Leveraging Performance of Multiroot Data Center Networks by Reactive Reroute

Mitigate congestion by combining spatial and spectral solutions

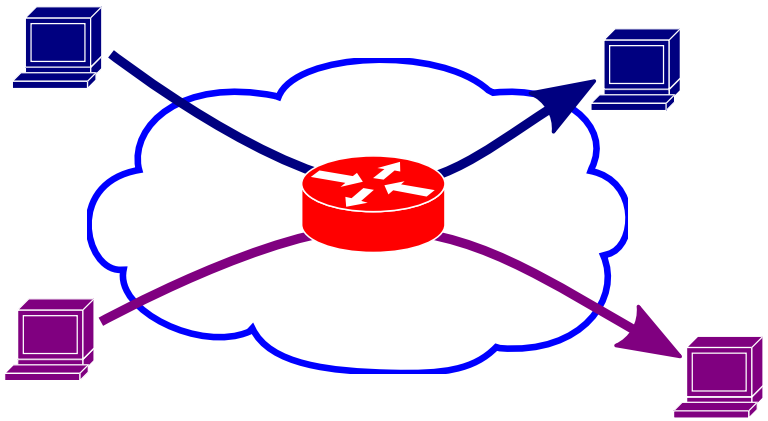
Adrian Sai-wah Tam Kang Xi H. Jonathan Chao

Department of Electrical and Computer Engineering
Polytechnic Institute of New York University

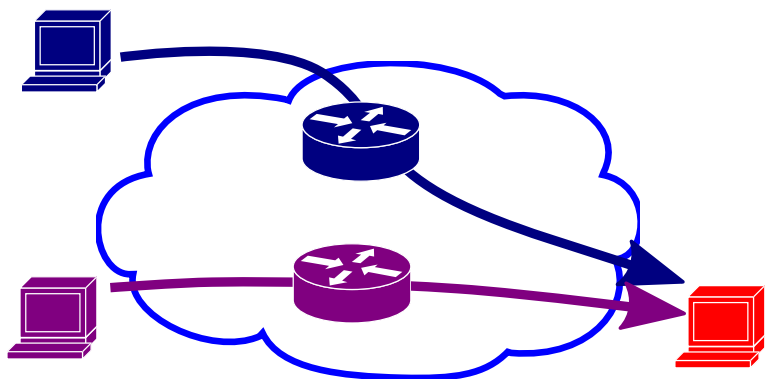
Hot Interconnects 2010

Congestion in Data Center Networks: Why congested?

Flows competing for bandwidth

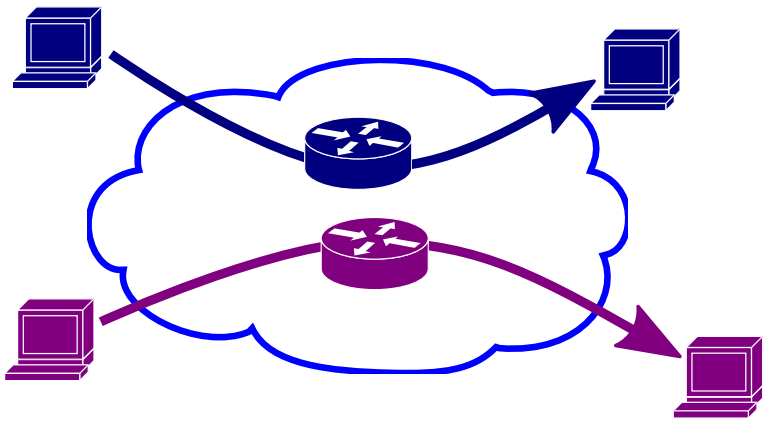


Flows overflowing a host



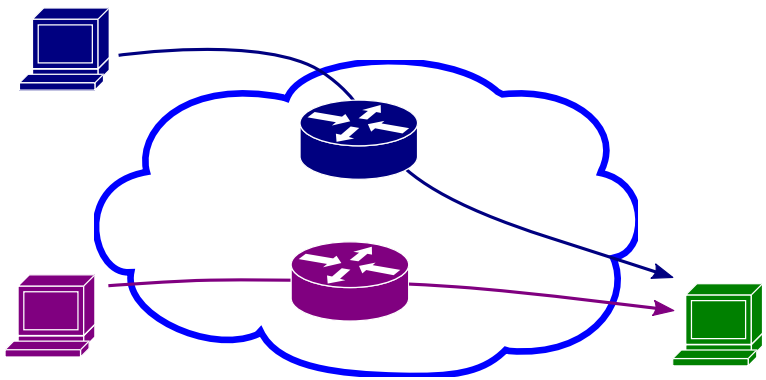
Solution

Flows competing for bandwidth



Change flows' route (spatial)

Flows overflowing a host



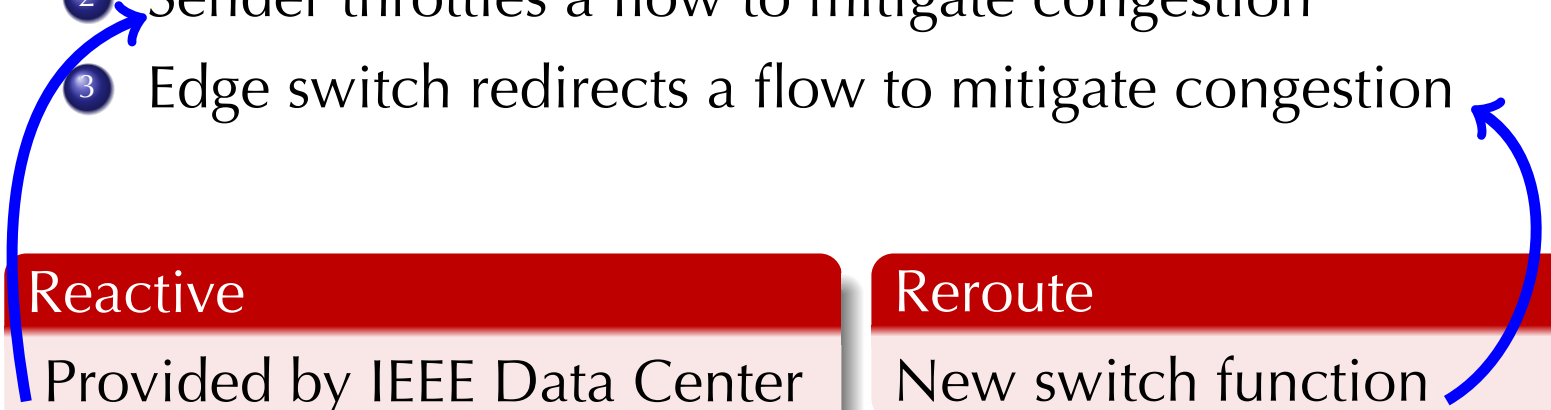
Change flows' bandwidth (spectral)

