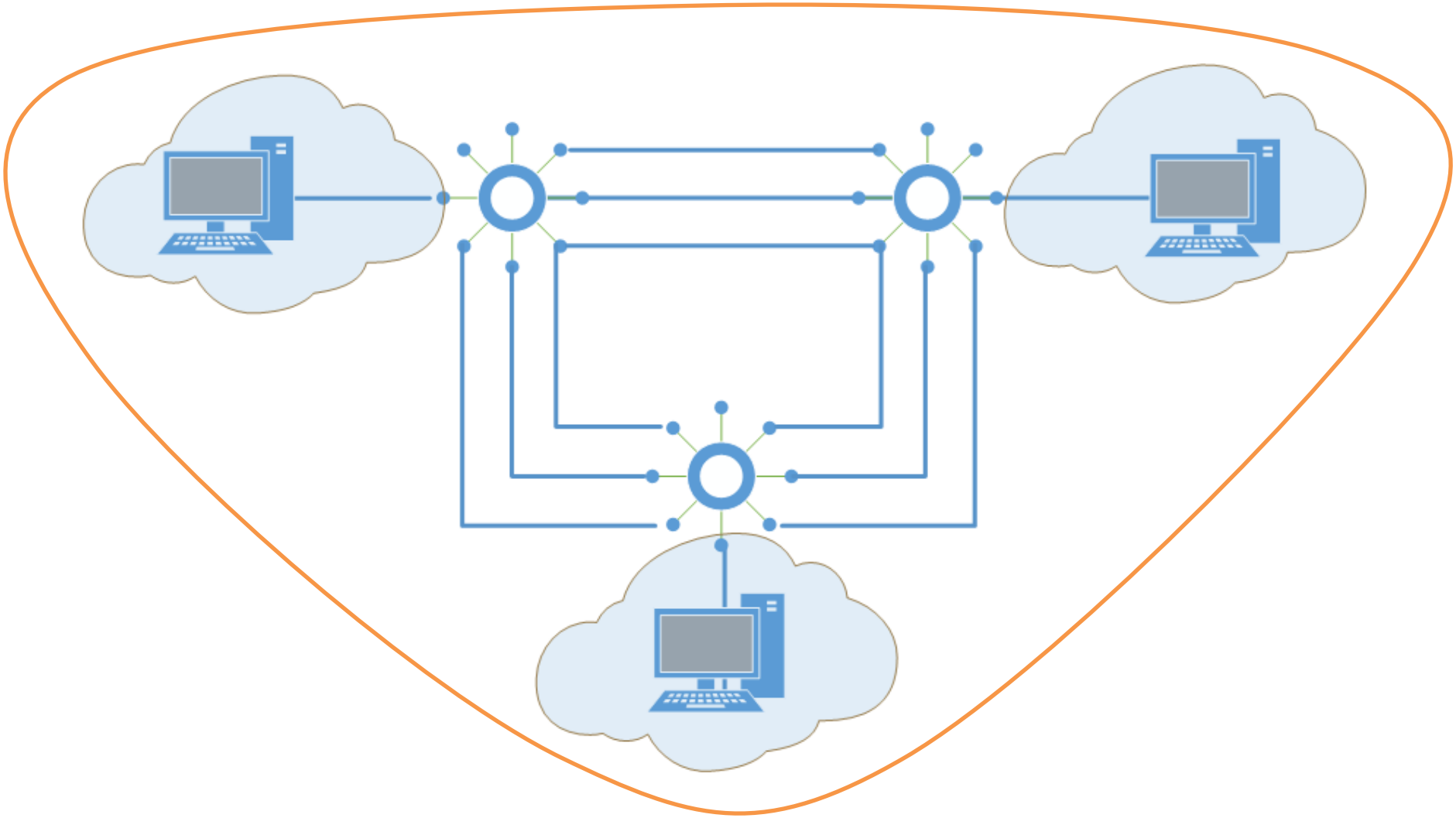
CONCLUSION

Path Diversity: Considered Models



