Dragonfly+: Low Cost Topology for Scaling Datacenters

A. Shpiner, Z. Haramaty, S. Eliad, V. Zdornov, B. Gafni and E. Zahavi

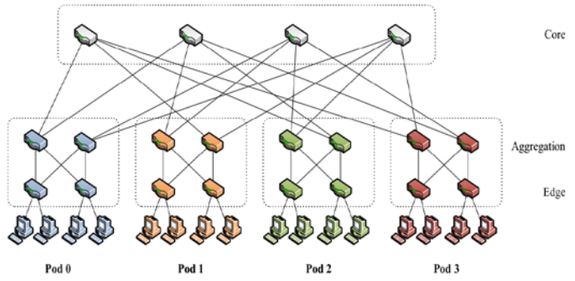
HiPINEB, 2017

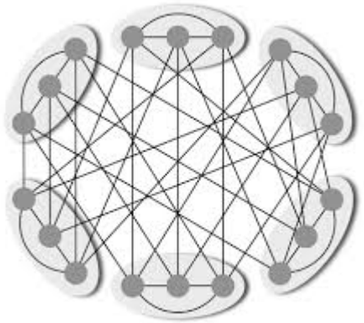
Mellanox TECHNOLOGIES

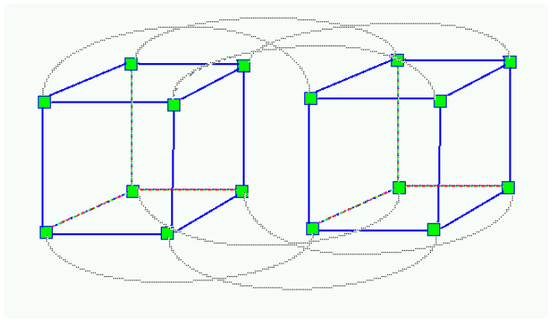
Connect. Accelerate. Outperform.™

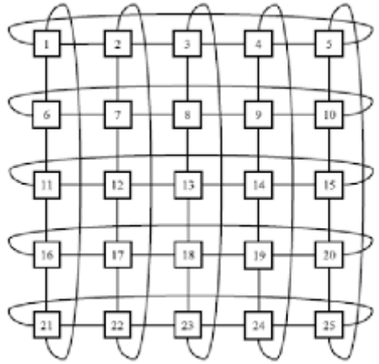
Introduction

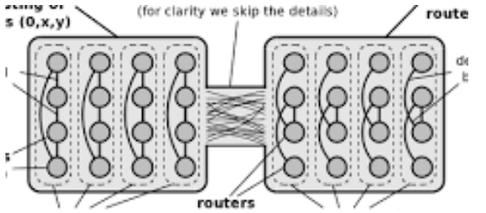
- HPC clusters rely on interconnect network to communicate among processes.
- Network topology impacts the application performance.
- Common topologies: Fat Tree, Dragonfly, Hyper-cube, Torus, SlimFly
- Keys to topology evaluation:
  - Network throughput - for various traffic patterns.
  - Network diameter – min/average/max latency between end-hosts.
  - Scalability – cost of adding new end-hosts
  - Cost per end host – number of network routers/ports per end-host.
  - And more …





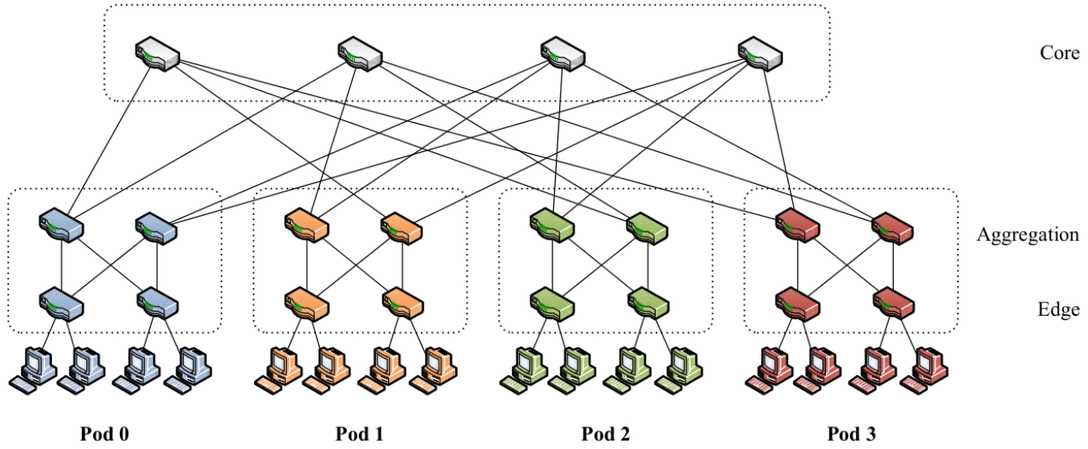






Fat tree

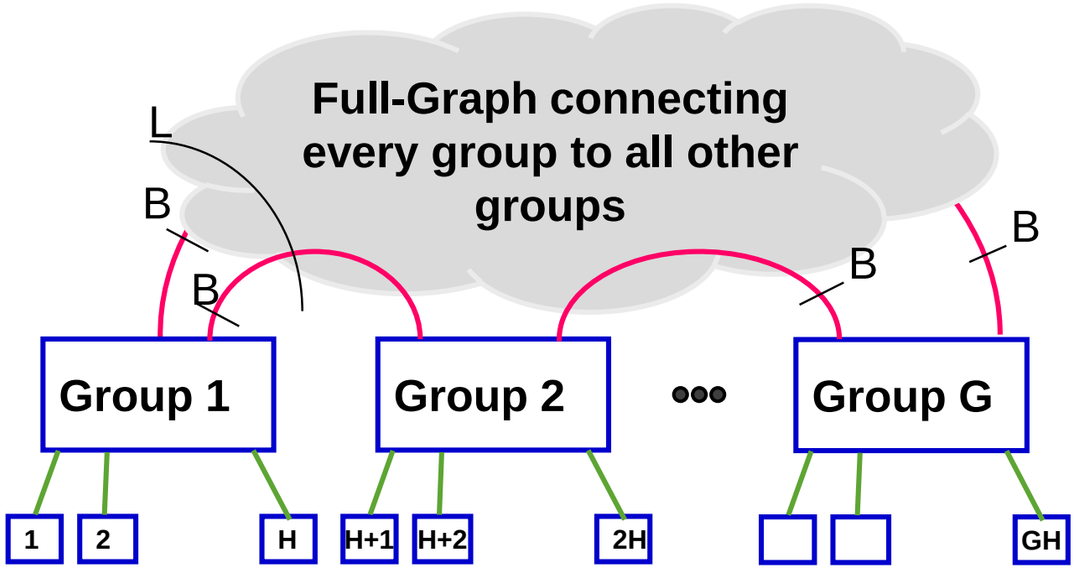
- Most common topology.
- Pros:
  - Maximal network throughput for a variety of traffic patterns with relatively simple routing.
  - Scalable.
  - Fault-tolerant through its path diversity.
  - Credit loop deadlock free routing without additional resources (virtual lanes).
- Cons:
  - Large diameter
  - Relatively costly due to the large amount of routers and links.

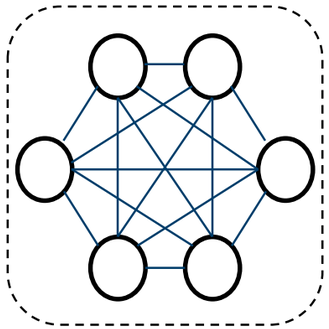


Dragonfly [Kim et al., 2008]

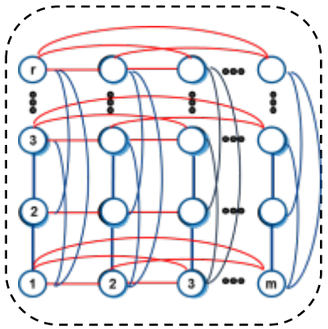
- State-of-the-art in HPC
- Focus on reducing the number of long links and network diameter.

- Hierarchical topology dividing hosts to groups
- Groups are all-to-all connected.
  - Each group has at least one direct link to other group.
- Dragonfly flavors diverge on group topology
  - The default recommendation is 1D flattened butterfly.