Reactive reroute as a solution to congestion control

1. When congested, network switch notifies sender
2. Sender throttles a flow to mitigate congestion
3. Edge switch redirects a flow to mitigate congestion

Reactive reroute as a solution to congestion control

1. When congested, network switch notifies sender
2. Sender throttles a flow to mitigate congestion
3. Edge switch redirects a flow to mitigate congestion



Reactive

Provided by IEEE Data Center Bridging Standards

Reroute

New switch function

Reactive

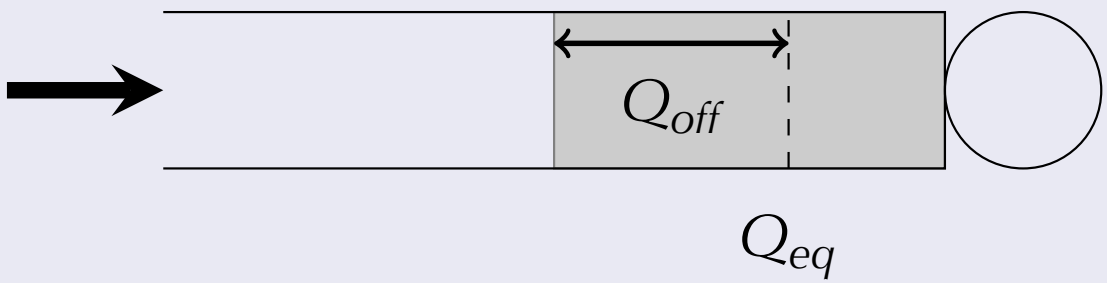
IEEE Data Center Bridging Standards

- IEEE 802.1Qbb: Priority Flow Control
- IEEE 802.1Qau: Congestion Notification

IEEE 802.1Qau

- Link-level Congestion Control
- L2 send rate of a flow reactive to congestion
- RP: Reactive Point, i.e. NIC in hosts
- CP: Congestion Point, i.e. queues in switches
- QCN: Quantified Congestion Notification, created by CP and received by RP
- Goal: Maintain usage at Q_{eq}

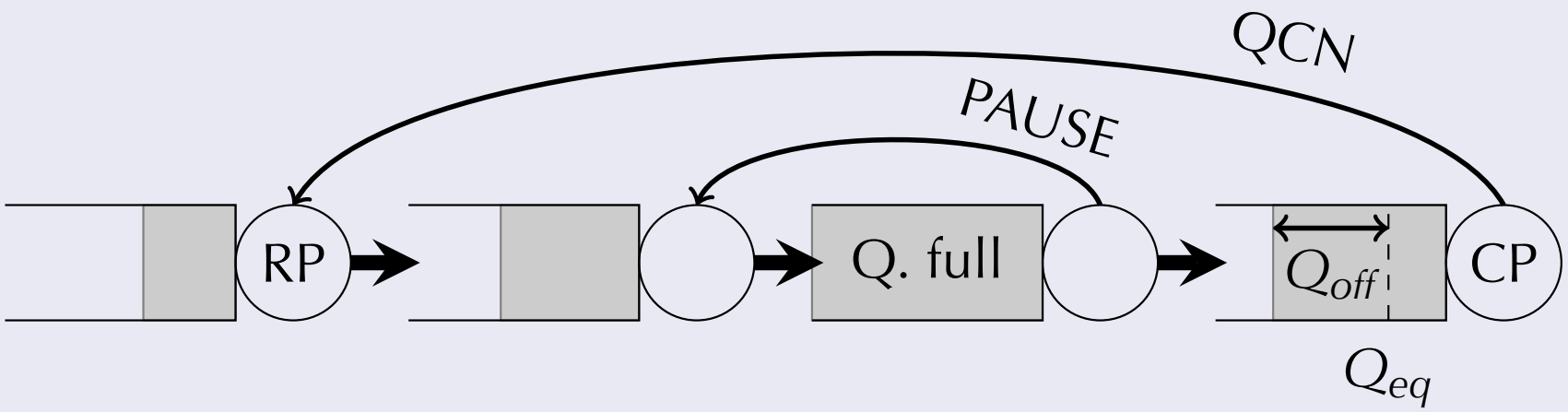
Queue at CP



IEEE 802.1Qbb

- Link-level Flow Control
- Similar to stop-and-go
- PAUSE to upstream to prevent overflow