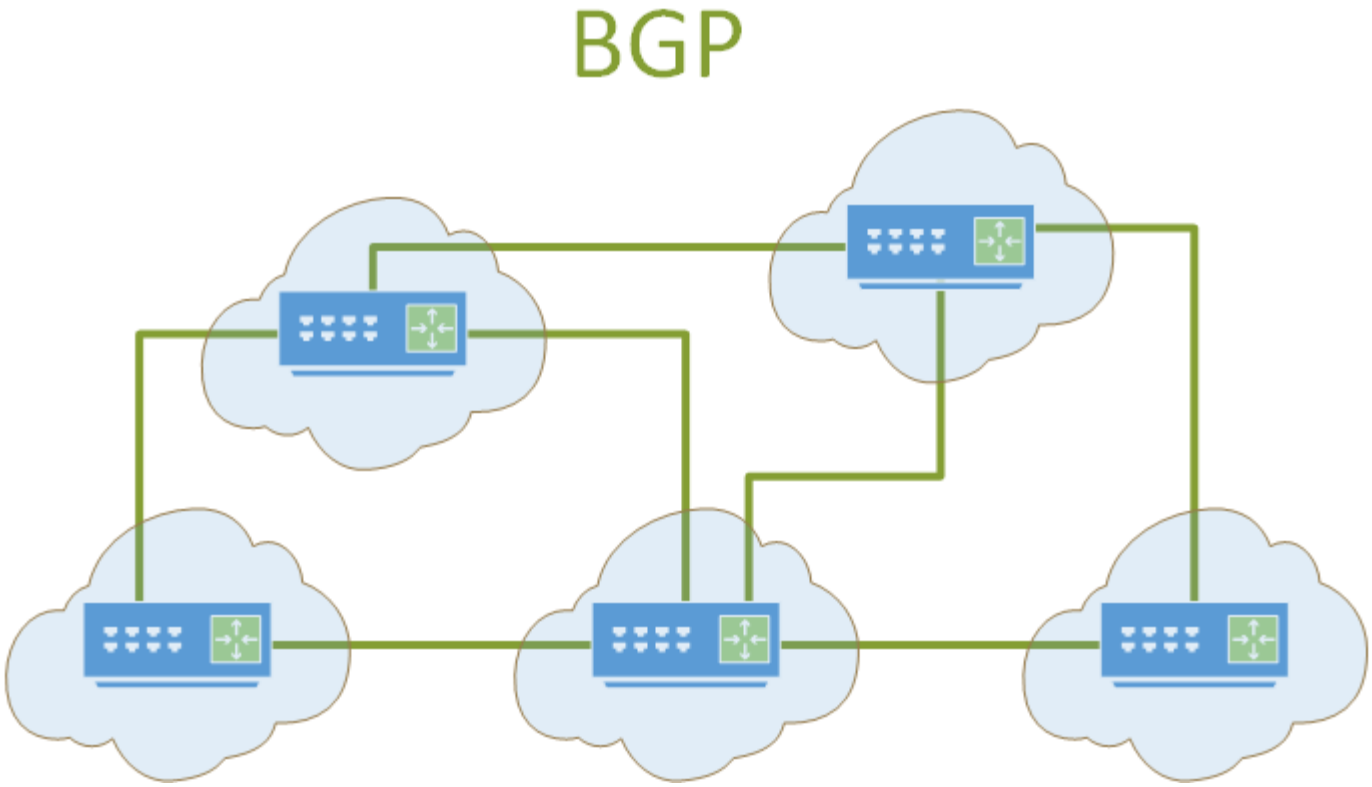
THE FOLLOWING ARE BACKUP SLIDES AT THE MOMENT

