A Cost and Scalability Comparison of the Dragonfly versus the Fat Tree

Frank Olaf Sem-Jacobsen
frankose@simula.no
Simula Research Laboratory

HPC Advisory Council Workshop
Barcelona, Spain, September 12, 2013

ACKNOWLEDGEMENTS

- Sven-Arne Reinemo
- Tor Skeie
- Mathias Myrland
- Eitan Zahavi, Mellanox
- Gilad Shainer, Mellanox

AGENDA

- Motivation
- Method
- The fat-tree topology
- The dragonfly topology
- Blocking, cost, and scalability
- Conclusion

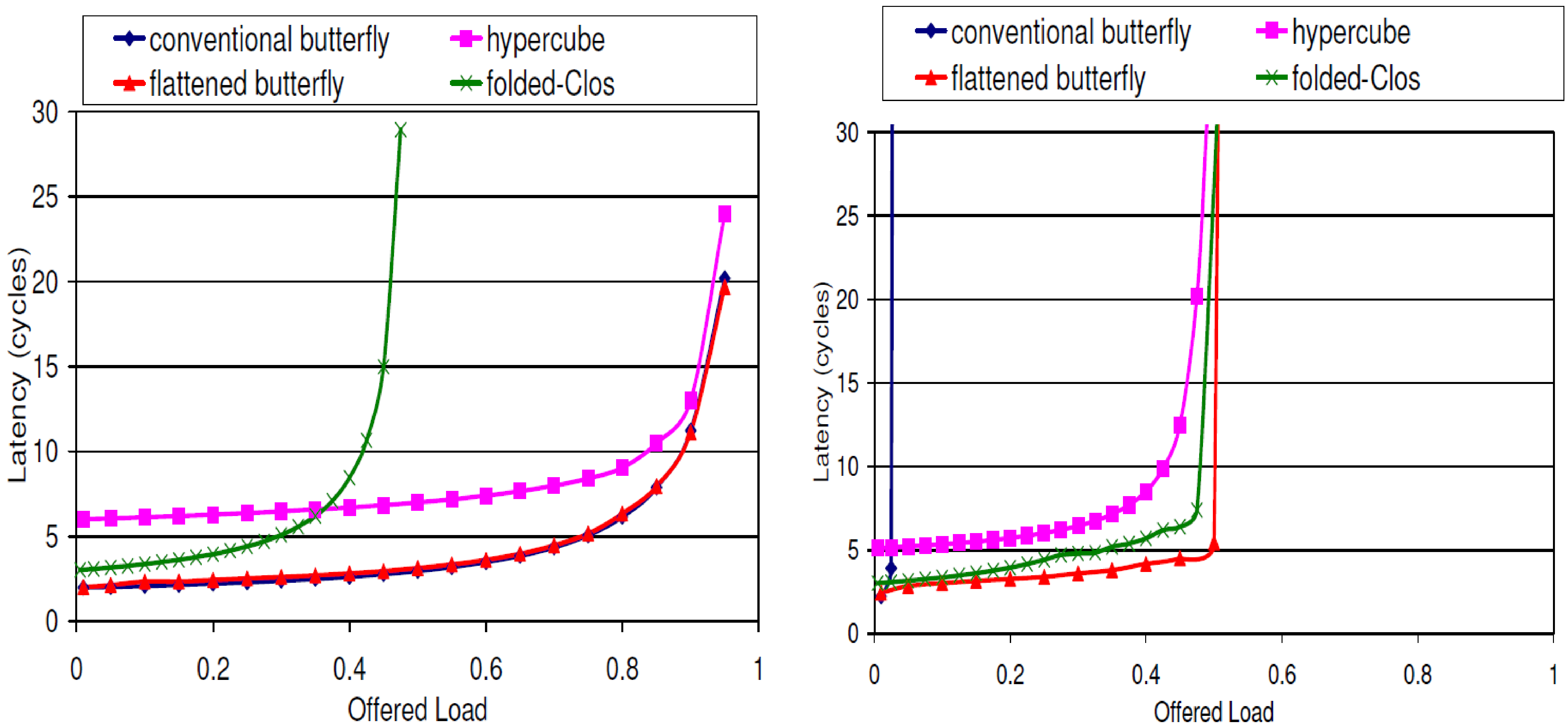
MOTIVATION

The fat-tree is the dominating topology for InfiniBand networks, but the proposed dragonfly topology has been suggested as an alternative. In that context we would like to answer the following questions:

- Is the dragonfly a viable alternative to the fat-tree?
- How do they compare in the fundamental properties blocking, cost and scalability?
- What about traffic patterns?
- A comparison on equal terms with regards to CBB.
- When should you choose which?

Why the dragonfly

- The range of traffic patterns considered is important
- Uniform/random versus well-defined/shift/MPI collectives
- Where is the crossing point for costs/performance for the dragonfly and fat tree?



(a) Uniform

(b) Worst case

J. Kim, W. Dally, and D. Abts. Flattened buttery: a cost-ecient topology for high-radix networks. In *Proceedings of the 34th annual international symposium on Computer architecture*, pages 126-137. ACM, 2007.

APPROACH

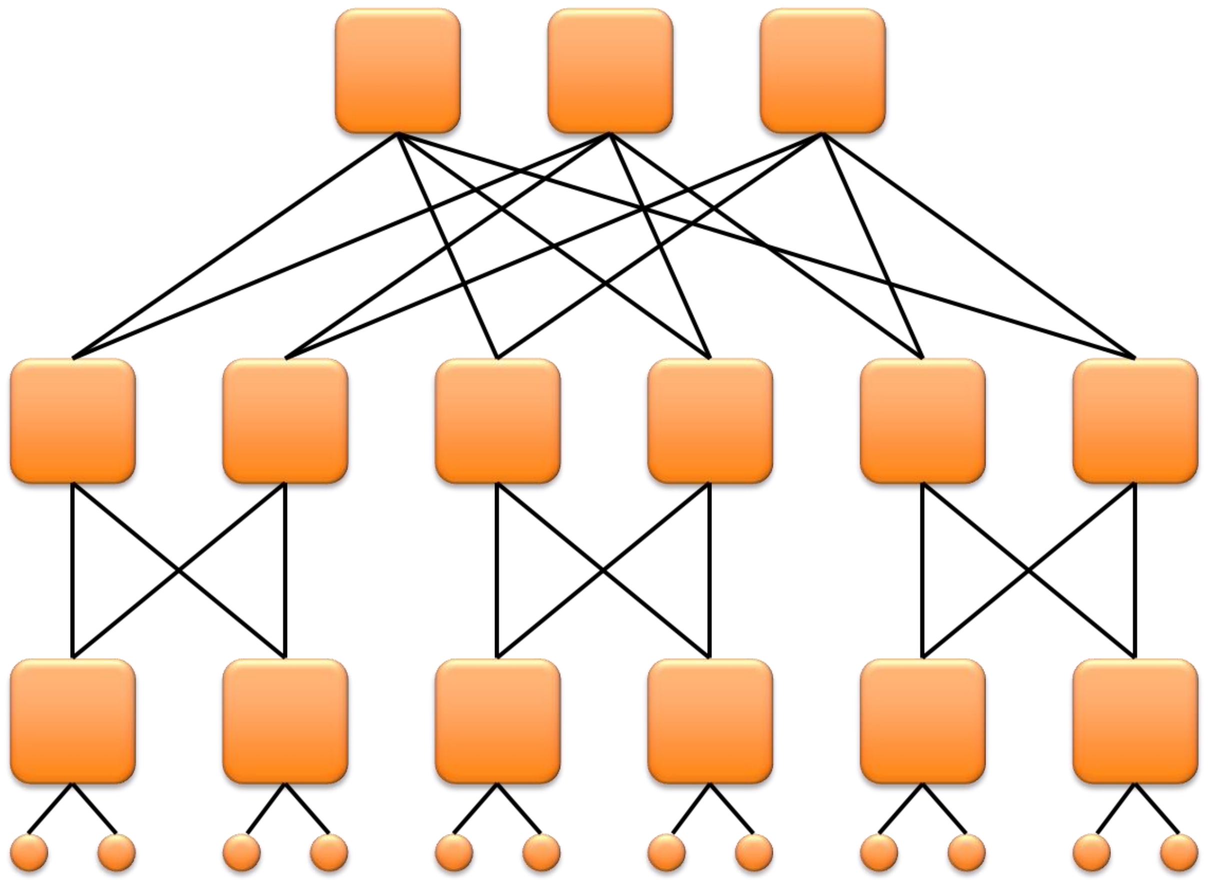
We aim to establish the lower and upper performance bound for the (dragonfly) topology independent of any routing algorithm.