Queues and Pause

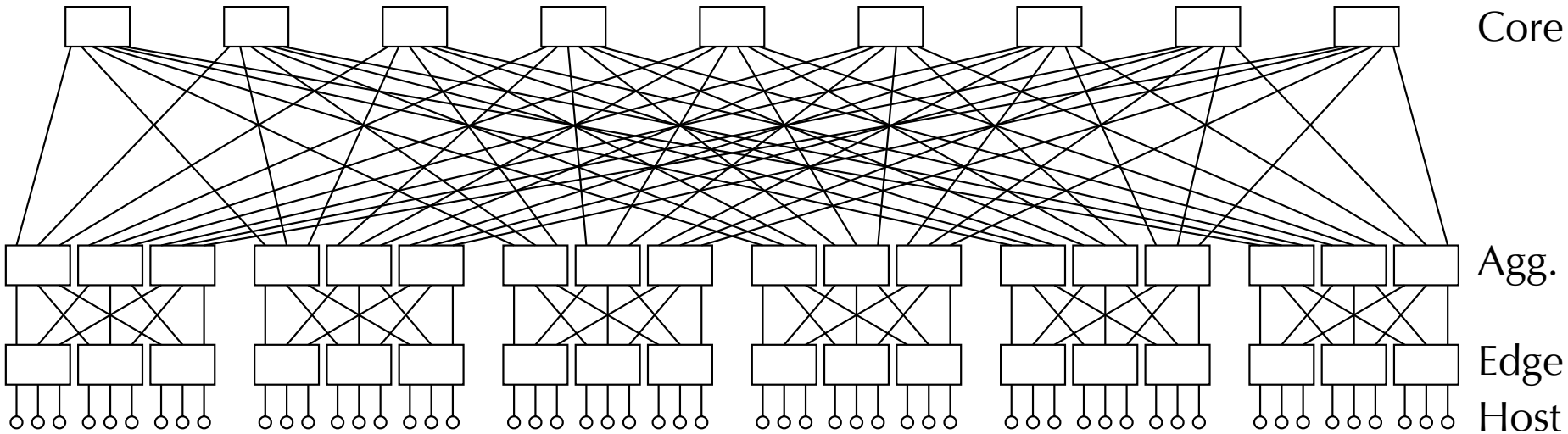


QCN: Quantified Congestion Notification

CP: Congestion Point (output buffer) RP: Reaction Point (host NIC)

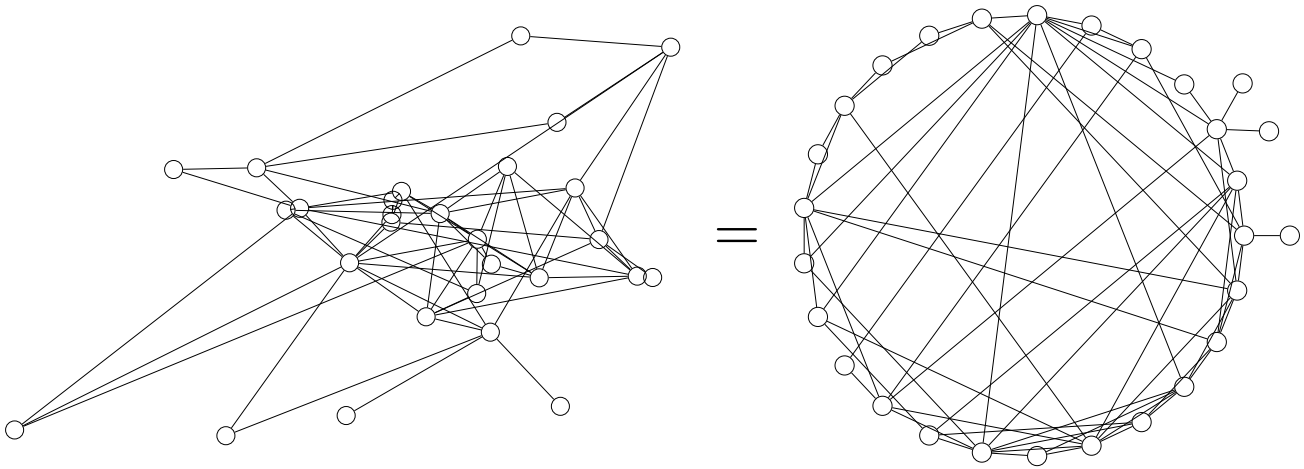
Reroute

Fat-tree Topology



- Fat-tree has a lot of redundant paths from any host to another
- Can we exploit the multipath to mitigate congestion?

General Topology



(topology source: Rocketfuel)

- Irregular, mesh-like topologies
- Also a lot of redundant paths
- Can we exploit the multipath to mitigate congestion?

DCN with Multipath Routing

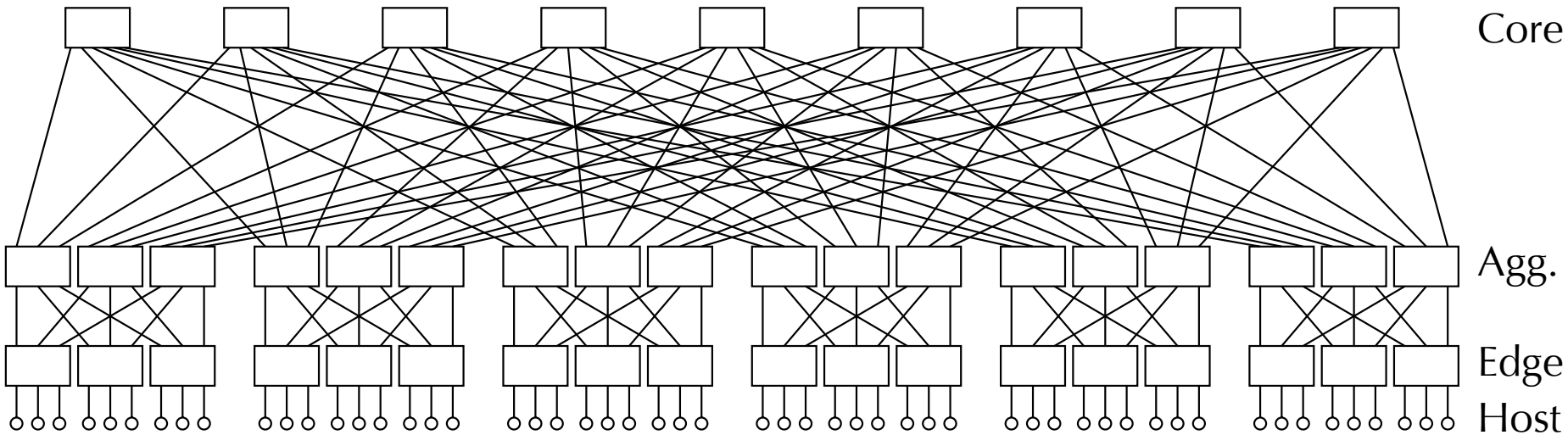
- Assume IEEE 802.1Qau/bb support
- Multipath routing for flows at all switch