Results at a Glance



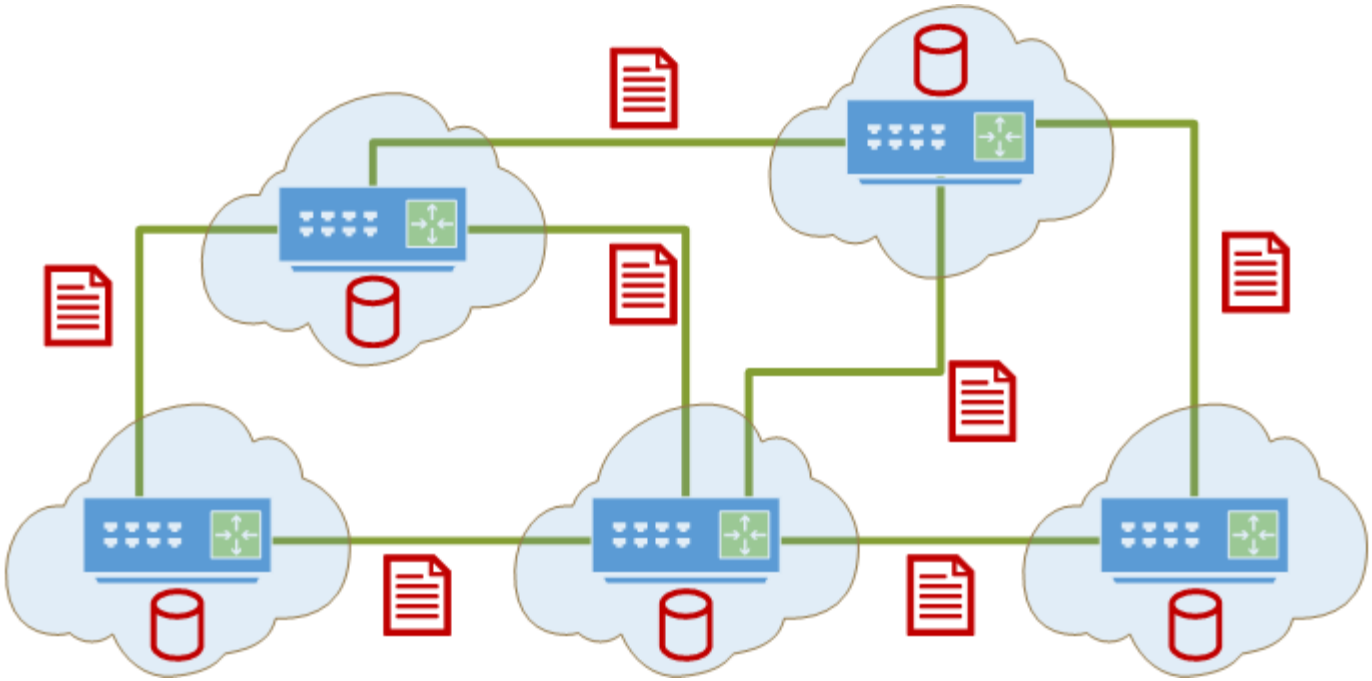
Overcoming BGP's limitations

- Much research in the last 20 years



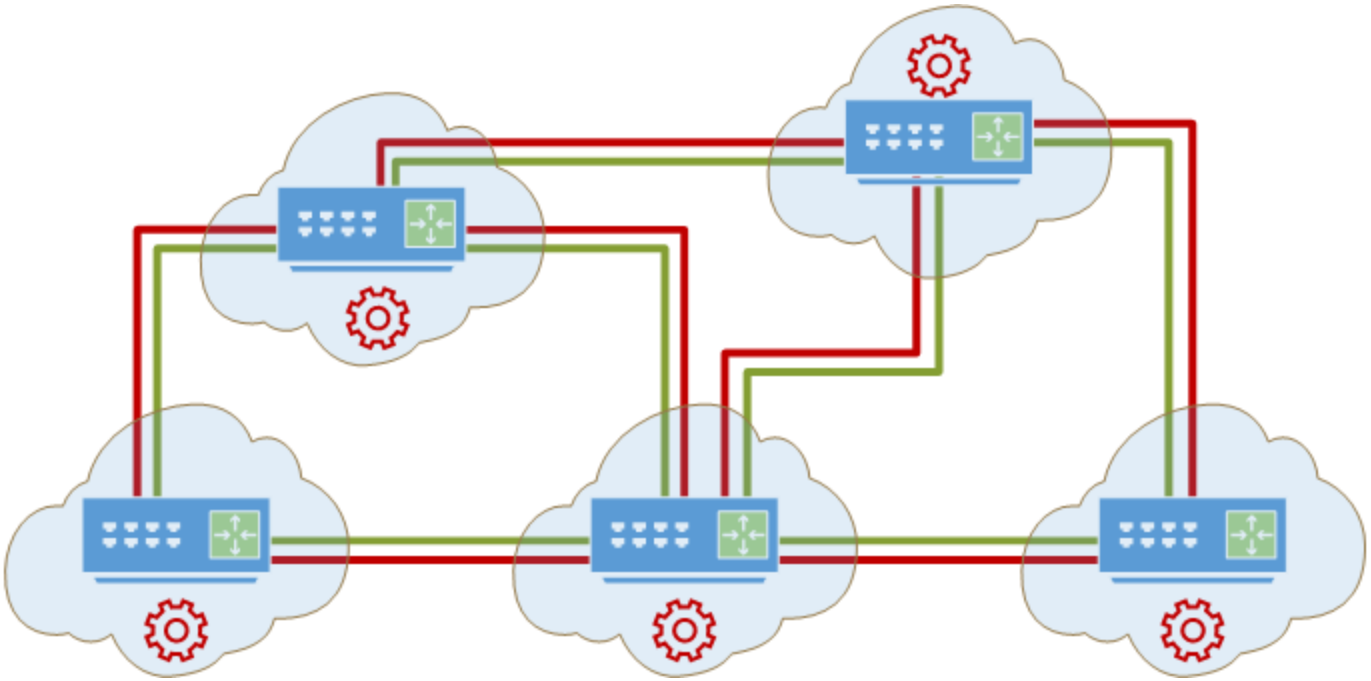
Overcoming BGP's limitations

- Much research in the last 20 years