The study consists of three parts:

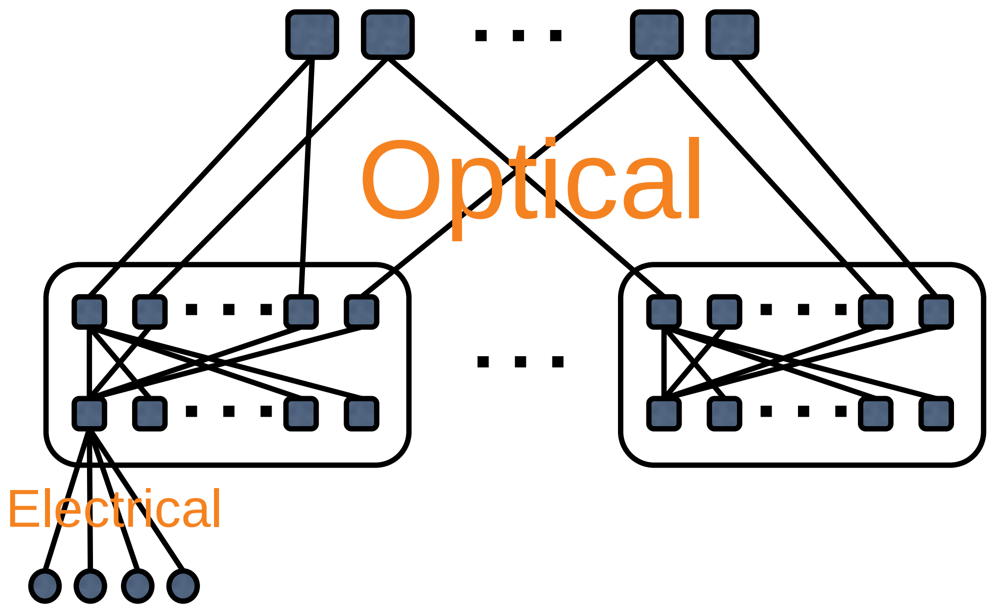
- Worst case analysis for the CBB ratio of the dragonfly and the fat-tree, giving a lower bound on performance.
- Permutation traffic analysis using linear programming, giving an upper bound on performance.
- Cost-scalability analysis by generating all possible topology sizes and apply a cost model to evaluate cost-efficiency.

THE FAT-TREE

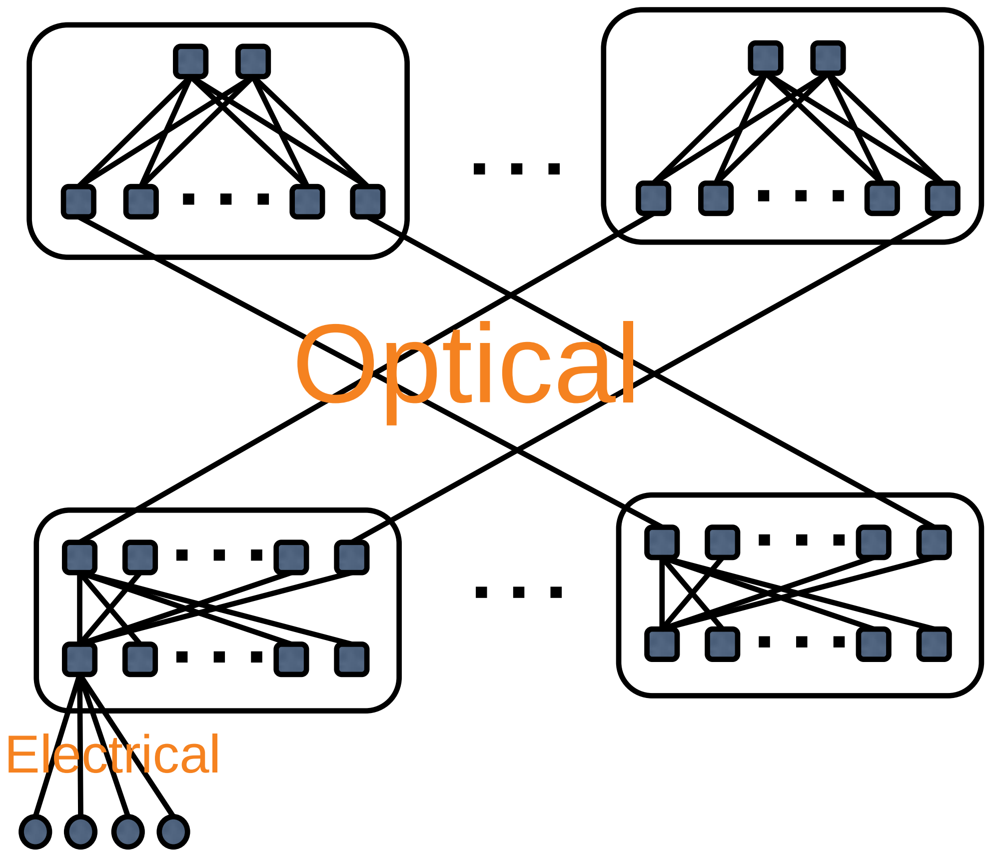
- Most common topology for InfiniBand based computers
- Can be routed deadlock free without additional resources such as virtual lanes
- Fault-tolerant through its path diversity
- Full bisection bandwidth for arbitrary permutations
- Scalable, also with respect to cost
- Performance suffers due to static routing, but adaptivity is supported



THE FAT-TREE IMPLEMENTED



3 tiers



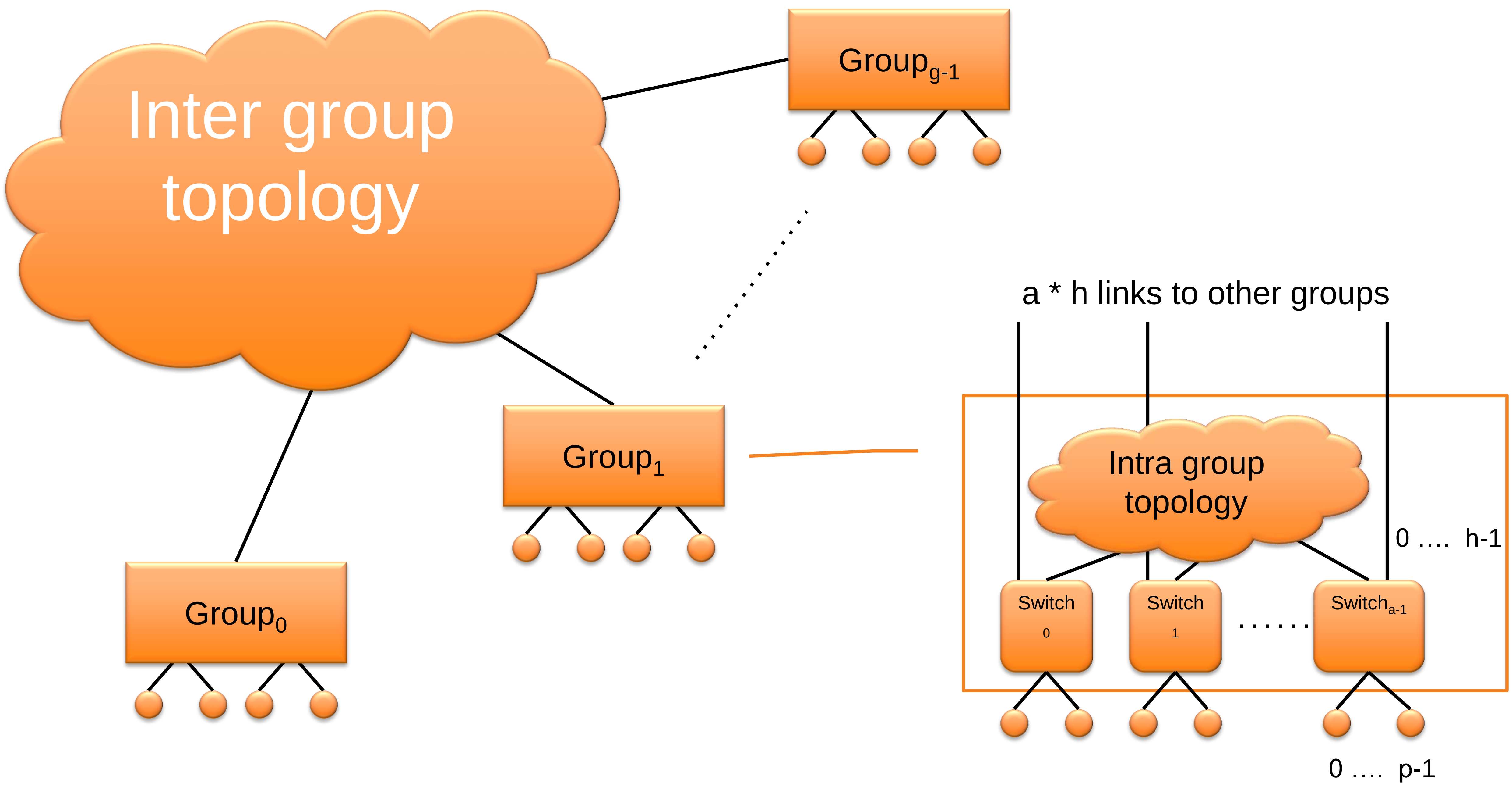
4 tiers

THE DRAGONFLY

- Recently proposed by John Kim et al. in [1].
- Caused discussion in the HPC community about is suitability for IB and as an exascale topology.
- The dragonfly is a hierarchical topology with the following properties:
  - Several groups are connected together using all to all links, i.e. each group has at least one link directly to each other group.
  - The topology inside each group can be any topology. The recommendation in [1] is the flattened butterfly.
  - Focus on reducing the number of long links and network diameter.
  - Requires non-minimal global adaptive routing and advanced congestion look ahead for efficient operation.
  - CBB ratio = 2 for its standard implementation

[1] John Kim et al. "*Technology-Driven, Highly-Scalable Dragonfly Topology*" in proceedings of the 35th
International Symposium on Computer Architecture, 2008.