DCN with Multipath Routing

- Assume IEEE 802.1Qau/bb support → Spectral
- Multipath routing for flows at all switch → Spatial

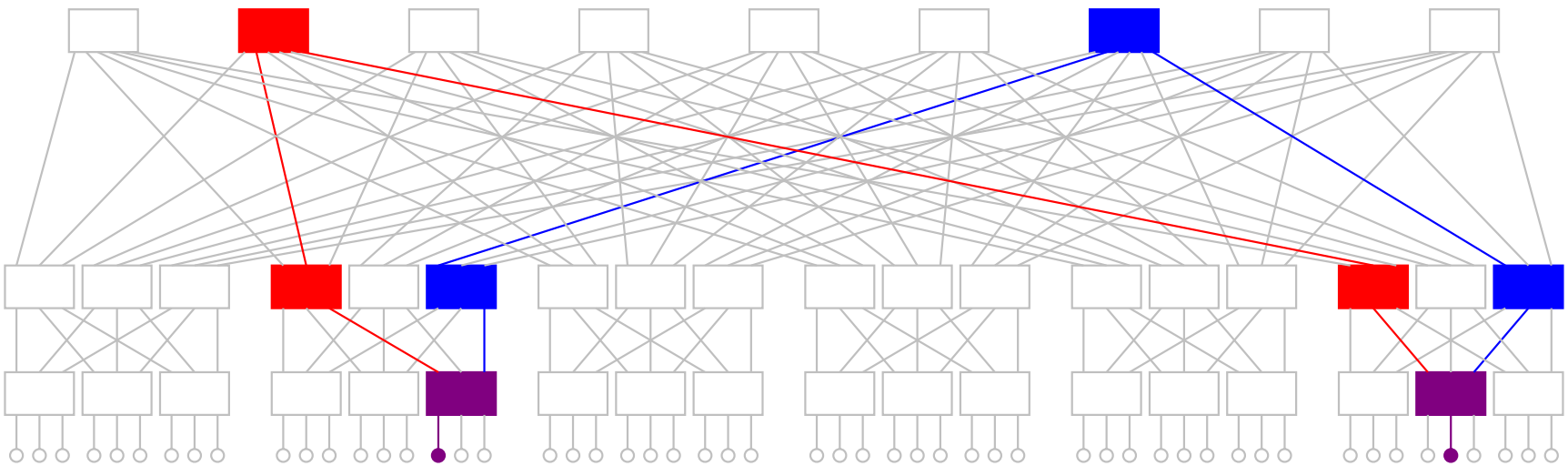
Routing in Fat-tree

- Edge: Hash-based, Flow-based, Destination-based routing
- Aggr & Core: Hash-based, Destination-based routing
- "Downward": Always destination-based
  "Upward": Hash-based and optionally flow-based



Routing in Fat-tree

- By default, hash-based routing unless destination is known
- Destination is known only for “downward” routing
- Hash the flow (e.g. 5-tuple) into output ports to randomize the next hop



Reaction to QCN

- Output port randomizing $\neq$ Load balancing
  ($\because$ Uneven flow size)
- Congestion may occur: IEEE 802.1Qau in action
- Edge switches are unique to hosts, $\therefore$ must see the QCNs

- Implement flow-based routing on edge switches to reroute upon congestion

Edge Switch Forwarding Algorithm

procedure route(Packet P)
  if route for P is found in destination-based routing table **then**
    $\nu \leftarrow$ output port according to dest-based table
  else if route for P is found in flow-based routing table **then**
    $\nu \leftarrow$ output port according to flow-based table
    update last encounter time of this flow in flow-based table
  else
    $\nu \leftarrow \text{Hash}(P)$
  end if
  if P is a congestion notification **then**
    if P is found in congestion signal record **then**
      increase the count in the record
    else
      create a new entry in the record with count=1
    end if
    if count in congestion signal record $\geq$ threshold **then**
      reset the count
      reroute the flow by updating flow-based table
    end if
  end if
  Send P to output port ν
end procedure

Edge Switch Forwarding Algorithm

- Upon congestion, edge switch receives QCNs
- When enough number of QCN is received for some flow, it is worth to reroute because it is big enough to have some impact

Edge Switch Forwarding Algorithm

- Upon congestion, edge switch receives QCNs
- When enough number of QCN is received for some flow, it is worth to reroute because it is big enough to have some impact
- Flow table to remember the reroute
- Erase the flow entry after some time of inactivity

Reroute

Different ways of reroute

1. Degenerated case: Do not reroute
2. Uniform random output port selection
3. Select the output port of minimum likelihood of congestion
4. Weight random output port selection (combine of above two)

Reroute

Different ways of reroute

1. Degenerated case: Do not reroute
2. Uniform random output port selection
3. Select the output port of minimum likelihood of congestion
4. Weight random output port selection (combine of above two)

Likelihood is estimated by an edge switch based on the QCNs it received

More on rerouting

- Because of rerouting, packets out-of-order is expected, but the amont of packets out-of-order is limited because of small buffer and high speed of DCN switches
- Route flapping might happen but controlled by:
  - Threshold of the number of QCNs to reroute a flow
  - Frequency of creating QCNs
  - A route-freeze timer to prevent a flow rerouted twice in a short time

Evaluation

Framework

- NS-3 simulation
- Random sending rate, Random sender-receiver, Admissible traffic
- Load = the agg. sending rate as link speed %
- Fat-tree: 10-port switches with 1Gbps links

Framework

- NS-3 simulation
- Random sending rate, Random sender-receiver, Admissible traffic
- Load = the agg. sending rate as link speed %
- Fat-tree: 10-port switches with 1Gbps links

Goal

How much improvement does reactive reroute provide?