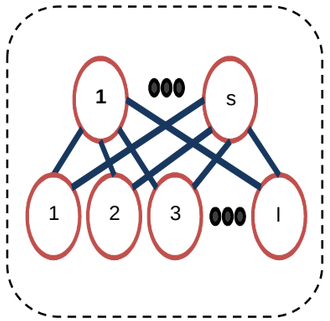




1D flattened butterfly
(completely connected)



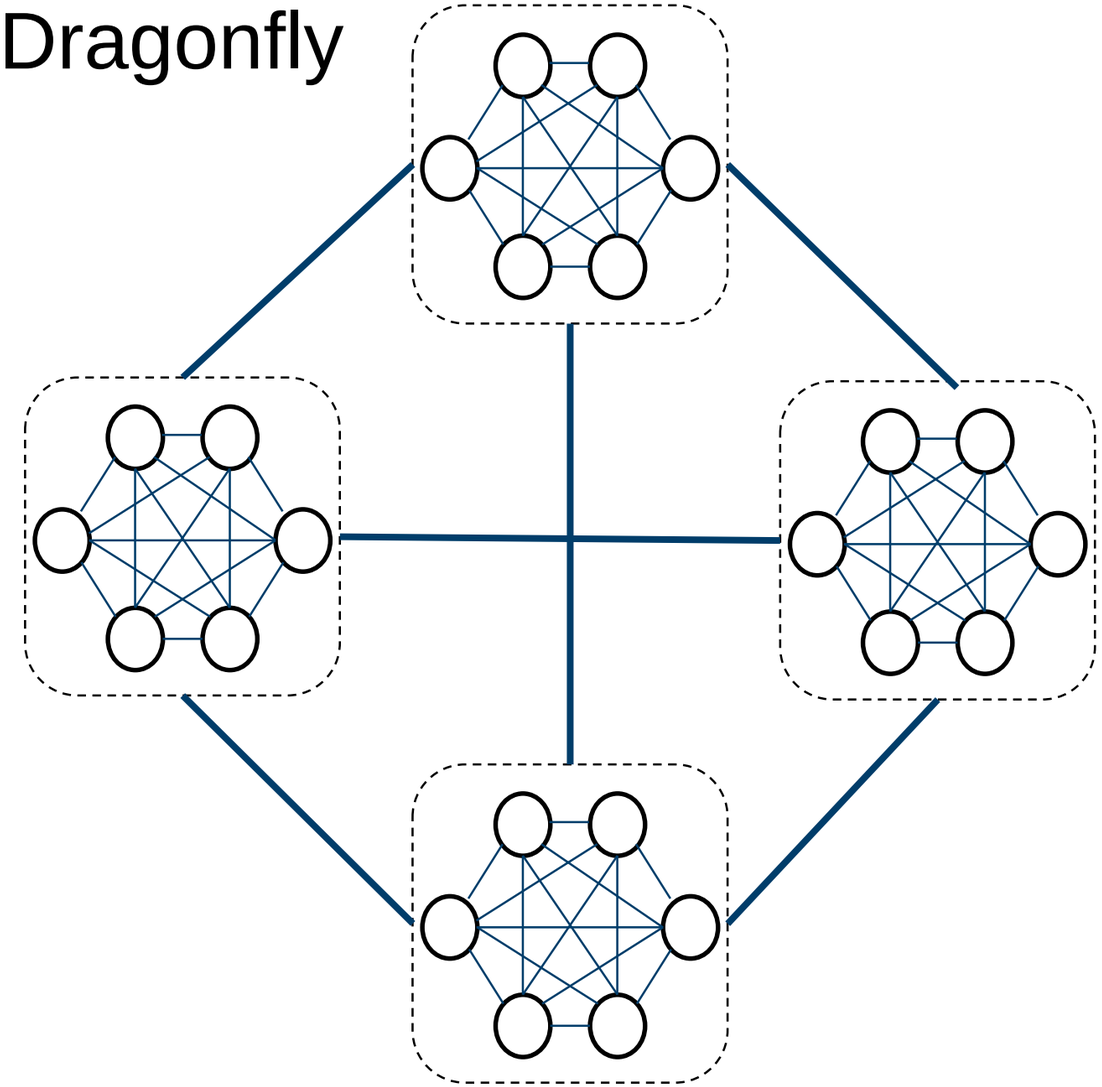
2D flattened butterfly



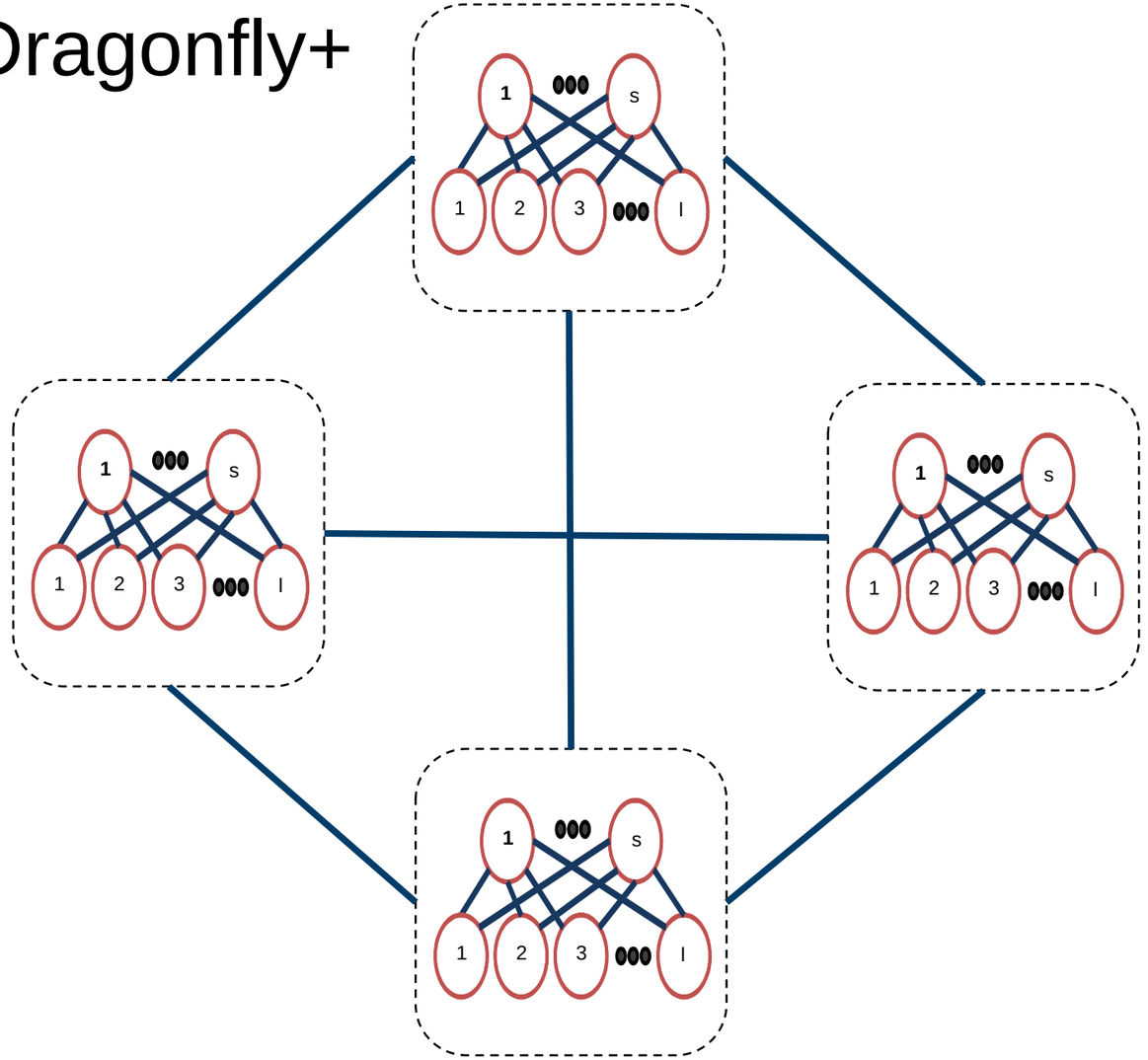
Dragonfly+
(bi-partite)