THE DRAGONFLY

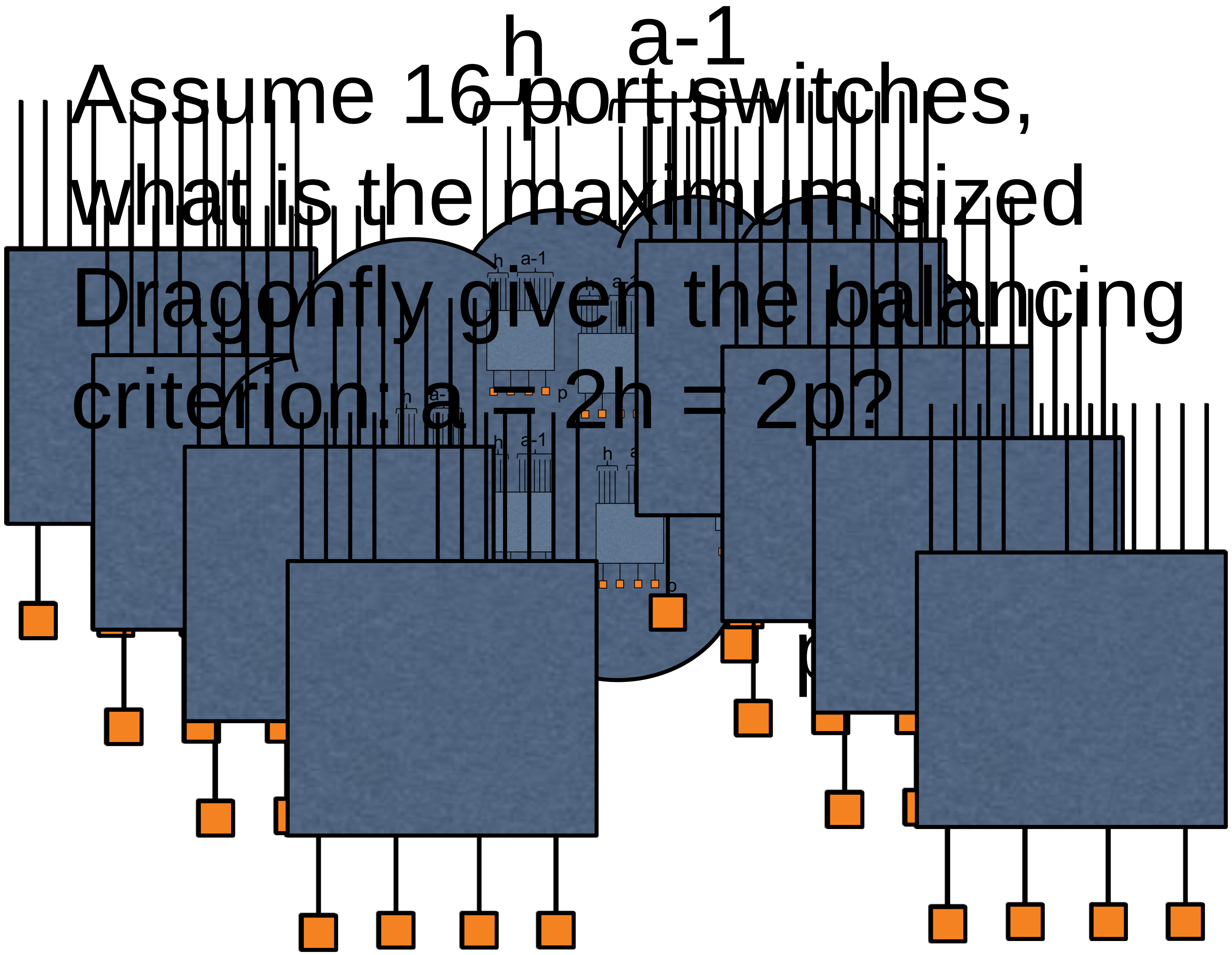


The recommendation in [1] is to keep: $a \geq 2p \geq 2h$

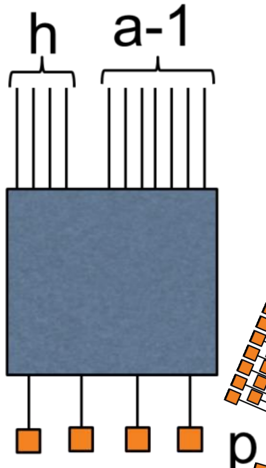
THE DRAGONFLY

Assume 16 port switches, what is the maximum sized Dragonfly given the balancing criterion: a = 2h = 2p?