Throughput

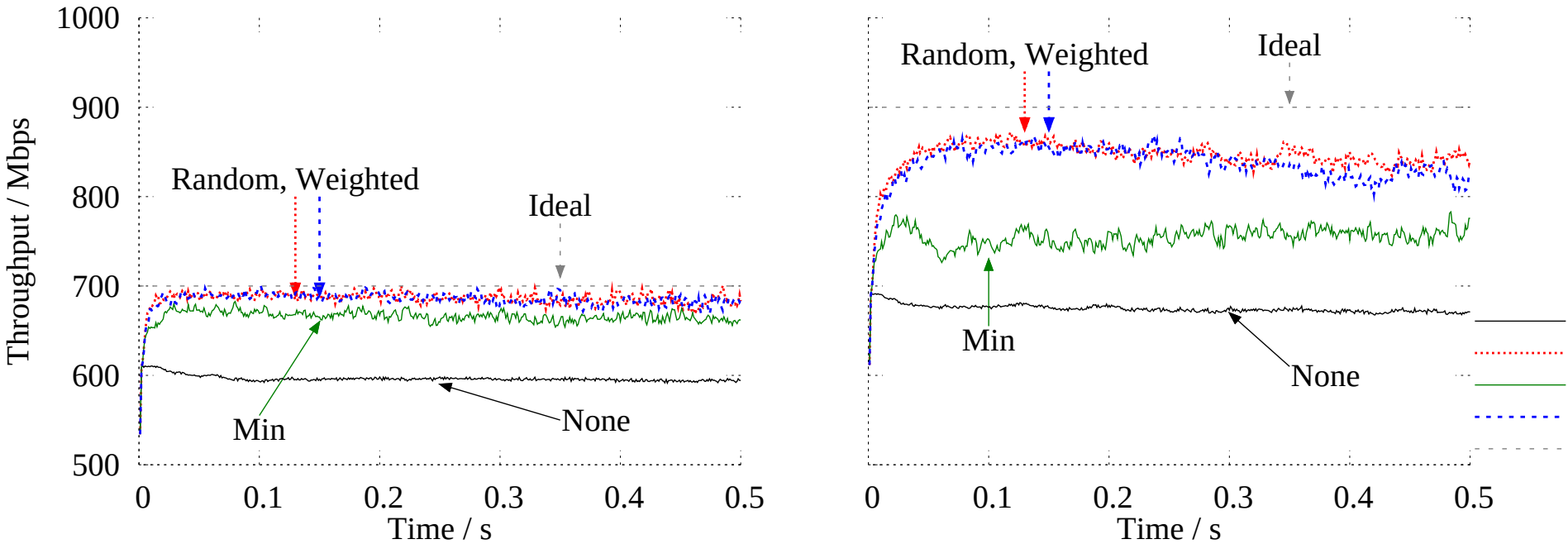


Figure: Average throughput per link of edge switch sending to host at (left) 70% load and (right) 90% load, UDP traffic