Dragonfly vs Dragonfly+

Dragonfly



Dragonfly+



Dragonfly+ group

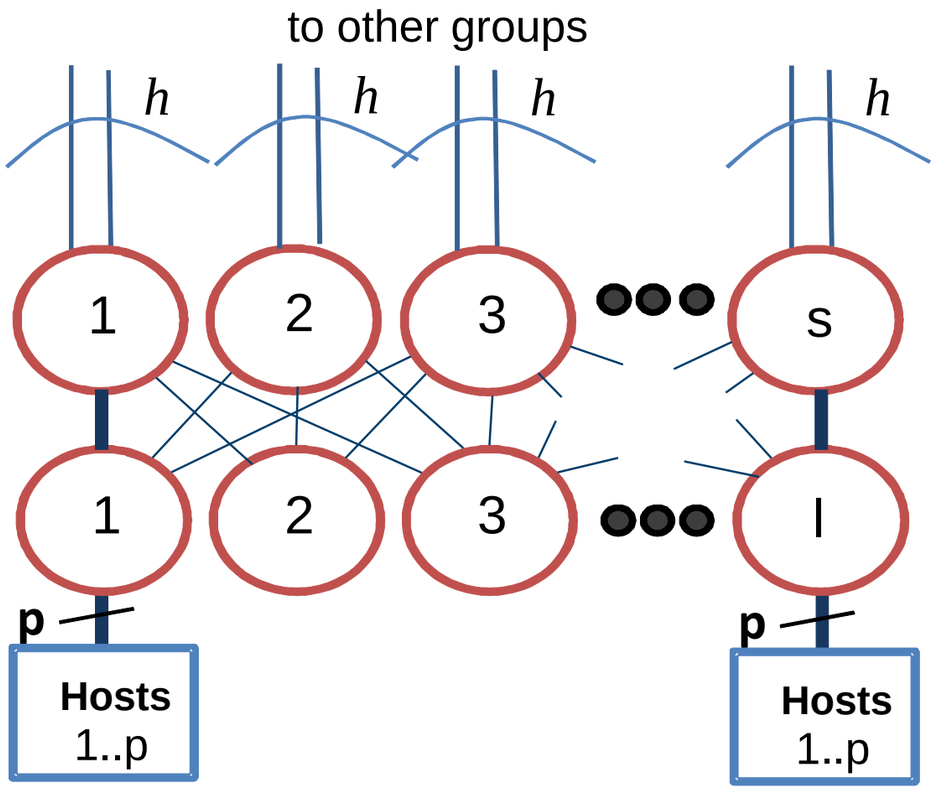
- Dragonfly+ intra-group is connected in full bipartite manner.
  - Leaf router is connected to p hosts and s spine routers.
  - Spine router is connected to l leaf routers and to h spine routers of other groups.

- For keeping full bi-sectional bandwidth inside the group:

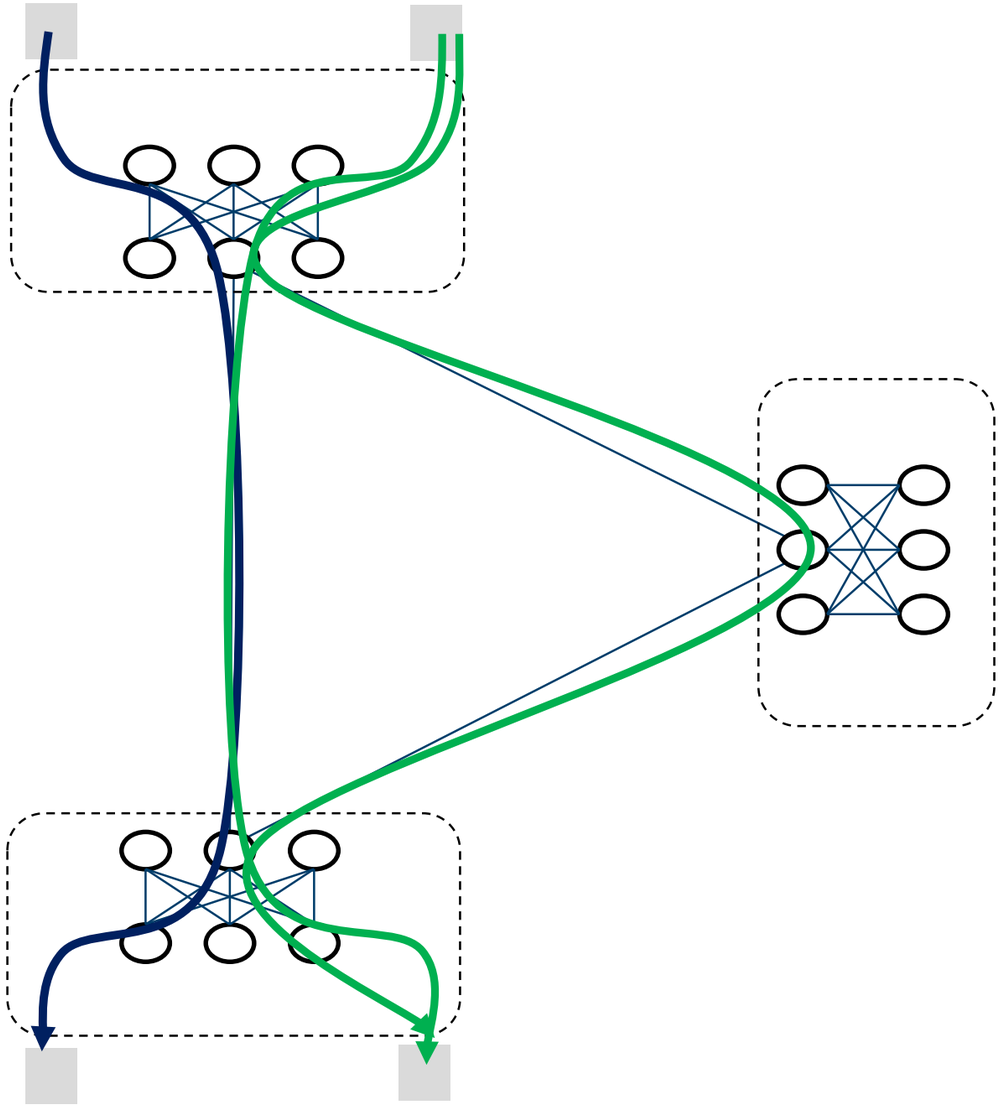
$$p = l = s = h$$

- Router radix k defines the number of hosts in the group:

$$N_{group} = pl = {k^2}/{4}$$

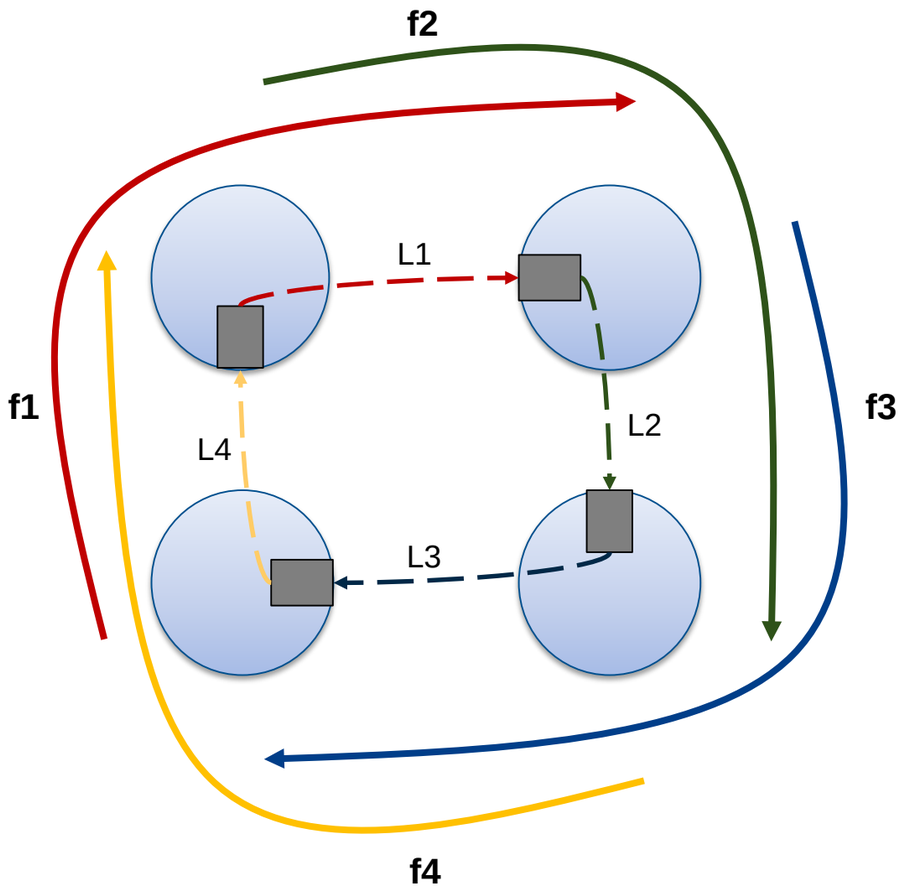


Multi-Path Non-Minimal Routing



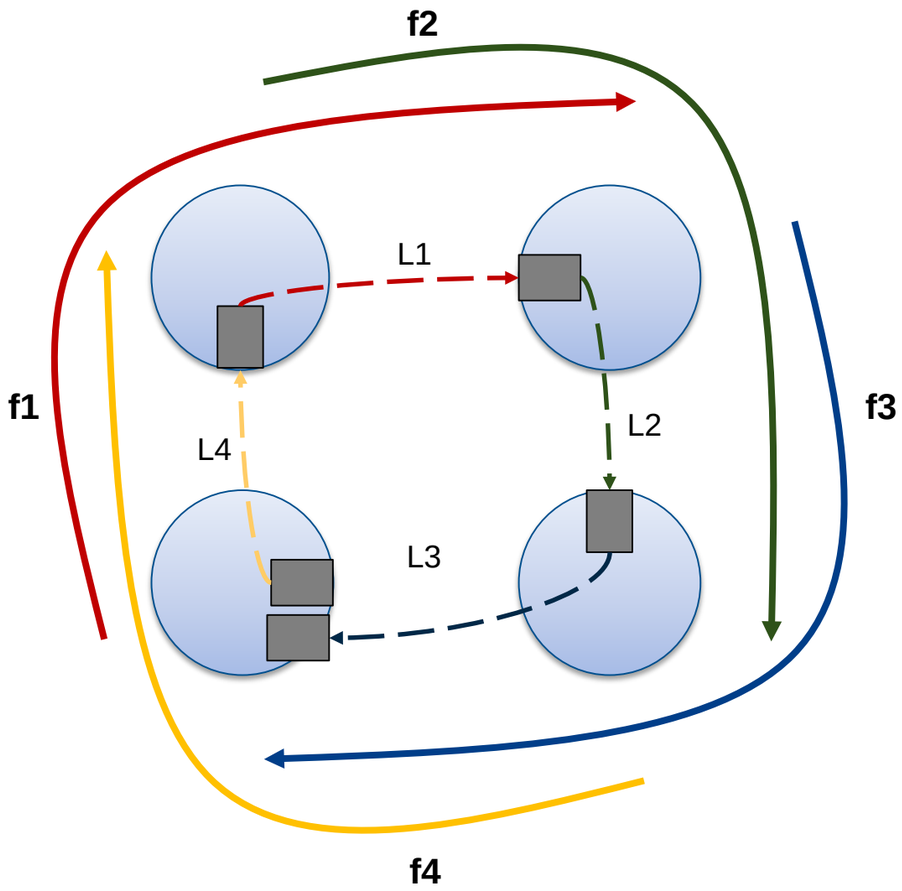
- To obtain maximal throughput for various traffic patterns *non-minimal multi-path routing* is required.
- *Fully Progressive Adaptive Routing (FPAR)* with *Adaptive Routing Notification (ARN)* messages.
- FPAR extends previously known approaches
  - Vs. UGAL-L [Singh, 2005]: defined two additional route priorities over the minimal
  - Vs. PAR [Jiang et al., 2009] : decision in every router
  - Vs. CRT and PB [Jiang et al., 2009] : faster convergence with ARN
  - Vs. OFAR [Garcia et al., 2012]: using VL number instead of proprietary bits

Credit Loop Deadlock



- Given topology, routing rules and traffic demand, *cyclic sequence* of router buffers, such that every router in the sequence sends traffic to the next router in the sequence.
- Credit loop locks when buffers overflow.

Credit Loop Deadlock

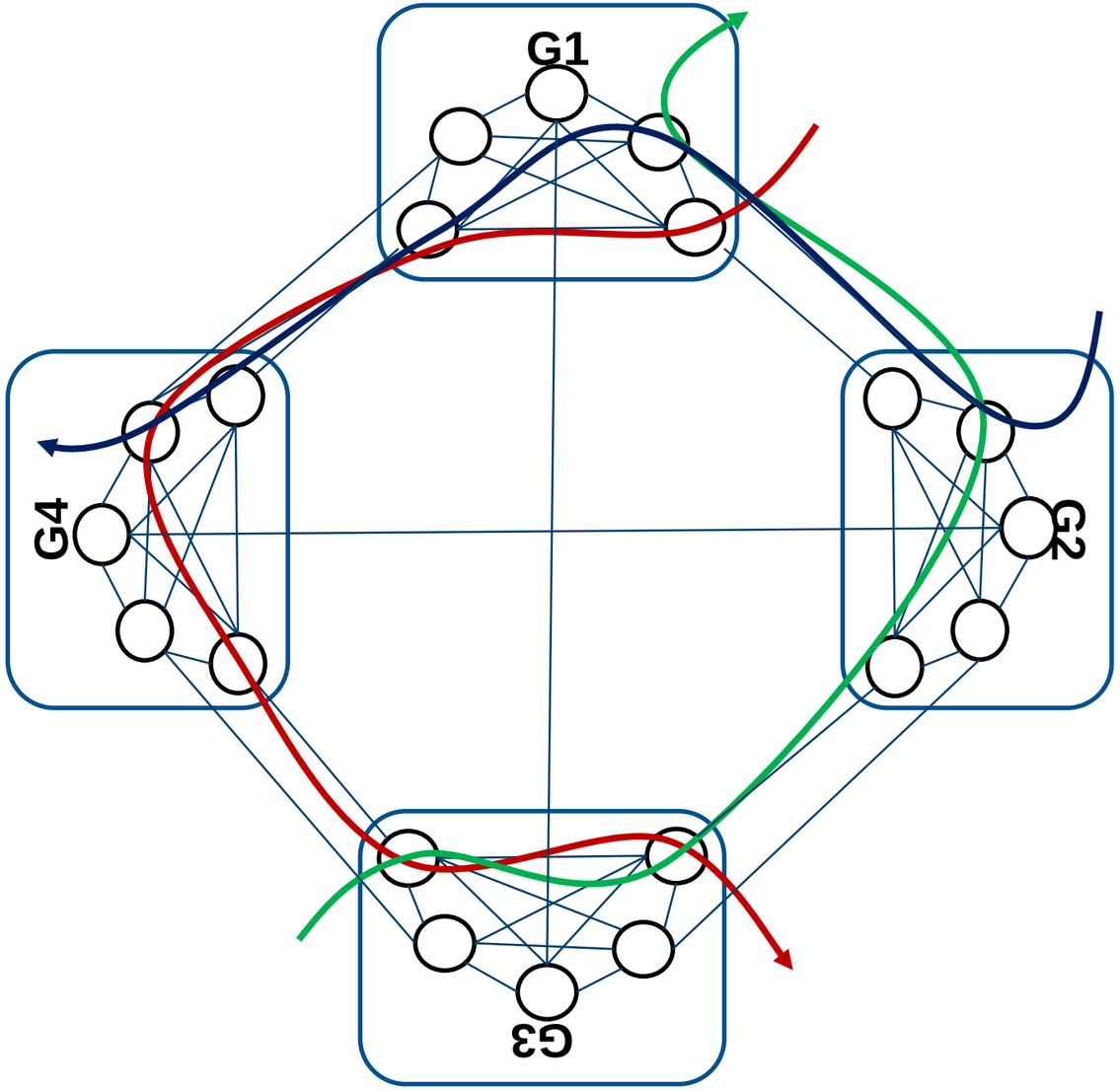


- Given topology, routing rules and traffic demand, *cyclic sequence* of router buffers, such that every router in the sequence sends traffic to the next router in the sequence.
- Credit loop locks when buffers overflow.