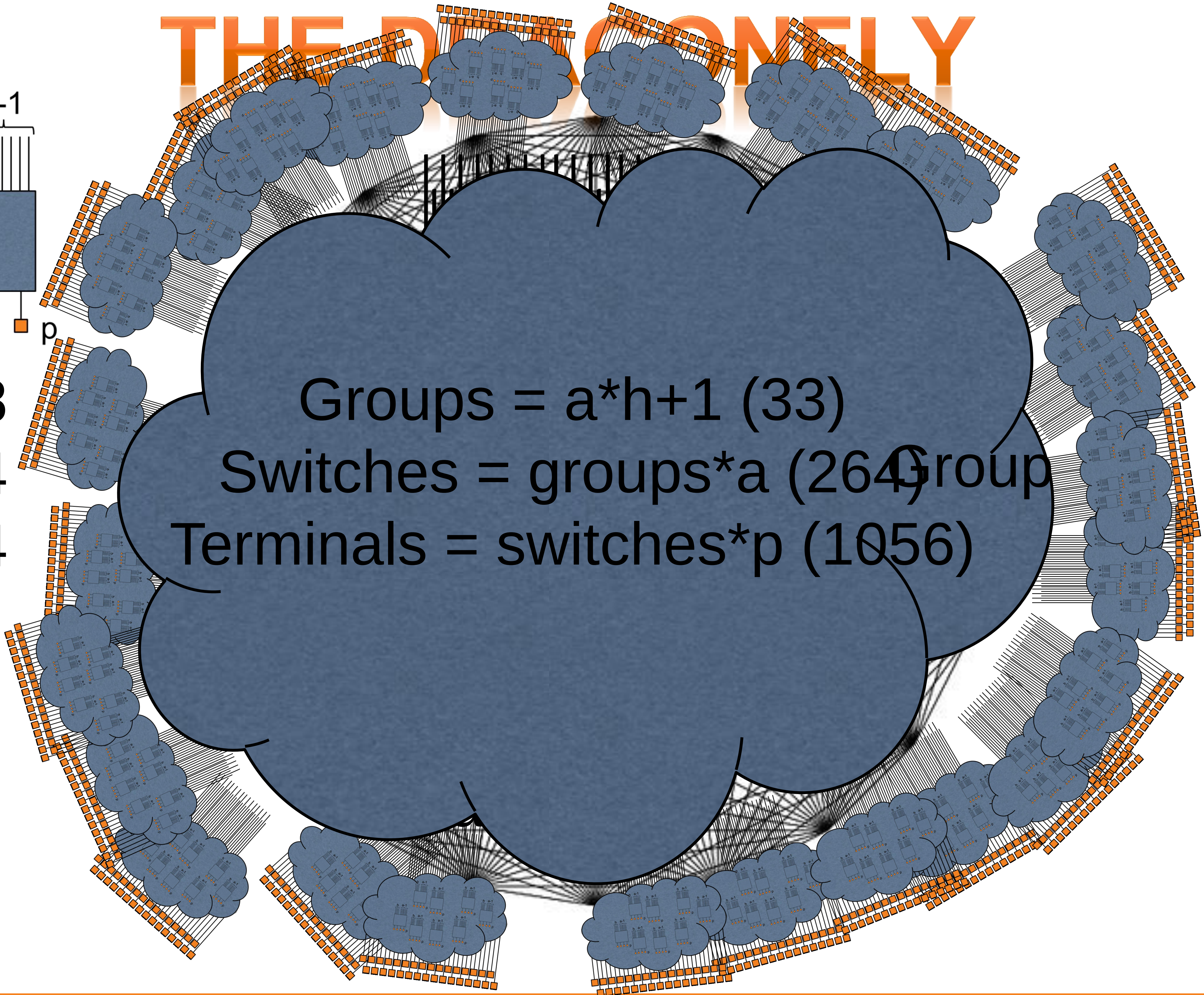
a=8
p=4
h=4
a=2h=2p



THE DRAGONFLY

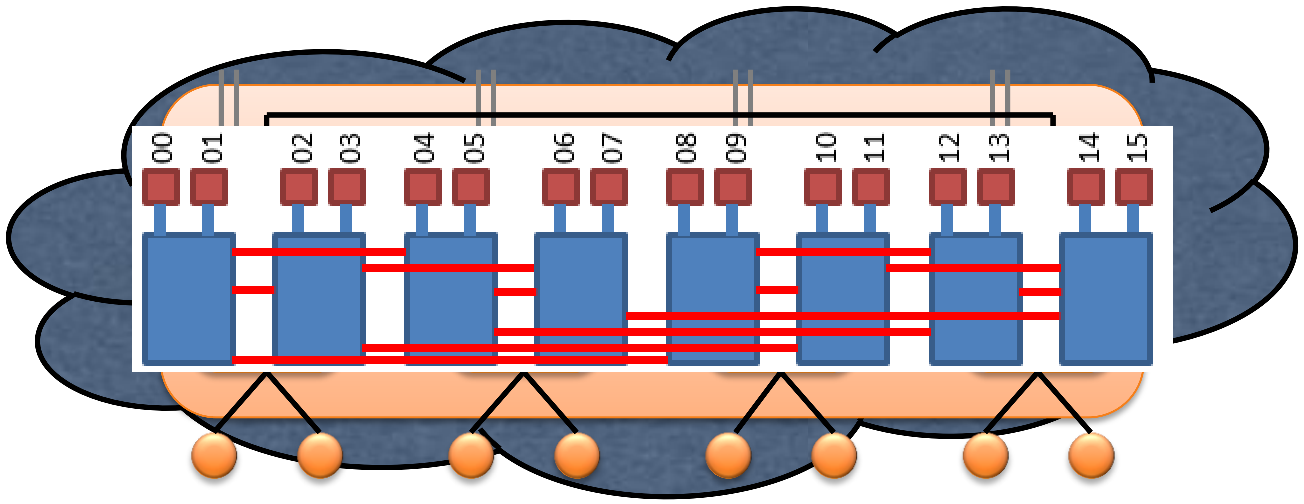


a=8
p=4
h=4



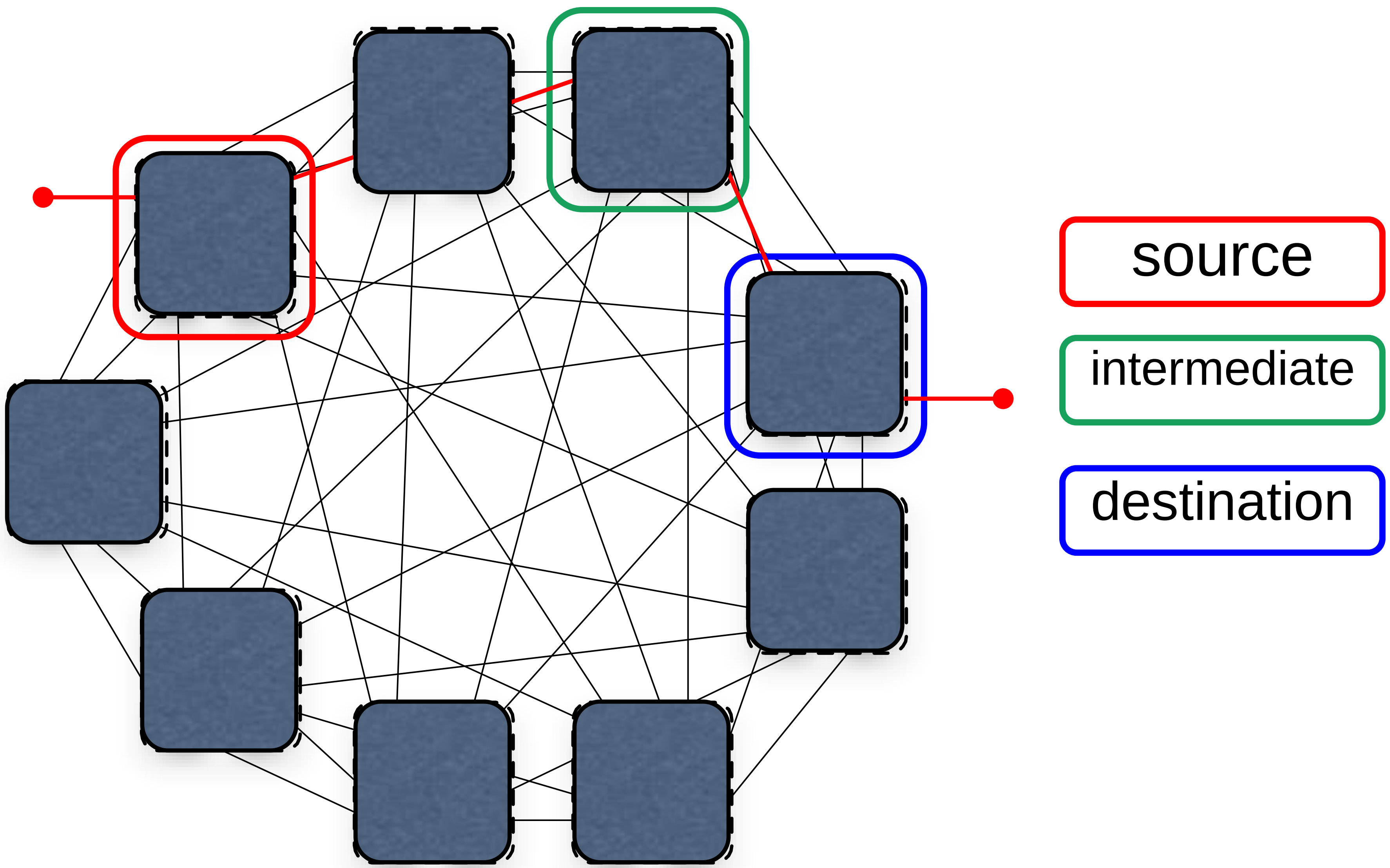
Groups = a*h+1 (33)
Switches = groups*a (264)
Terminals = switches*p (1056)

FBFLY GROUP



- A single flattened butterfly group with 8 terminals and 8 external connections.
- Fully connected, but requires also non-minimal adaptive routing for path diversity and load balancing.
- 2D flattened butterfly requires more internal routing.

NON-MINIMAL HOPS



Dragonfly with a=4, p=2, and h=2. Can make non-minimal hops in the source (s), intermediate (i), and destination (d) group.

CBB EQUATIONS

- Global CBB ratio

$$CBB_{global} = \frac{2ap}{ah + 1}$$

- Local CBB ratio

$$CBB_{local} = 2 * \frac{c(s) * p + u * h \;* \min(CBB_{global}, 1.0)}{\sqrt[n']{a}}$$

$$u = \min(1.0, c(i) + c(d))$$

CBB EQUATIONS

The worst case occurs when the probability for making a non-minimal hop is one in all three groups : $c(s) = 1, c(i) = 1, c(d) = 1$

$$CBB_{local} = 2 * \frac{1 * p + 1 * h \ * \min(CBB_{global}, 1.0)}{\sqrt[n']{a}}$$

For a standard dragonfly

$$CBB_{global} = \frac{2ap}{ah+1} \sim 2(1.98)$$

$$CBB_{local} = 2 * \frac{1 * p + 1 * h \ * 1}{a} = \frac{2\,(p+h)}{a} = 2$$

CBB EQUATIONS

Using an LP solver to optimally place the paths for 10 000 permutations yields

$$c(s) = 0.74,\ c(i) = 0.15,\ c(d) = 0.74$$

for random permutations and

$$c(s) = 0.56,\ c(i) = 0.48,\ c(d) = 0.58$$

for group external permutations, i.e. all destinations are are outside the group.

CBB EQUATIONS

Example for the uniform traffic case:

c(s) =0.74, c(i) =0.15, c(d) = 0.74

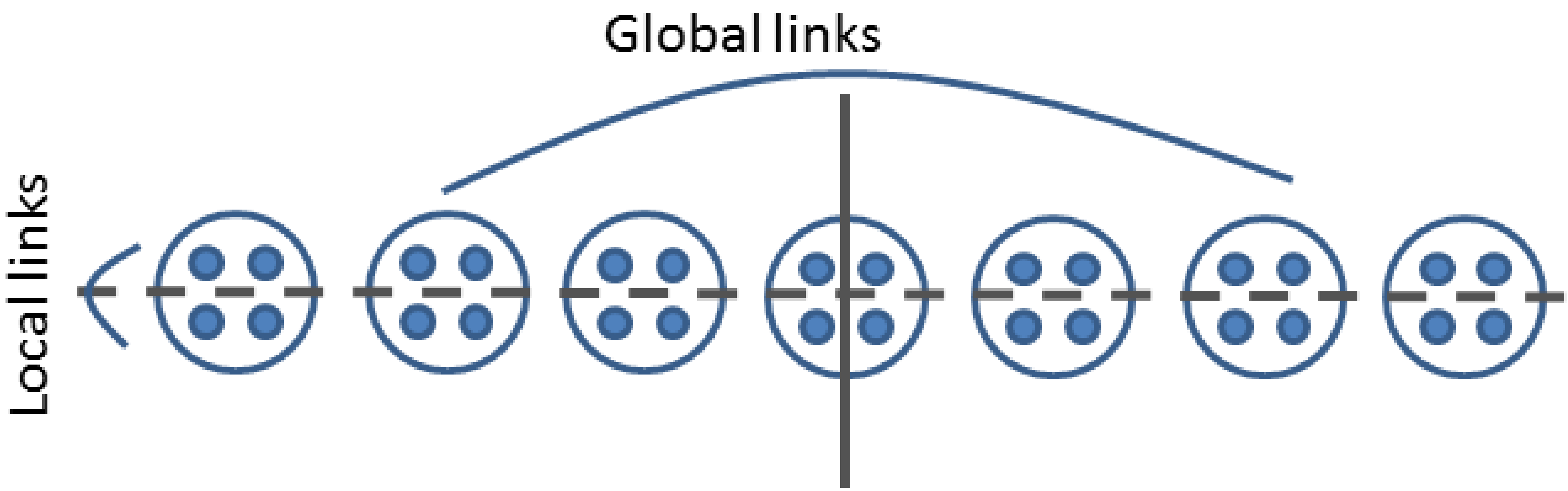
For a standard dragonfly

$$CBB_{global} = \frac{2ap}{ah+1} \sim 2(1.98)$$

$$CBB_{local} = 2 * \frac{0.74 * p + 0.89 * h \ * 1}{\sqrt[n']{a}}$$

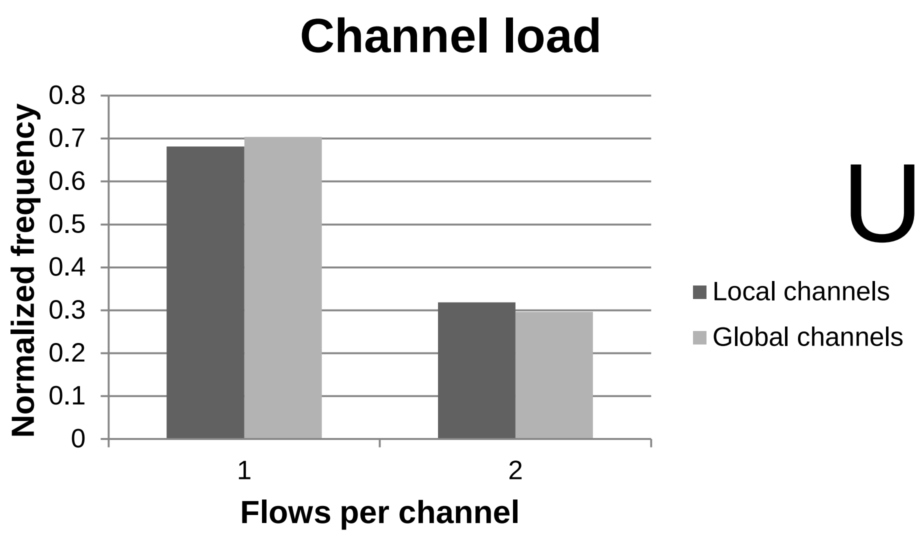
$$= \frac{2\,(0.74p + 0.89h)}{a} = 1.63$$

OVERALL CBB RATIO

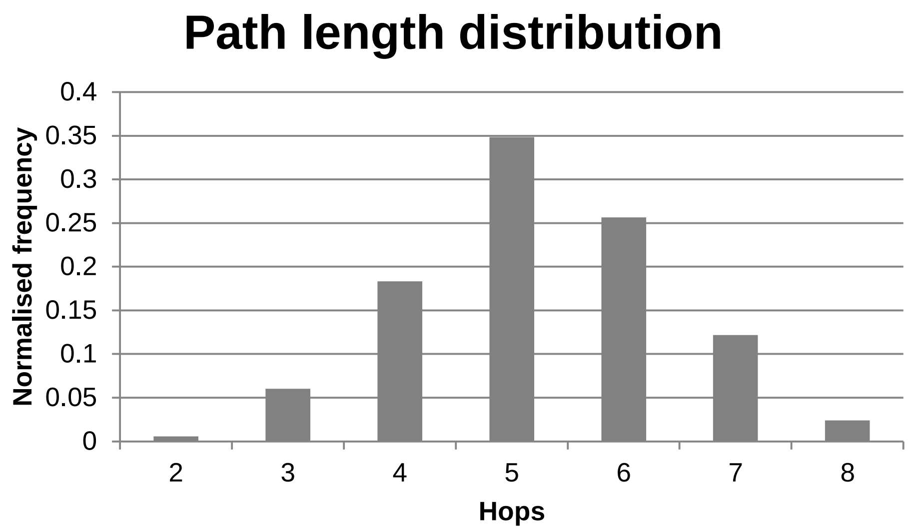


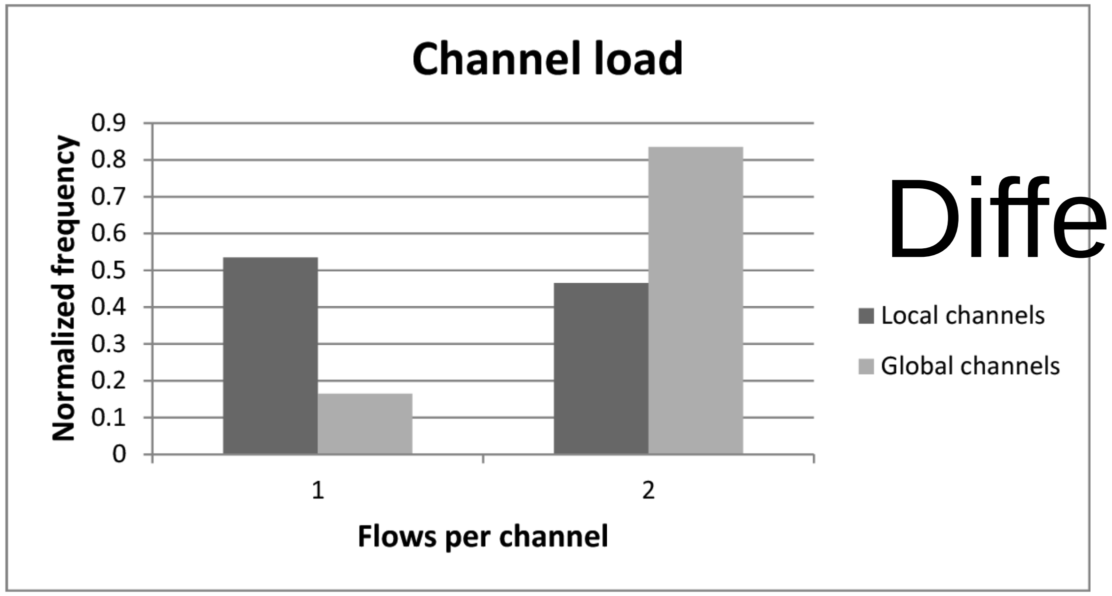
$$CBB = \max(CBB_{local}, CBB_{global})$$