Latency

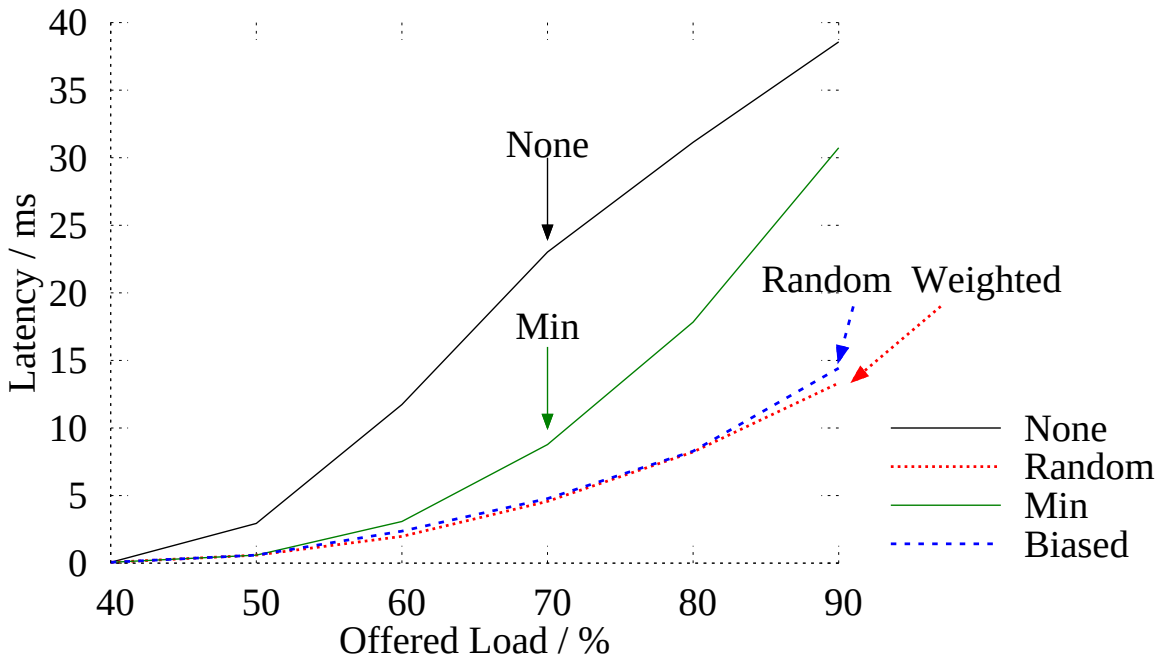


Figure: Mean link layer latency vs load

Queue length

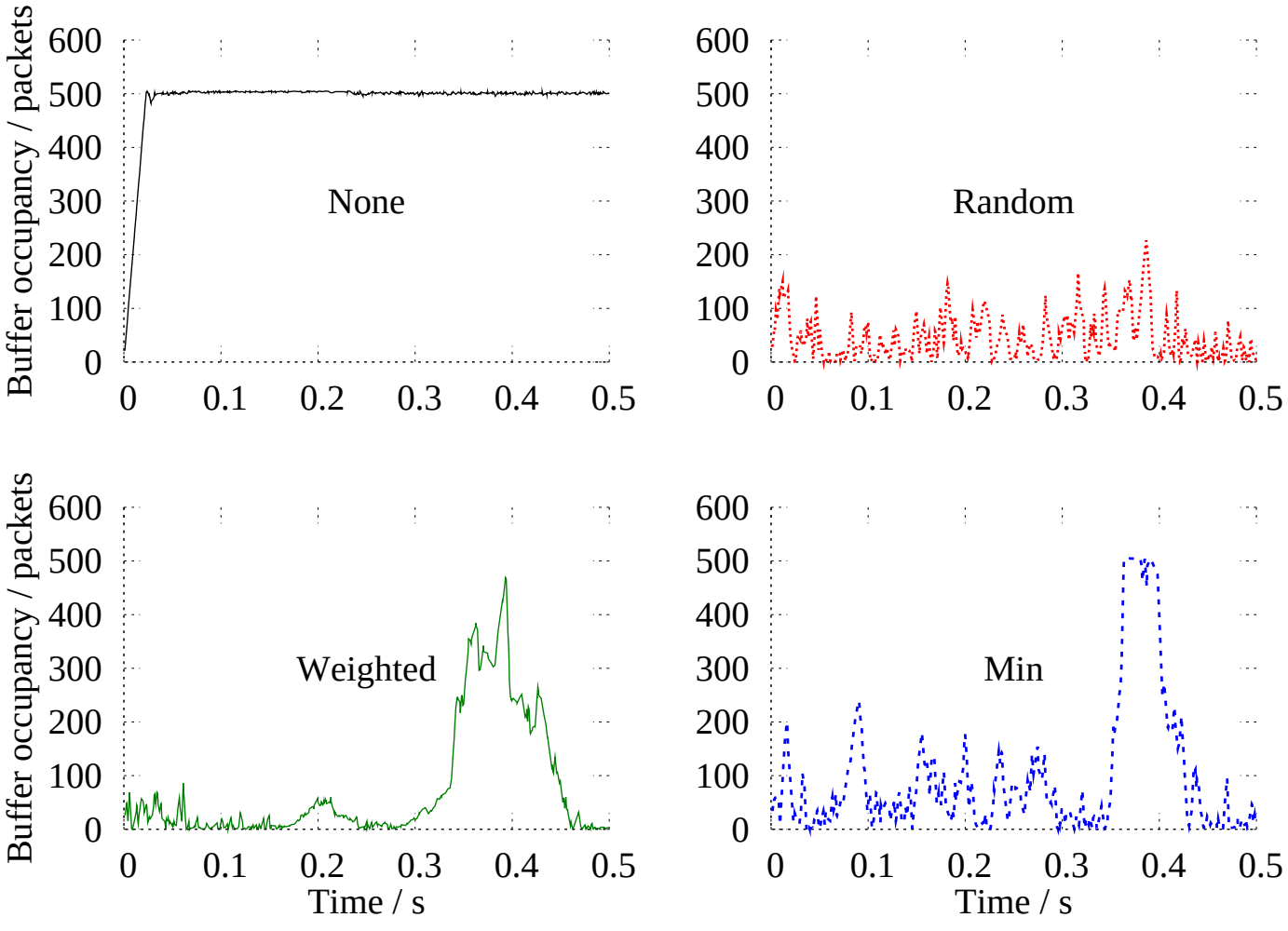


Figure: Buffer occupancy under 70% load

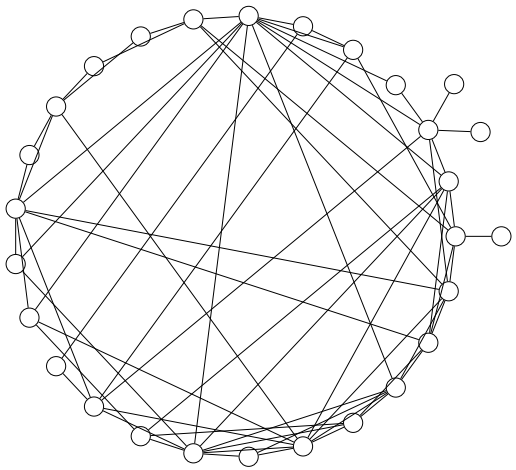
Conclusion

Reroute vs Performance

- Reactive reroute significantly improves performance
- Different reroute strategies are evaluated. Amongst, uniform and weighted random give best performance

End

How about irregular networks?