  - Extend BGP: e.g. Miro, Nira



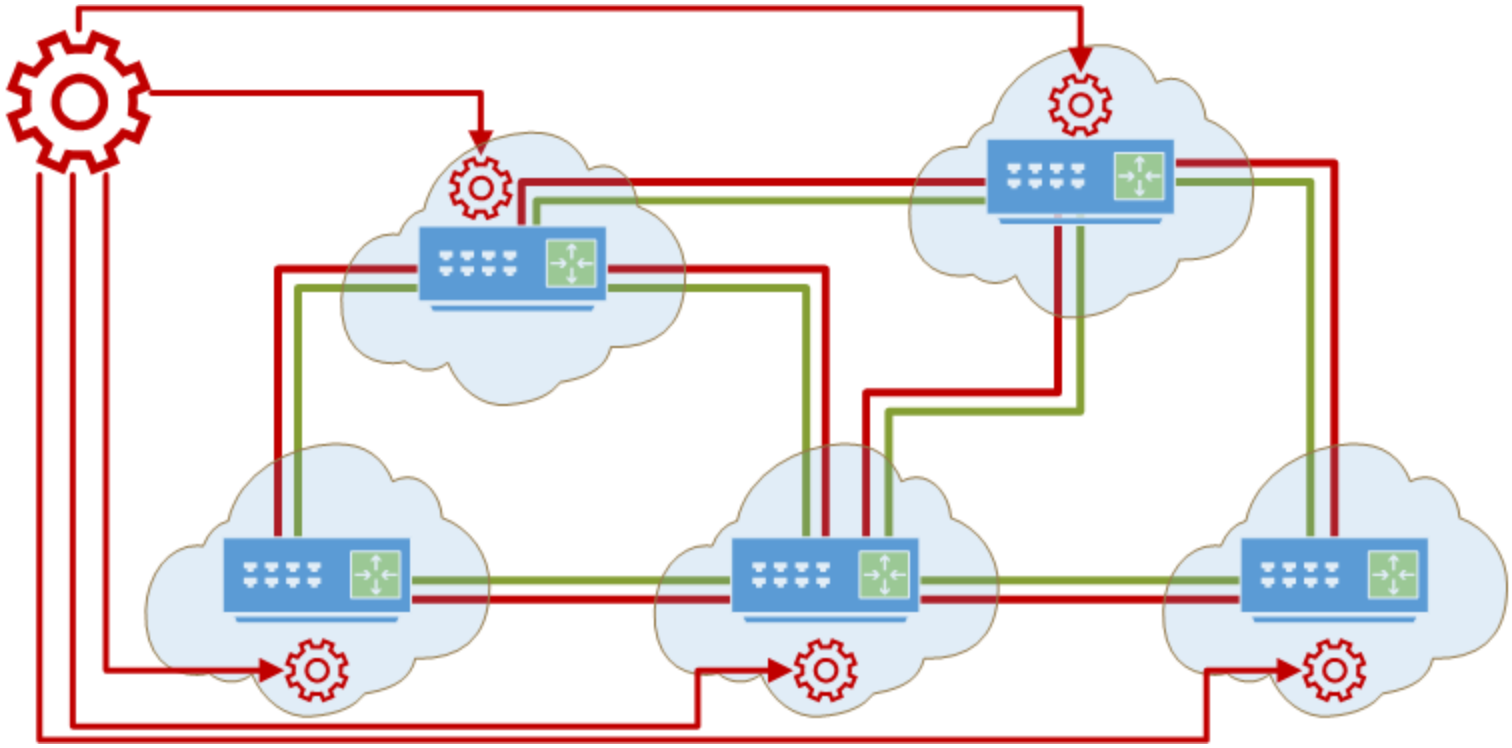
Overcoming BGP's limitations

- Much research in the last 20 years
  - Complement BGP: Bandwidth Brokers



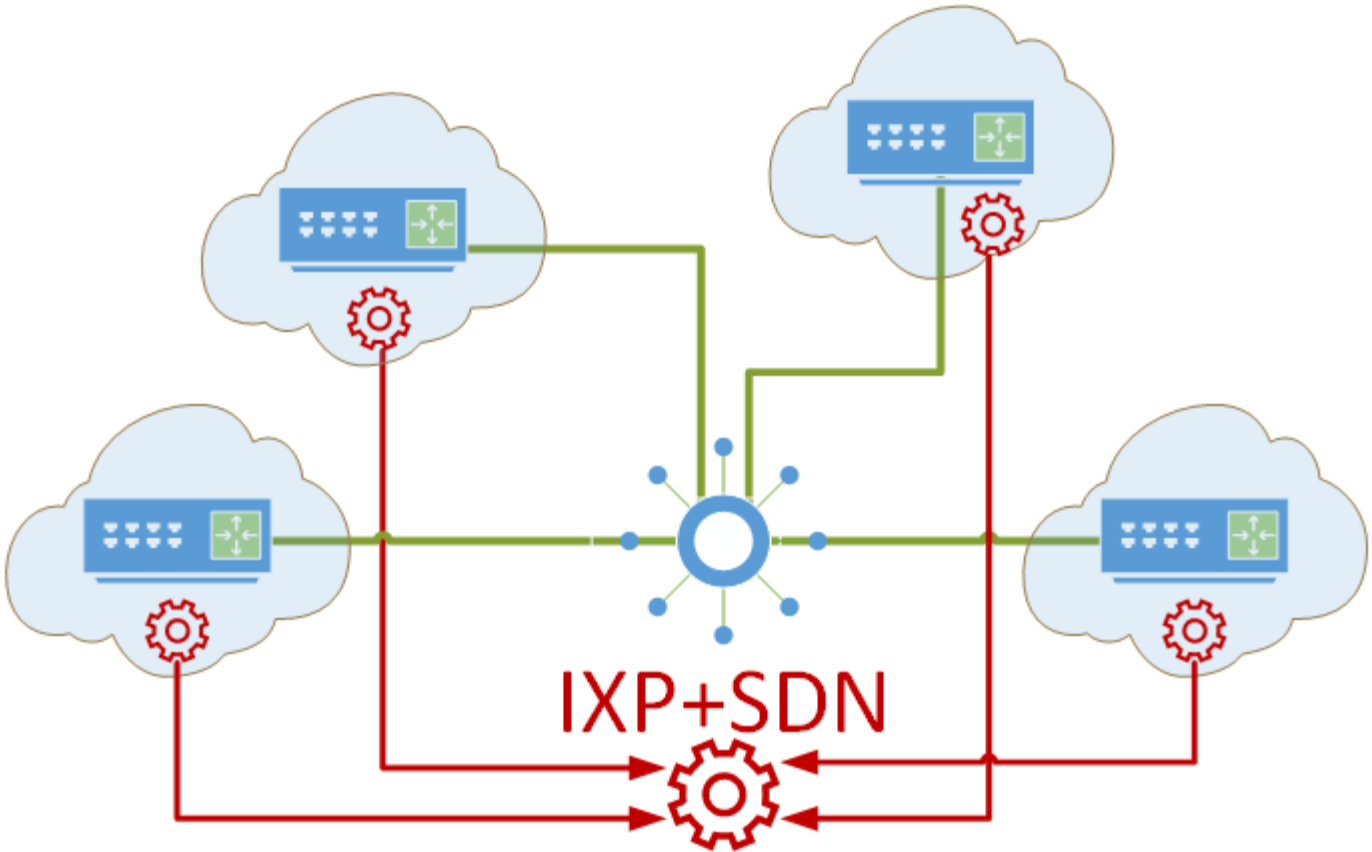
Overcoming BGP's limitations