- Solution: change virtual lane (VL) in specific points

Credit Loop Deadlock in Dragonfly

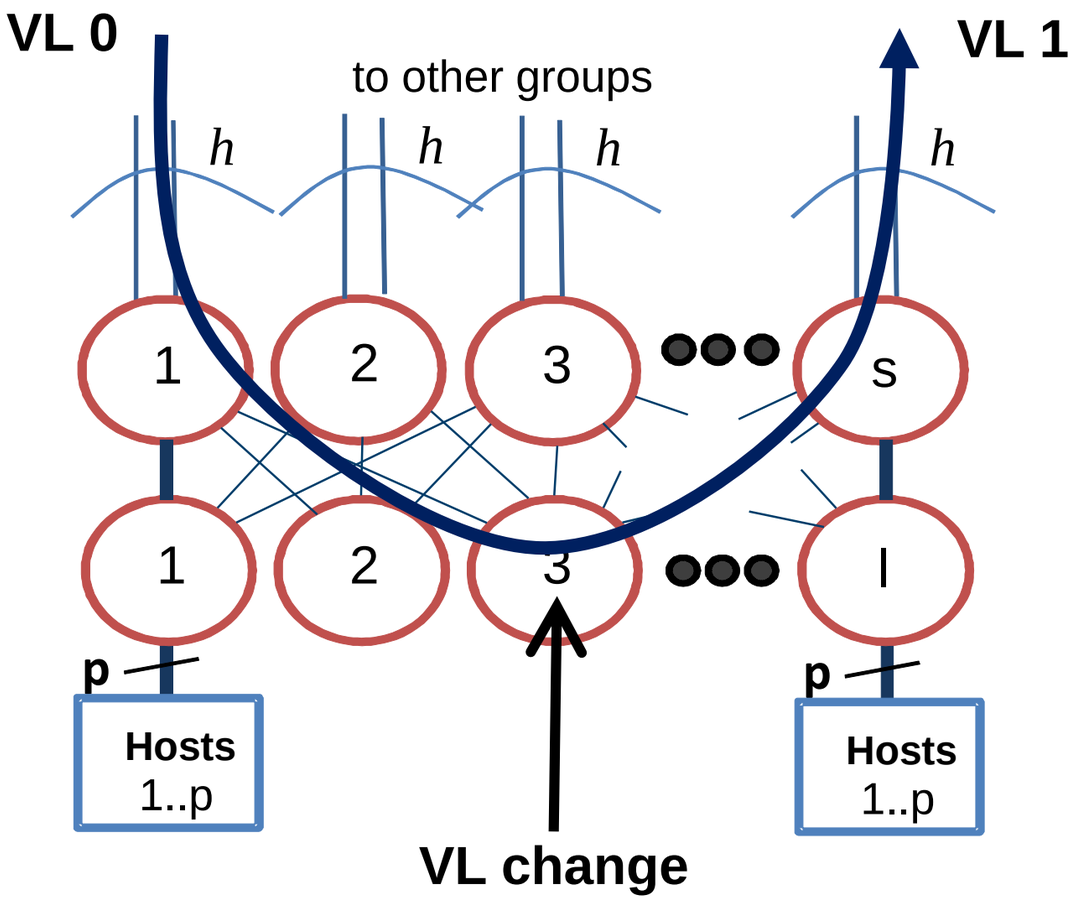
- In Dragonfly credit loop may happen.
- Credit loop lock is prevented when using
  3 VLs [Kim et al., 2008]
  4 VLs [Prisacari et al., 2014] .
- Similarly, in Dragonfly+ credit loop may happen.
  - Solved by **2 VLs**.



Note: inter-group links are shown partially and connected randomly

Credit Loop Deadlock in Dragonfly+

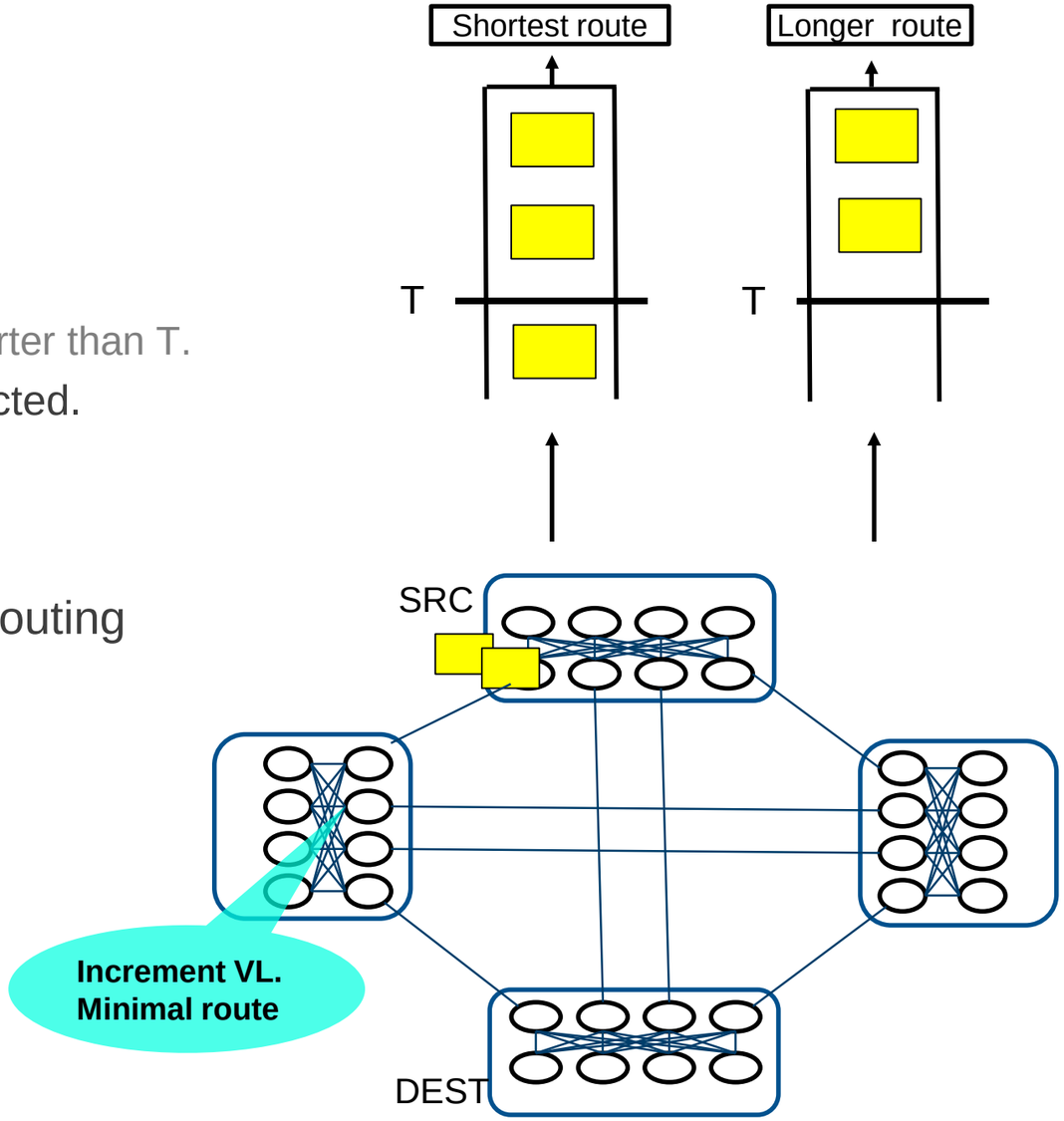
- Solution: Packet forwarding from “down” to “up” increments VL.
  - 2 VLs are enough for preventing deadlock



Fully Progressive Adaptive Routing (FPAR)

- Routing decisions are evaluated in every router:
  - Assuming predefined queue length threshold T
  - Routing rules:
    - Lower priority route is chosen if:
      - all the egress queues on higher priority routes are longer than T,
      - and there is an egress queue on the lower priority router that is shorter than T.
    - Otherwise, higher priority route with shortest egress queue is selected.

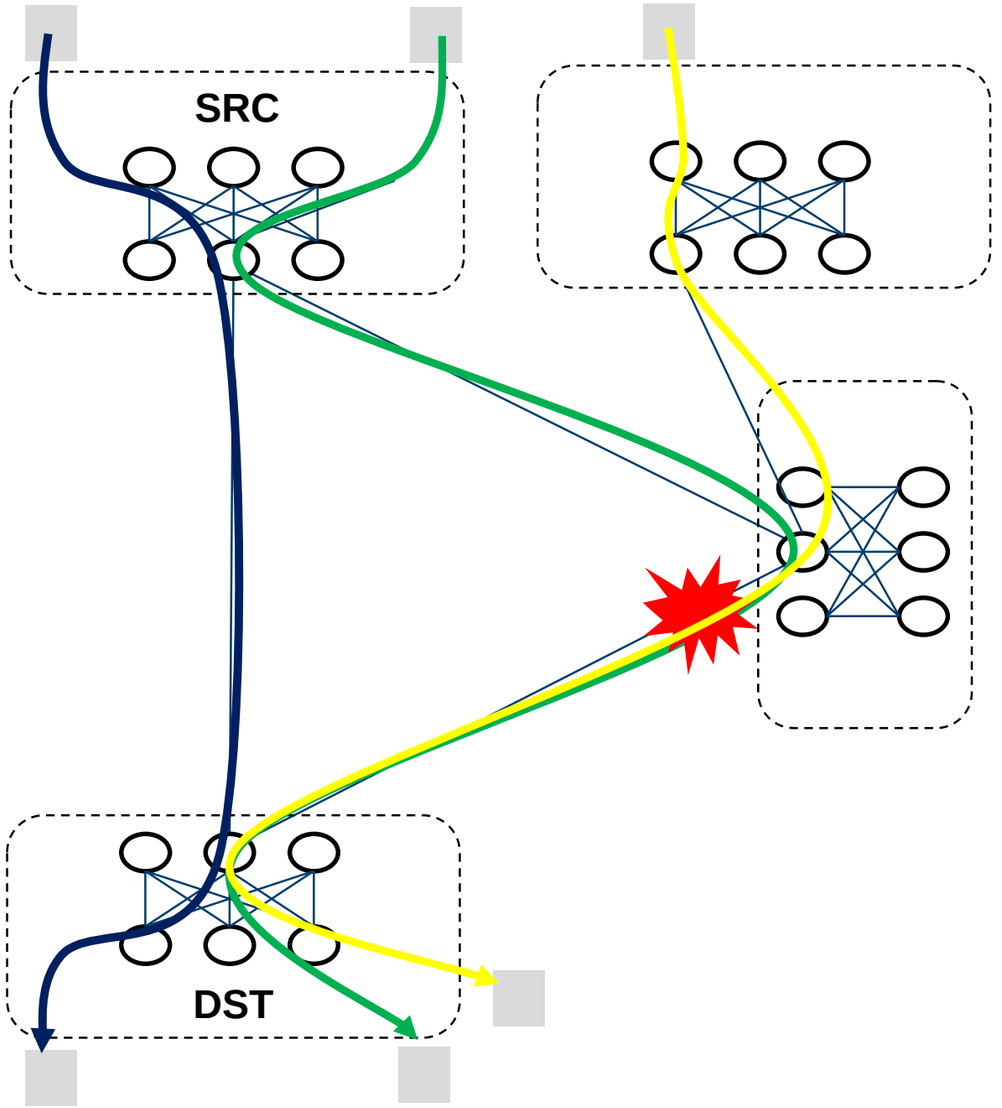
- Avoiding routing live-lock
  - Distinguish between the flows that are restricted to a minimal routing and the flows that are allowed for free balancing.
  - FPAR rules are based on packet's VL:
    - packet with incremented VL must be forwarded on min-hop route.