CHANNEL USAGE

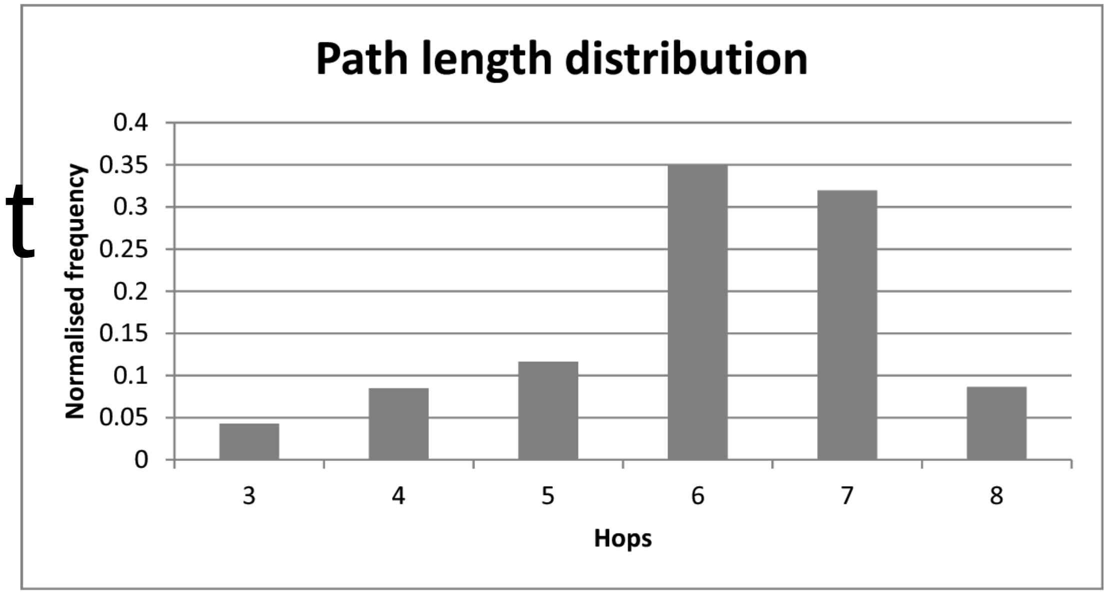


Uniform



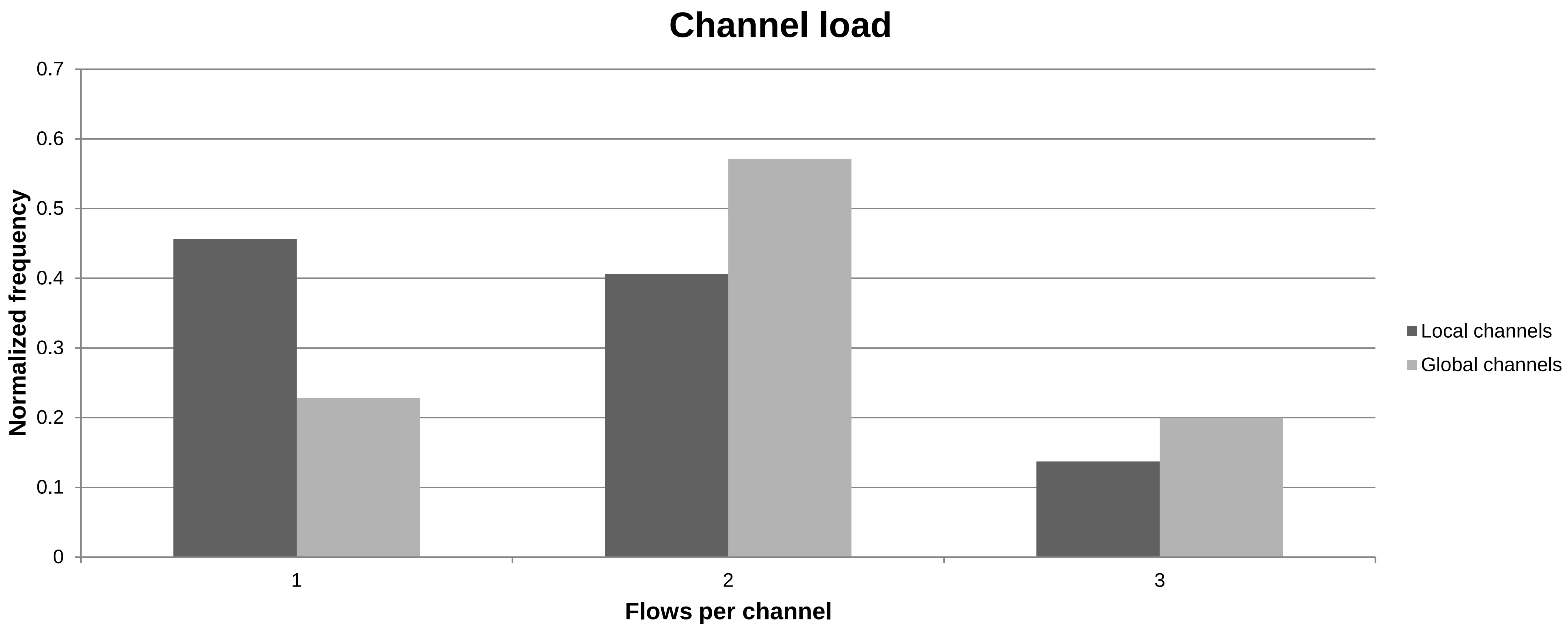


Different



Statistics for 10 000 uniform (above) and random different group (below) permutations on a dragonfly with flattened butterfly groups.

CHANNEL USAGE WITH MINIMAL LOCAL ROUTING



Minimal local routing leads to increased maximum channel load on the local channels (the increased global channel load is a function of the LP constraint)

SCALABILITY AND COST

We compare the following four topologies:

- Two dragonflies with 1-d and 2-d flattened butterfly groups, respectively
- A 3- and 4-level fat-tree.
- We looked at how the blocking (CBB) and cost of these topologies develop as the size increases.
- Load is derived from the worst case, the random permutations and the group external cases described earlier.

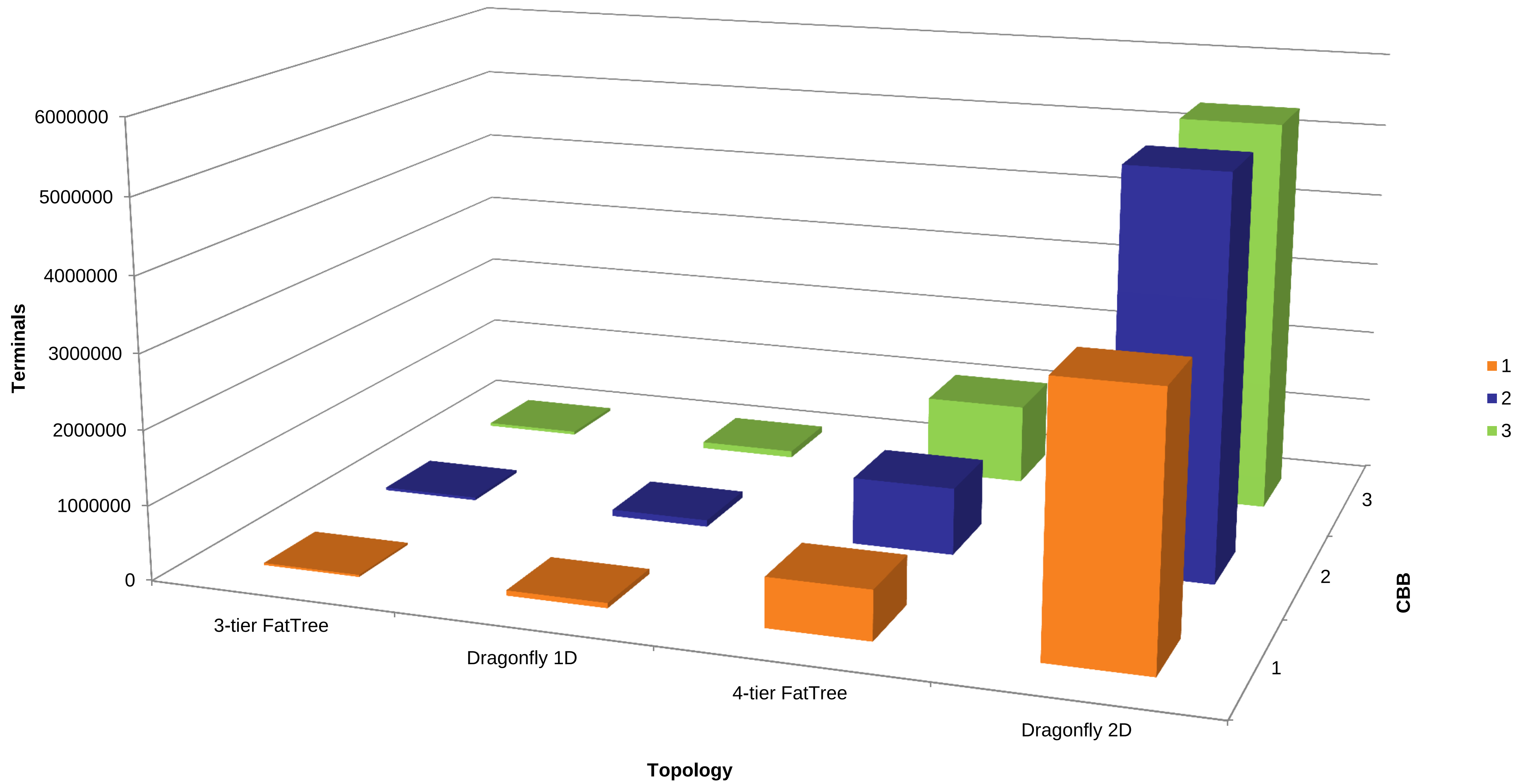
SCALABILITY AND COST

The cost of the topology is defined as:

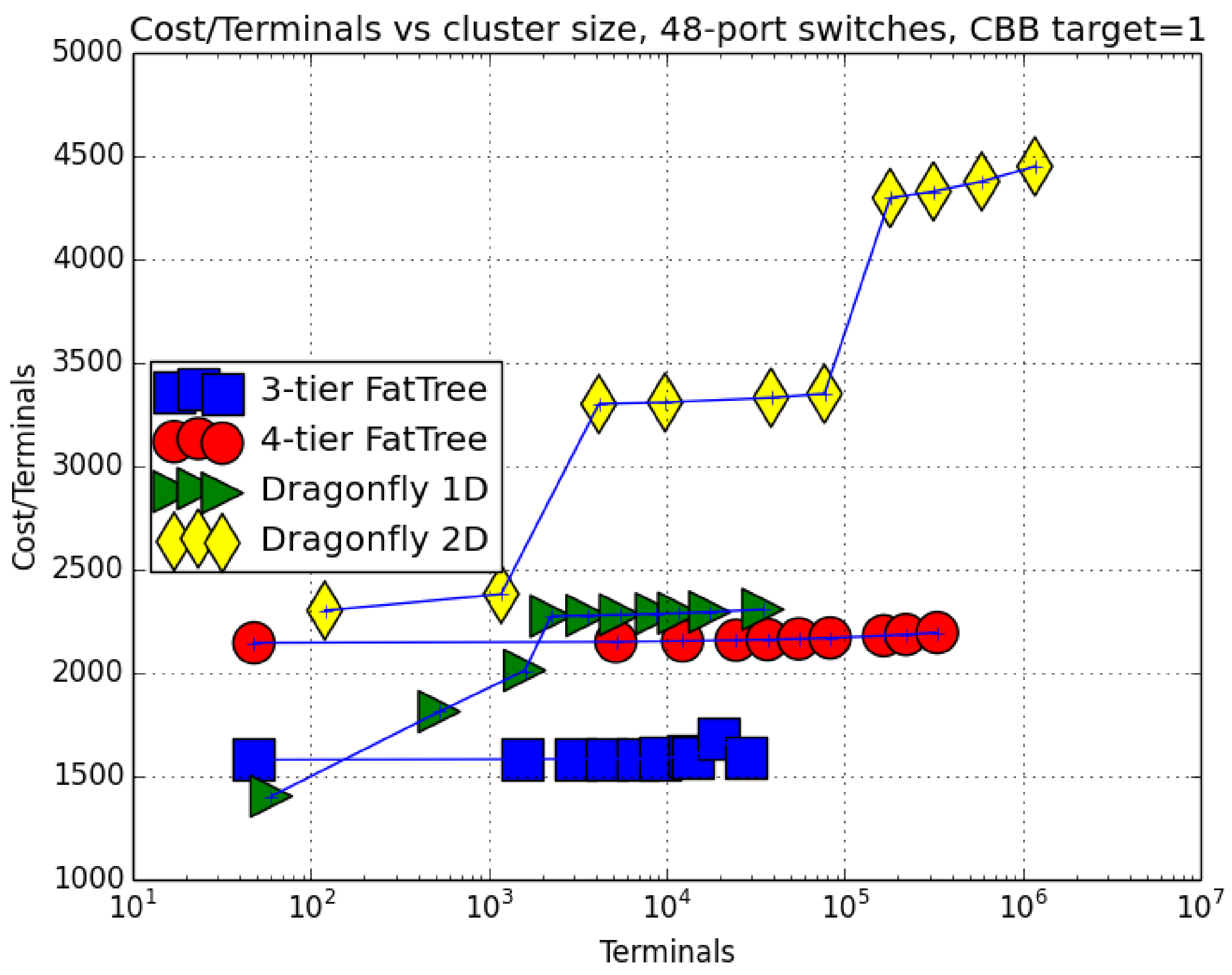
$$
\begin{aligned}
cost = \; & \#switches \times switch\_cost \\
+ \; & \#short\_links \times 2m \times cost\_per\_meter\_s \\
+ \; & \#long\_links \times avg\_length \times cost\_per\_meter\_l
\end{aligned}
$$

http://www.kernelsoftware.com/products/catalog/mellanox.html

The sizes

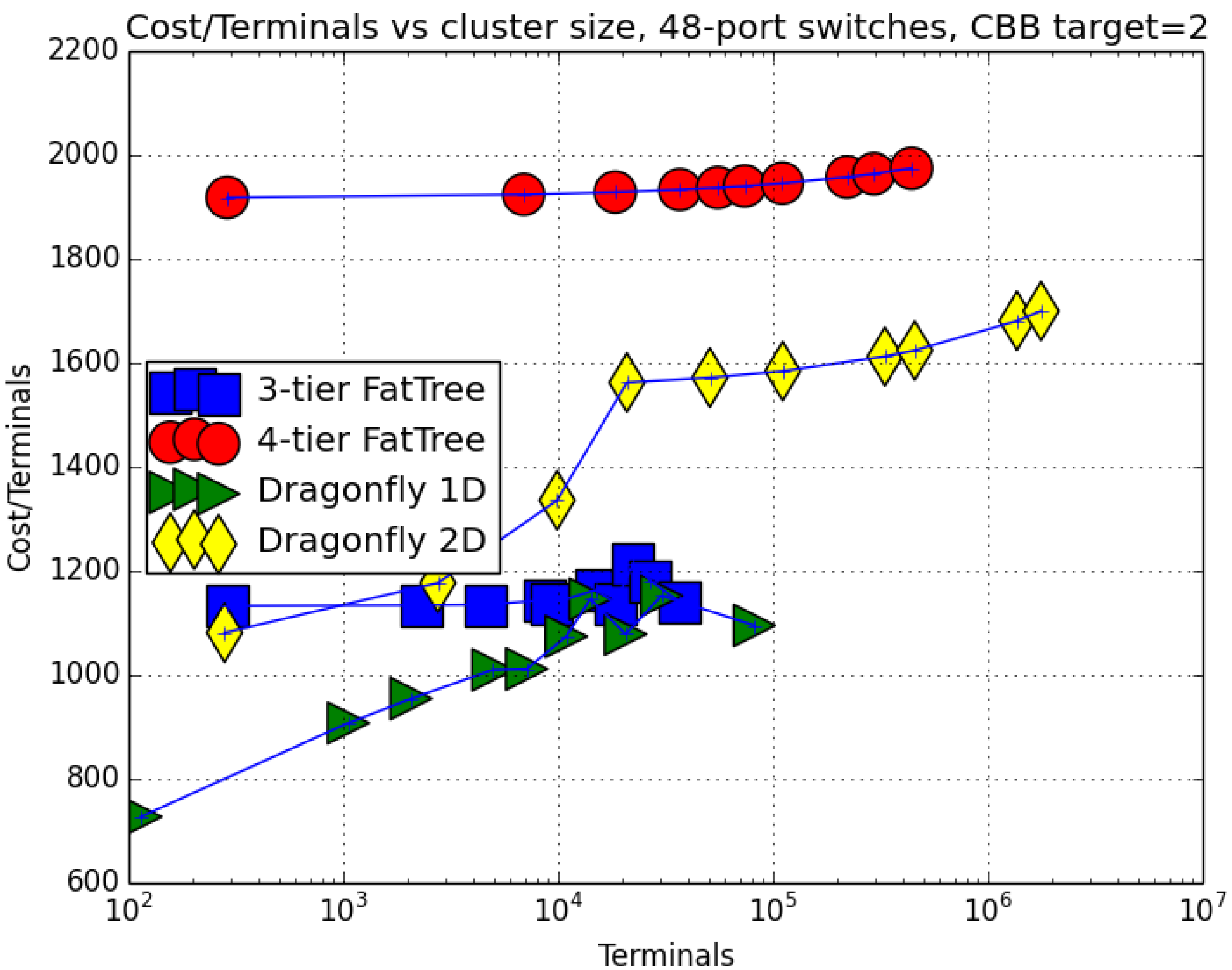

Terminals
6000000
5000000
4000000
3000000
2000000
1000000
0
3-tier FatTree
Dragonfly 1D
4-tier FatTree
Dragonfly 2D
Topology
1
2
3
CBB
1
2
3

SCALABILITY AND COST CBB = 1

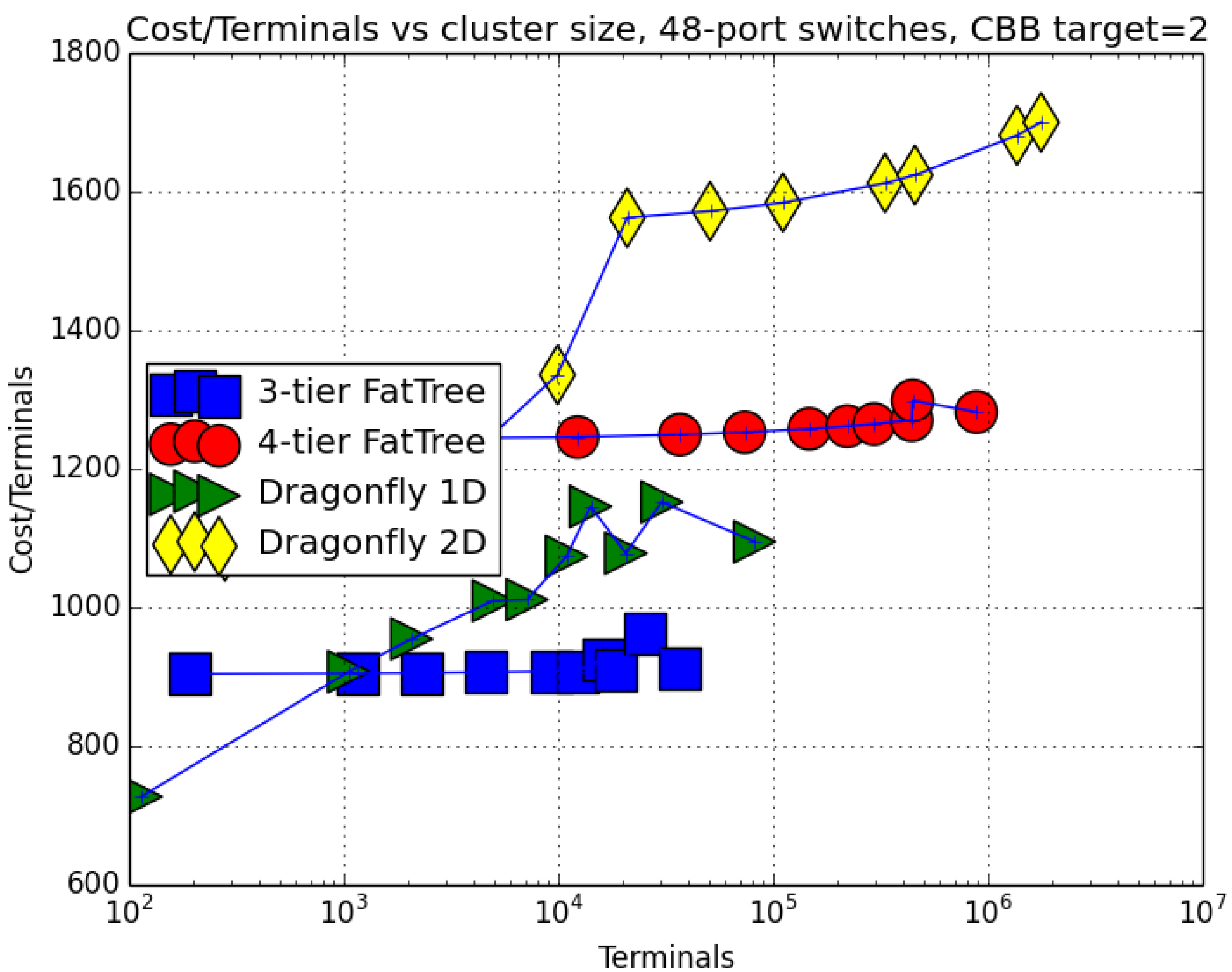


For CBB=1 the 3- and 4-tier fat-trees are more cost efficient than the dragonfly.