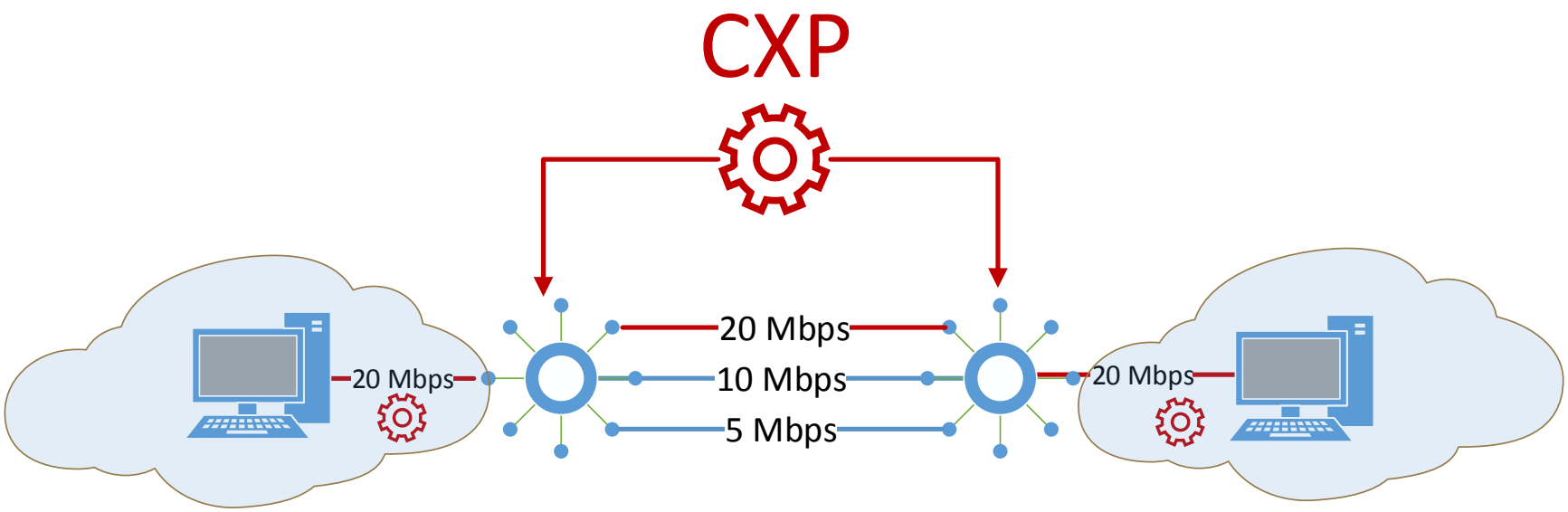
- Much research in the last 20 years
  - Complement BGP: Bandwidth Brokers, Path Computation Elements (PCE)



Overcoming BGP's limitations

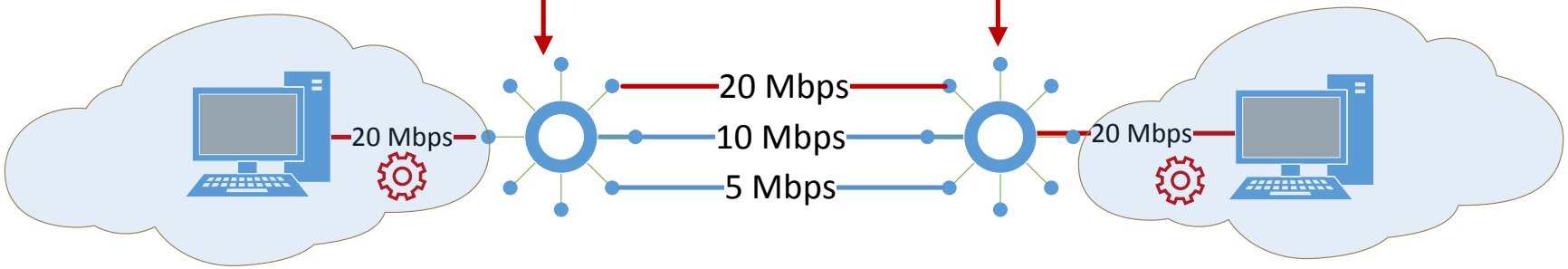
- Much research in the last 20 years
  - Complement BGP: Bandwidth Brokers, Path Computation Elements (PCE), Software-Defined Exchanges (SDX)




CXP
20 Mbps
10 Mbps
5 Mbps
20 Mbps
20 Mbps

CXP

just played around a little with PDF



just played around a little with PDF