Adaptive Routing Notifications for Remote Congestion (ARN)

- *Remote congestion* - congestion that shall be resolved in a router earlier on the route than the congested.

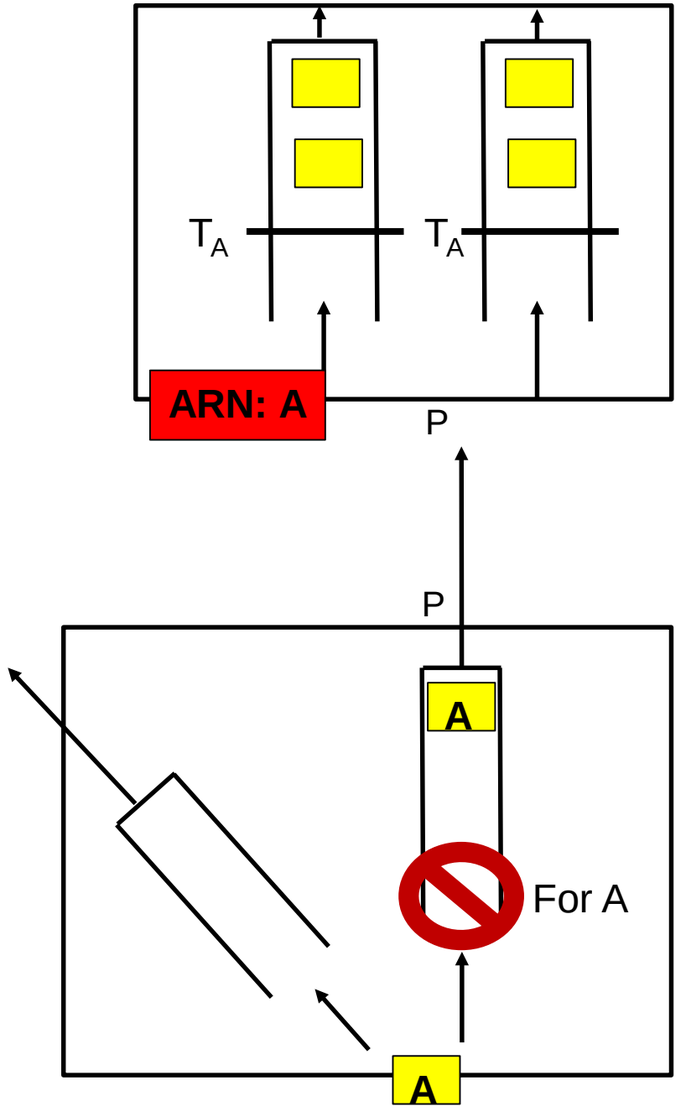
- Solutions: rely on passing information about congestion to the router that can resolve it.
  - Global knowledge
    - Requires centralized entity.
  - Accelerating congestion spreading [Kim et al., 08] to pass the congestion information faster.
    - However, causes victim flows.
  - Piggyback global link state over existing network packets [Jiang et al., 2009]
    - Might unnecessarily use bandwidth and relies on existence on other packets.



Adaptive Routing Notifications for Remote Congestion (ARN)

- Adaptive Routing Notification (ARN) messages
- Advantages:
  - Do not depend on existence of other messages for piggybacking,
  - Sent towards the resolving routers only, hence does not waste bandwidth on other links.

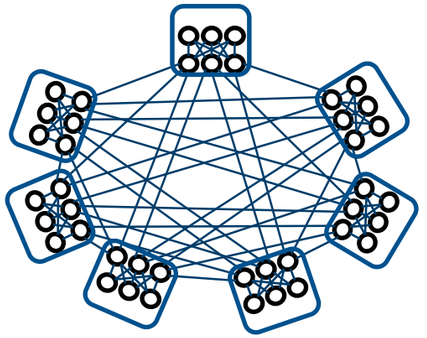
- Details in the paper….

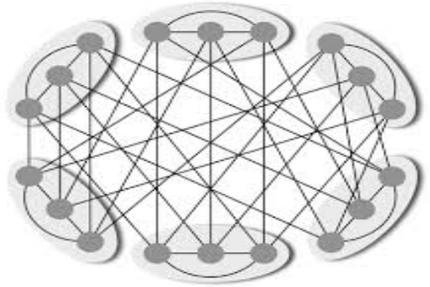


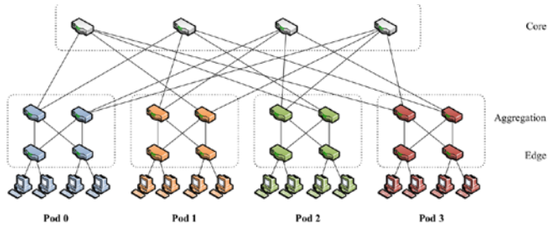
Analytical Analysis

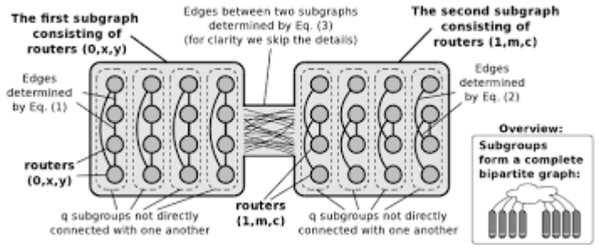
- Topologies to compare:
  - Dragonfly+
  - Dragonfly
  - Non-blocking 3-level Fat Tree.
  - 2:1 blocking 3-level Fat Tree.
  - SlimFly (in the paper…)

- Hypercube and Torus target different criteria
  - less expensive, but compromise on bi-sectional bandwidth







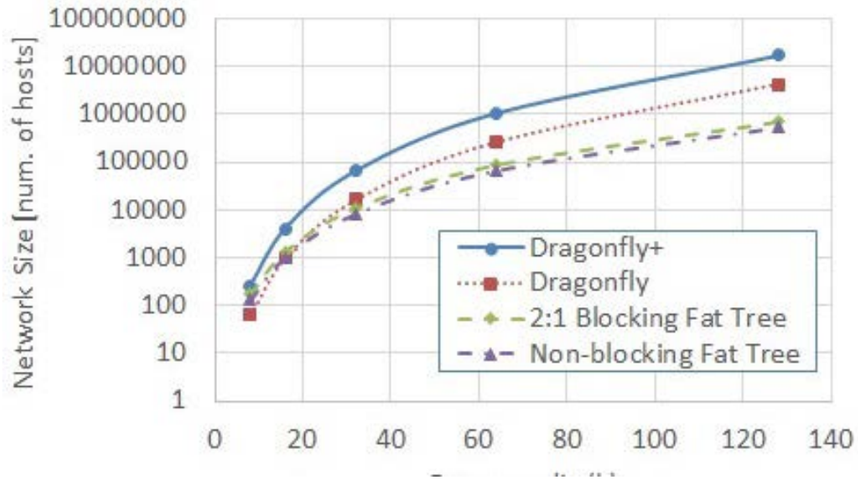


Scalability: Maximal Number of Hosts

Topology	Expression	For router radix k=36
Dragonfly+	$N_{dfp} = {k^4}/{16} + {k^2}/{4}$	105,300
Dragonfly	$N_{df} = {k^4}/{64} + {k^2}/{8}$	26,406
2:1 blocking Fat Tree	$N_{ft,b} = {k^3}/{3}$	15,552
Non-blocking Fat Tree	$N_{ft,nb} = {k^3}/{4}$	11,664