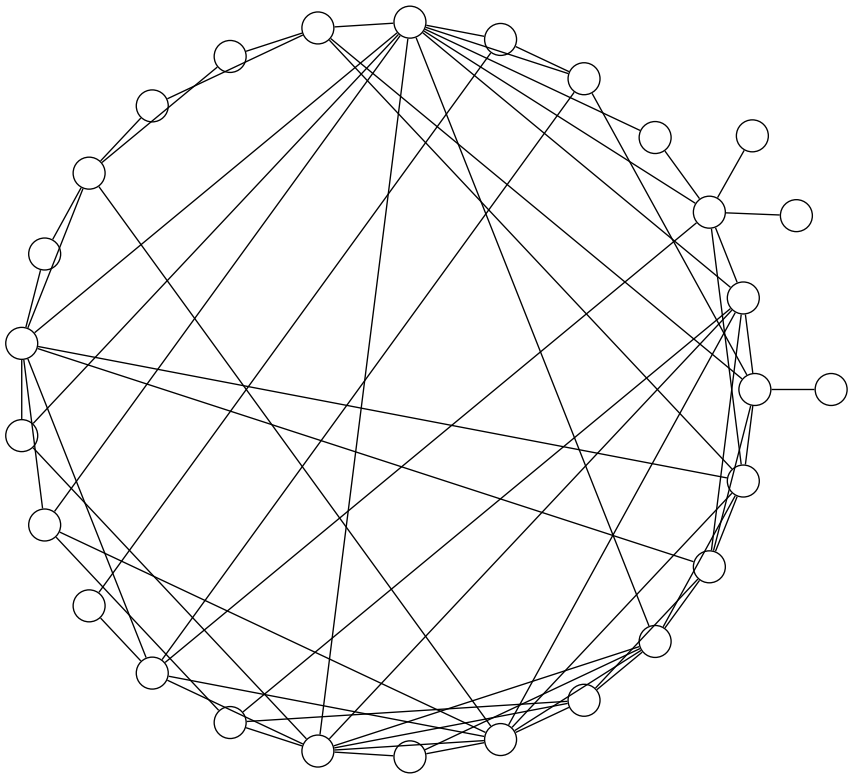


- Find multiple distict spanning trees
- Send packets on different tree → different routes
- Implemented using IEEE 802.1Q VLAN

Edge Switch in Irregular Networks

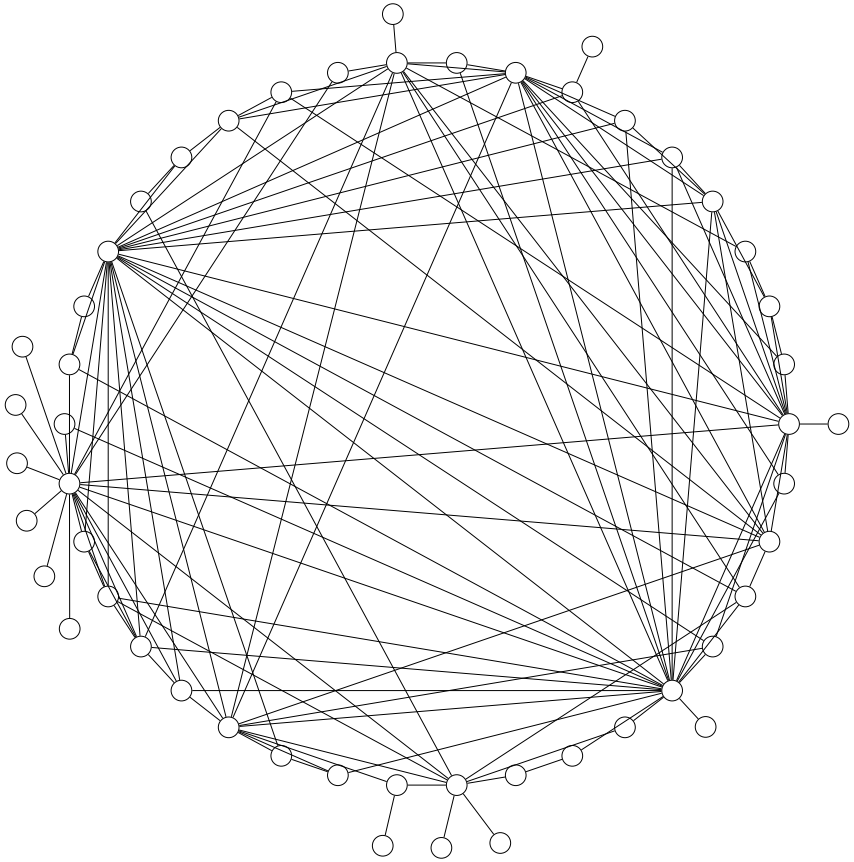
- Every switch is an edge switch to the directly-attached hosts
- Edge switch
  1. prepend VLAN header to packets sent by their hosts
  2. capture QCNs toward their host
  3. reroute flows orginated from their hosts by changing the VLAN ID