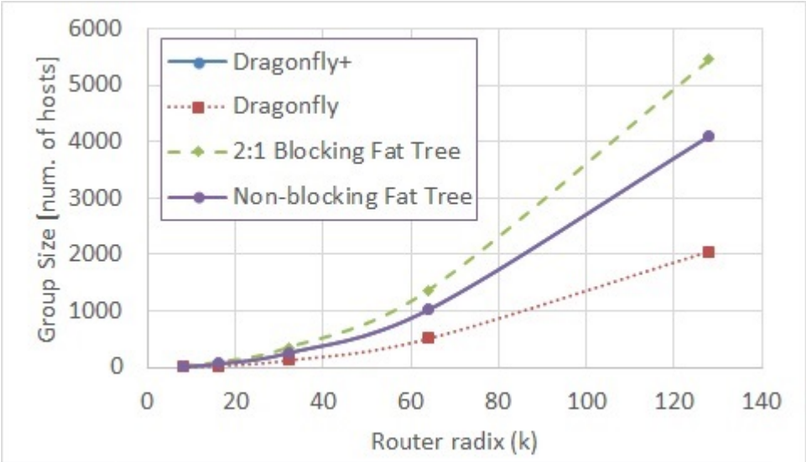
- Conclusion:
  - Dragonfly+ shows the best scalability compared to the other topologies.
  - For any router radix:

$$N_{dfp} \approx 4N_{df}$$



Locality: Group Size

- Larger group size → larger amount of traffic is group-internal.
  - Does not use inter-group global links.
- Dragonfly, Dragonfly+ and Fat Tree groups have full bi-sectional bandwidth.
  - Higher throughput for arbitrary traffic patterns.



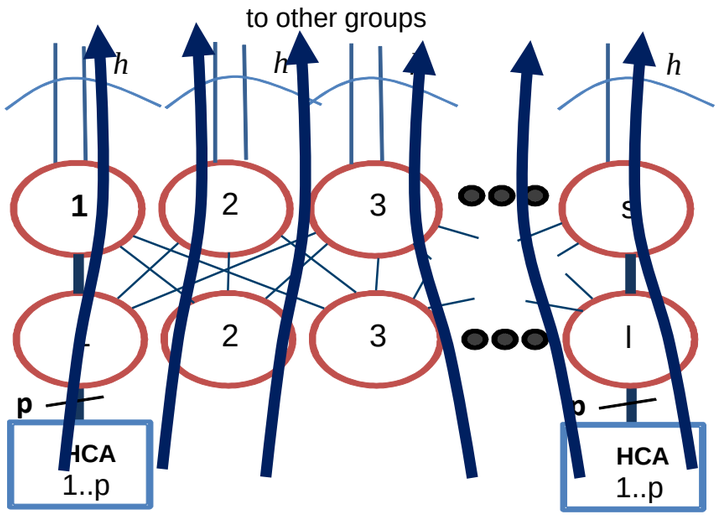
Topology	Expression	For router radix k=36
Dragonfly+	$G_{dfp} = {k^2}/{4}$	324
Dragonfly	$G_{df} = {k^2}/{8}$	162
2:1 blocking Fat Tree (2nd level pod)	$G_{ft,b} = {k^2}/{3}$	432
Non-blocking Fat Tree (2nd level pod)	$G_{ft,nb} = {k^2}/{4}$	324

- Conclusion: Dragonfly+'s group is larger than Dragonfly's

Network Throughput

- Measures of network throughput:
  - Uniform random traffic
  - Worst case permutation traffic.

- Uniform random traffic
- Dragonfly+, Dragonfly:
  - Following suggested rules-of-thumb: number of global link ports = number of hosts.
  - Only single global link is used per flow.
  - Global links serve traffic without causing bottleneck. And, thus, provide up to 100% network throughput.



$$p = l = s = h$$

Network Throughput

- Worst Case Permutation Traffic (defined by [Prisacari et al., 2014])

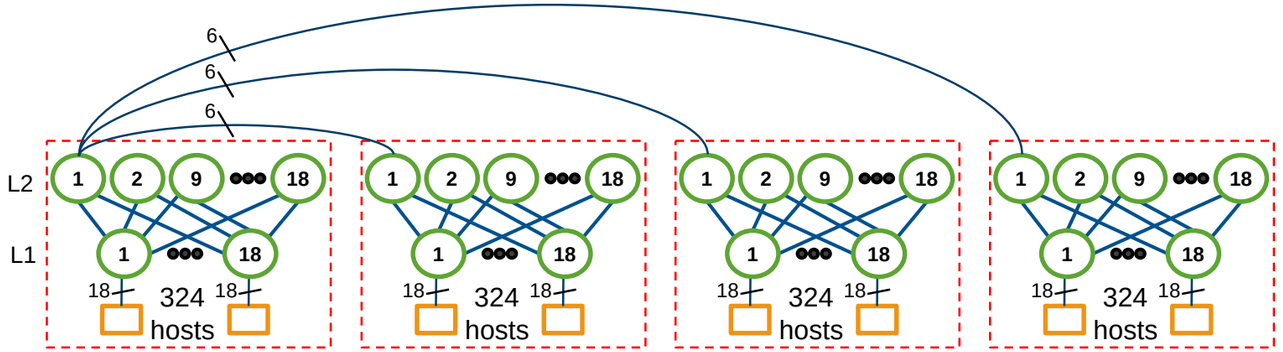
- Dragonfly
  - Bandwidth utilization is bounded by $\frac{3}{4p} = \frac{3}{k}$, For radix k = 36: **8.33%**.
  - [Prisacari et al., 2014] increase the bandwidth utilization to **42%**
    - by adding additional longer routes and additional VL.

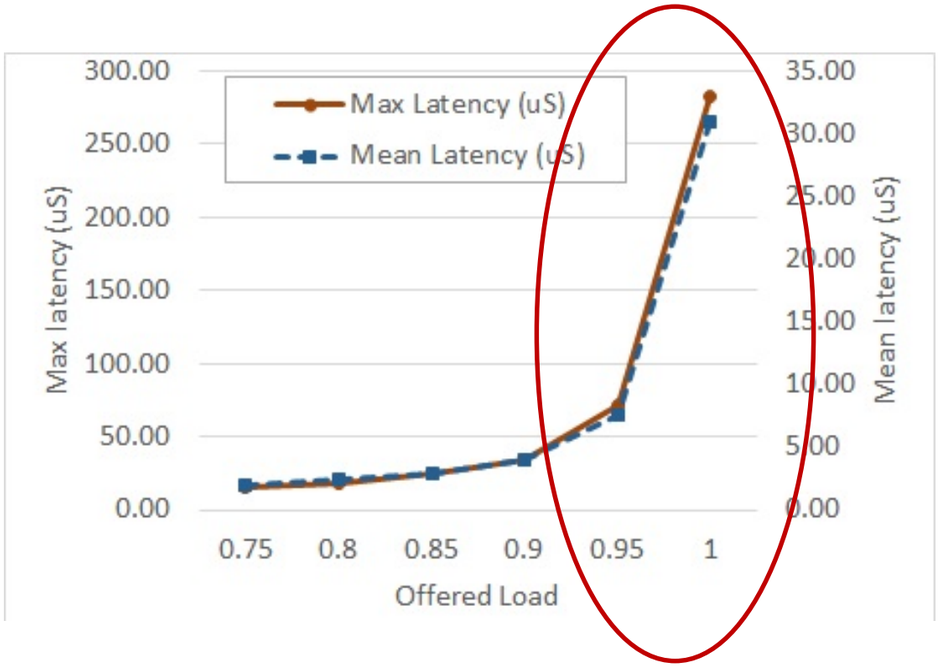
- Dragonfly+
  - **50%** network bandwidth utilization for worst case permutation, for any radix.
  - Details in the paper….