Topologies



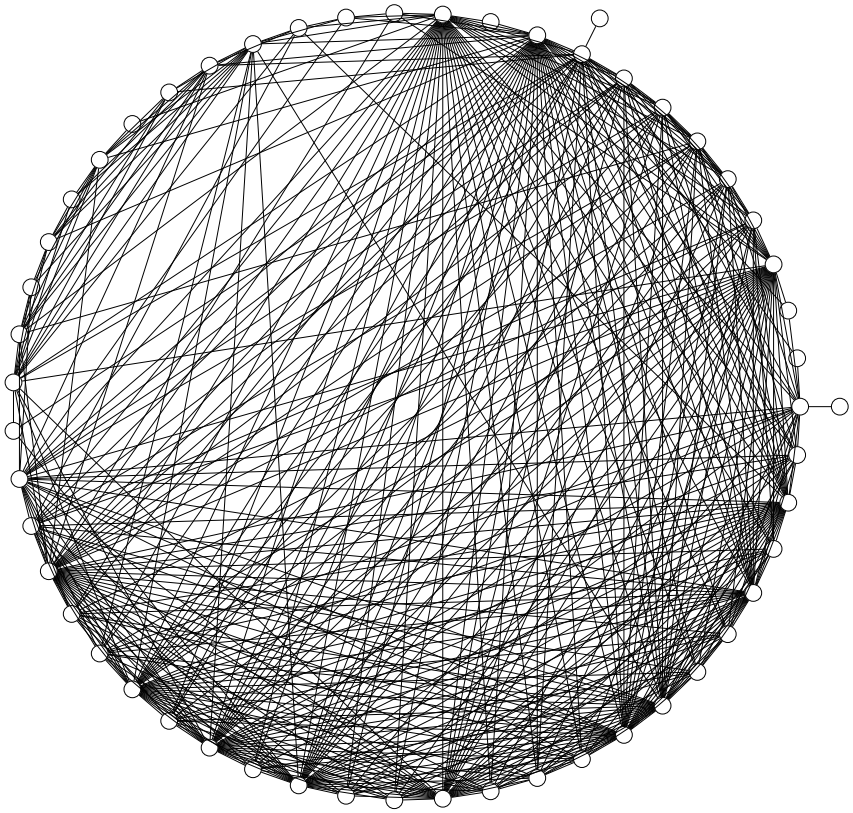
Network E
Small scale, low node degree

Topologies



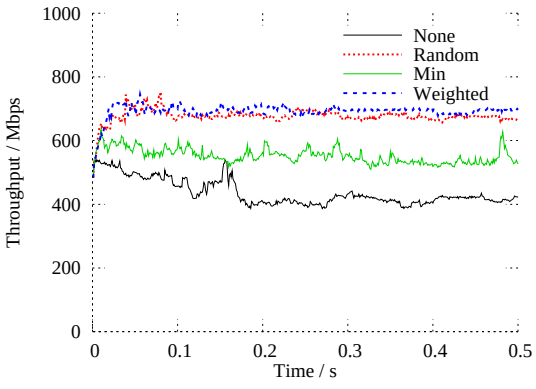
Network T
Many degree-1 nodes

Topologies

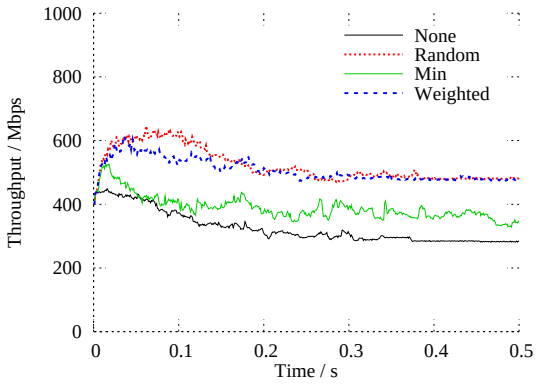


Network L
Large network with high node degree

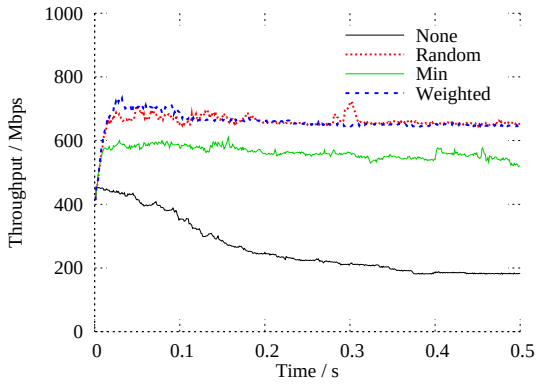
Throughput



Network E



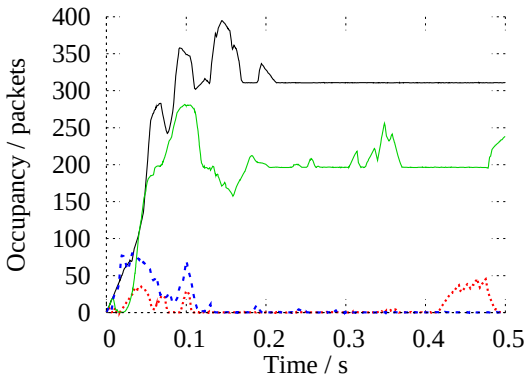
Network T



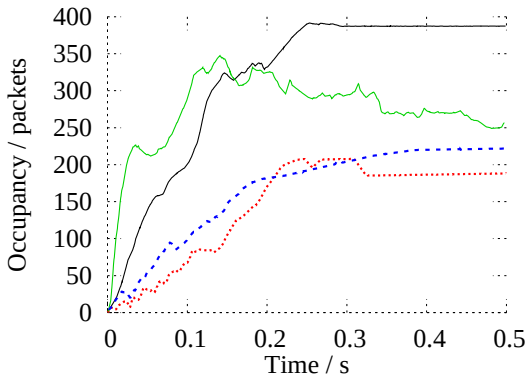
Network L

Figure: Average throughput per link of edge switch sending to host at 70% load

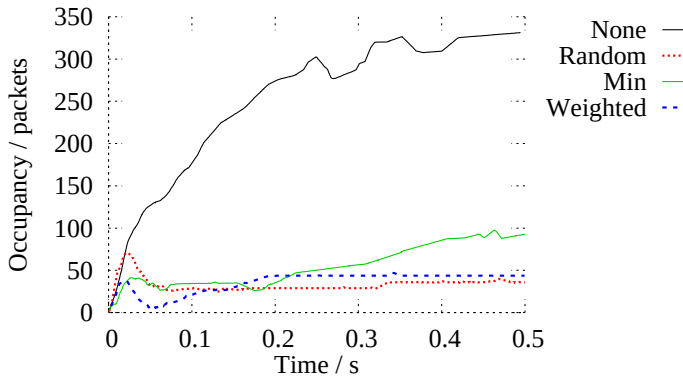
Queue length



Network E



Network T



Network L

Figure: Buffer occupancy of a switch vs time, under 70% load

Topology

- Regular topology gives better performance, because it is less likely to have bottlenecks
- Node degree is not a factor to performance, as long as multipath exists (i.e. not degree-1 nodes) for most routes
  - Network E has average node degree of 4 only

Reactive Reroute Solution

- Level-2 solution
- Exploit the high speed, low latency nature of DCN
- Solve the congestion problem in **two** dimensions: *spatial* and *spectral*