Simulations: Uniform Traffic

- Omnet++ based simulator
- Topology:
  - Dragonfly+ network
  - 1296 hosts
  - 4 groups.
  - 36-radix routers
- Traffic: uniform random destination

→ close to 100% network throughput





Simulations: Permutation Traffic

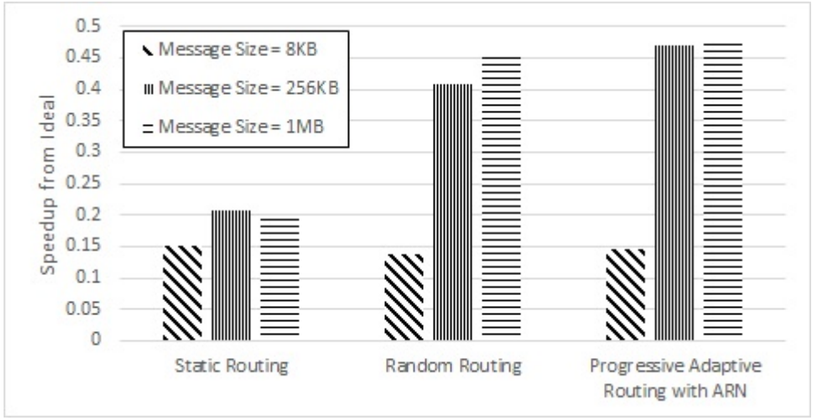
- Topology:
  - Largest-size network of K = 8 radix routers
  - 272 hosts

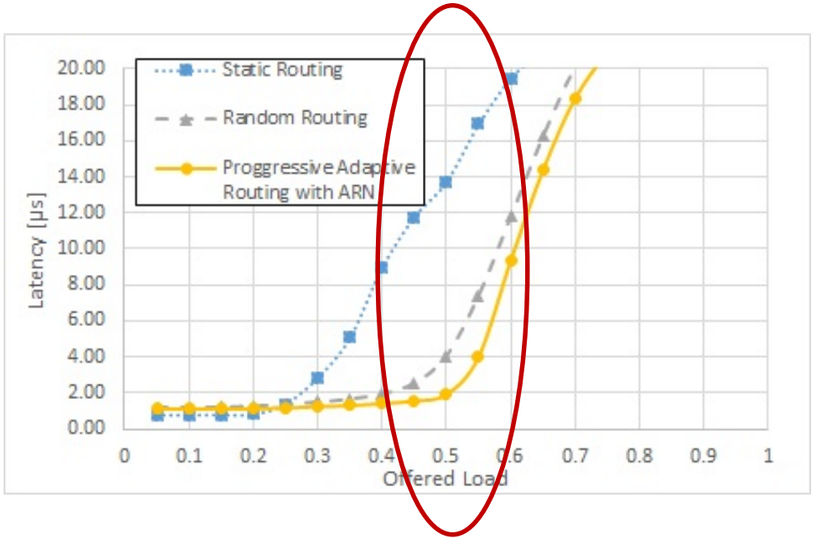
- Traffic:
  - 100 randomized permutations
  - Selected worst-performance single permutation
  - Varied message size: 8KB, 256KB, 1MB
  - Permutation completion time is reported upon receiving the last acknowledgement on message arrival.

→ FPAR achieves almost 50% network utilization

- static routing achieves only 20%.
  - Note: results for small messages permutation are indifferent to the routing scheme, since the traffic does not challenge the router buffers.

→ Bound of 50% network utilization for adaptive routing





Summary

- Dragonfly+ topology combines the benefits of Dragonfly and Fat Tree.
- Fully Progressive Adapting Routing technique with Adaptive Routing Notification messages.

	Dragonfly+	Dragonfly
Scalability	4N	N
Group size	2G	G
Worst case permutation throughput	~50%	~42%
Number of VLs	2	4
Cost: hosts per router	R	R
Uniform random traffic throughput	~100%	~100%
Diameter	3	3
Maximal route length	6	6

Thank You

Mellanox TECHNOLOGIES

Connect. Accelerate. Outperform.™

Backup

Dragonfly+ Advantages

- More scalable: allows connecting larger number of hosts to the network.
- Better worst case network throughput.
- Less Virtual Lanes (VLs) to prevent credit loop deadlock.
- Larger groups.

- Vs. Dragonfly with 2D-flattened butterfly groups
  - Lower diameter (avg/max)
  - Lower cost (hosts per router)
    - assuming full network utilization for intra-group traffic

Cost: Number of Hosts per Router

Topology	Expression	For k=36
Dragonfly+	$R_{dfp} = \frac{4}{k} N_{dfp}$	9
Dragonfly	$R_{df} = \frac{4}{k} N_{df}$	9
2:1 blocking Fat Tree	$R_{ft,b} = \frac{4}{k} N_{ft,b}$	9
Non-blocking Fat Tree	$R_{ft,nb} = \frac{5}{k} N_{ft,nb}$	7.2

- Conclusion:
  - Cost per host is equal between Dragonfly and Dragonfly+, while both are less expensive than non-blocking Fat Tree.

Diameter and Maximal Assured Route Length

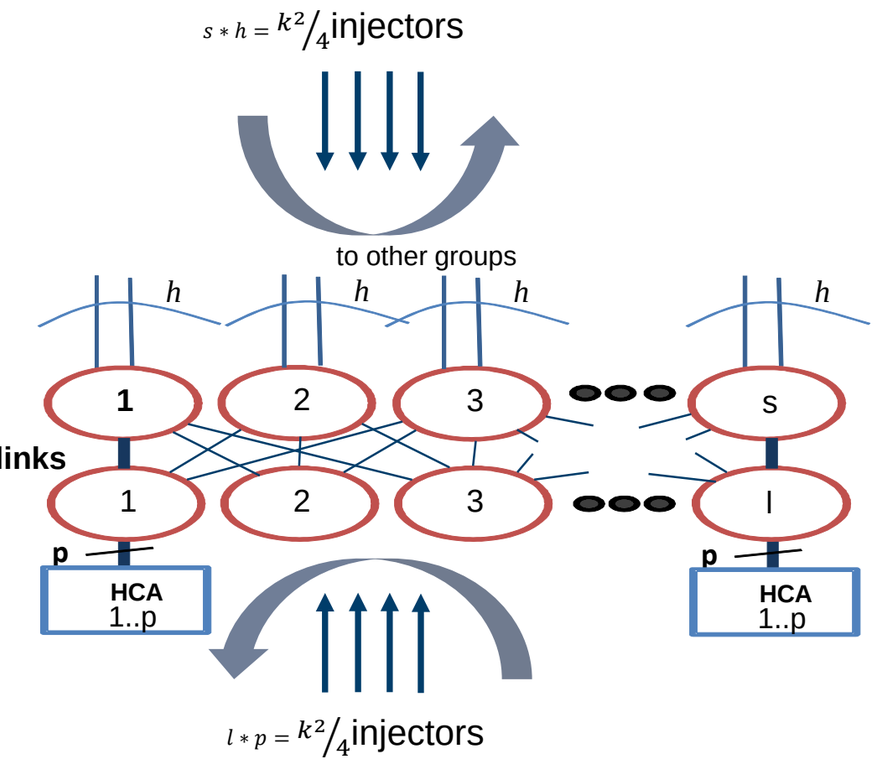
- Two measures:
  - Minimal hop route (diameter)
  - Maximal hop route (defined by adaptive routing)

Topology	Minimal Route Length	Maximal Route Length
Dragonfly+	3	6
Dragonfly	3	5 / 6
Fat Tree	4	4

Network Throughput

- Dragonfly+
  - *Non-minimal-hop routes* use *two* inter-group links, hence the inter-group network utilization is bounded by **50**%.
  - Dragonfly+ group consists of:
    - $k^2/4$ "up" links and $k^2/4$ "down" links between spine and leaf routers.
    - $k^2/4$ hosts and $k^2/4$ global links injecting traffic to the group, each one requires a single "up" link and a single "down" link in the worst case.
    - Thus, a total $k^2/2$ "down" links and $k^2/2$ "up" links are required.
      - twice the amount than the topology provides.
      - hence is sufficient for load of **50%** of link rate.

$p = l = s = h = k/2$



$k^2/4$ **"down" links and** $k^2/4$ **"up" links**

Comparison to SlimFly

- SlimFly [Besta et al., 2014]:
  - lowest network diameter compared to alternative topologies.
  - up to 45% of network utilization with non-minimal adaptive routing.

- Drawbacks compared to Dragonfly+:
  - Scalability: router radix defines the exact network size.
    - router radix k = 36: 6,144 hosts and 512 routers (by choosing q = 16; p = 12).
  - Credit loop deadlock avoidance requires 4 VLs
    - Larger than Fat tree(1), Dragonfly+(2) and Dragonfly(3).
  - Cumbersome structure
    - hence it cannot be based on common building blocks.