SCALABILITY AND COST
CBB = 2



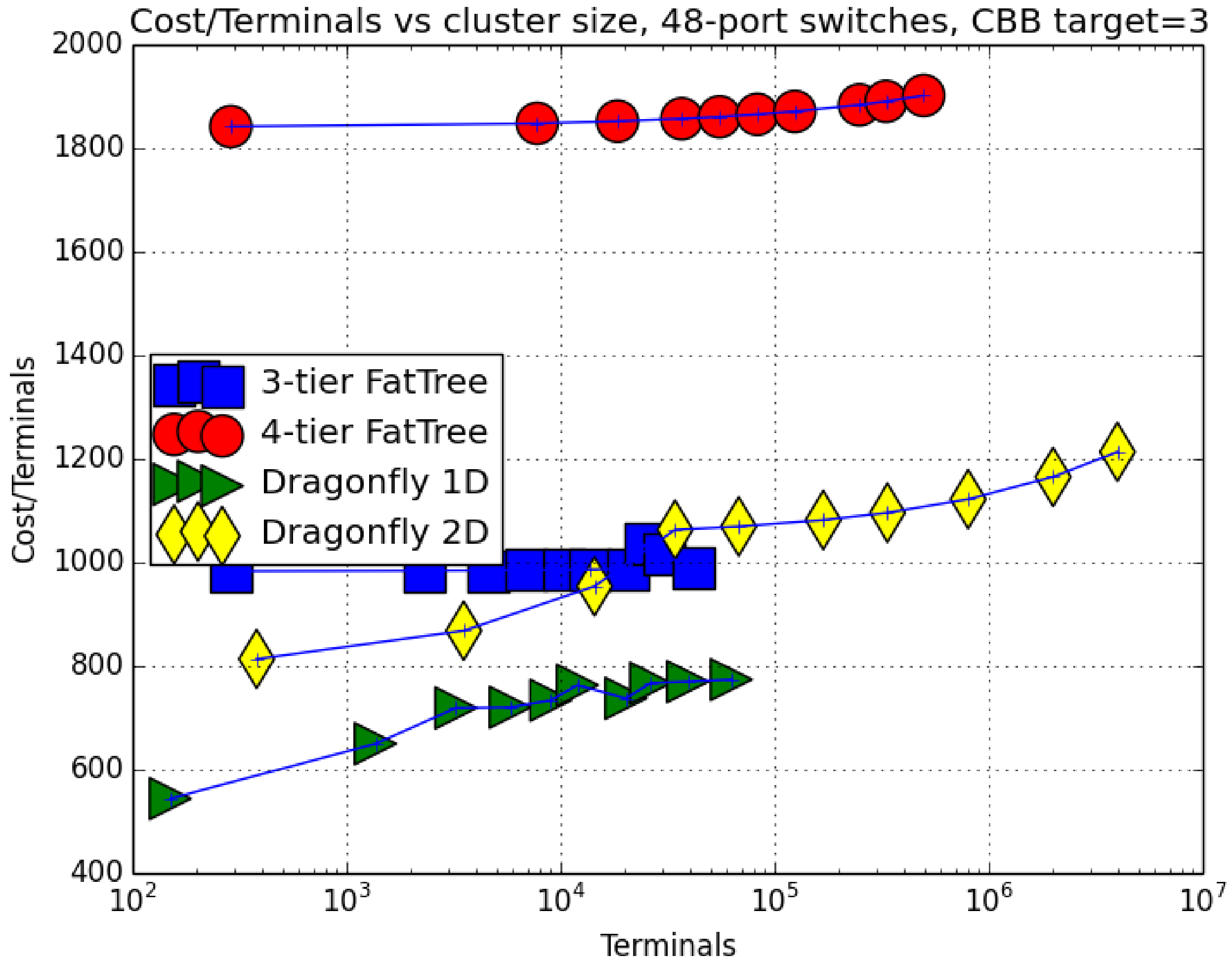
Slimming the top



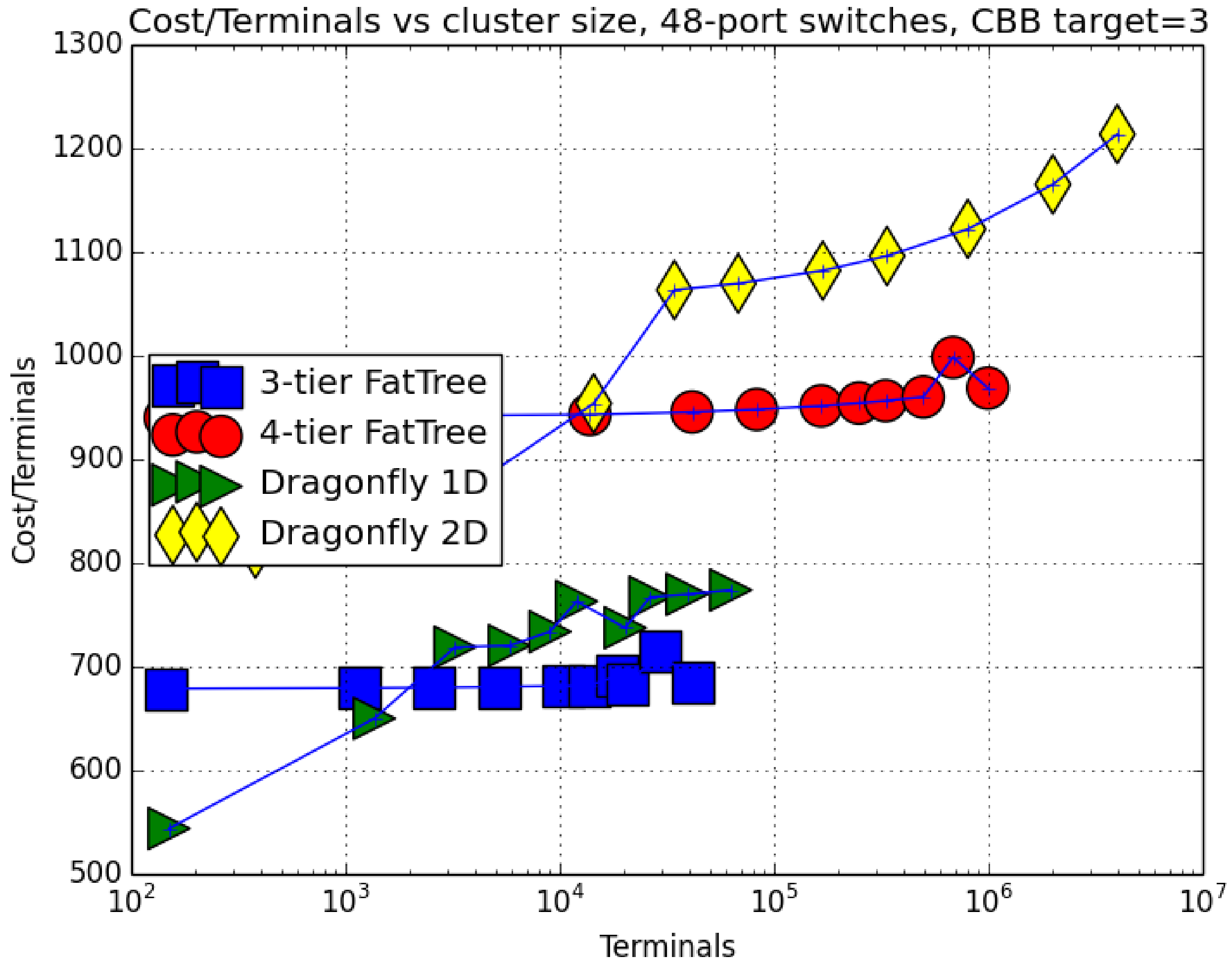
Slimming everything

For CBB=2 the dragonfly comes into its own, but depending on how the fat tree is designed.

SCALABILITY AND COST
CBB = 3



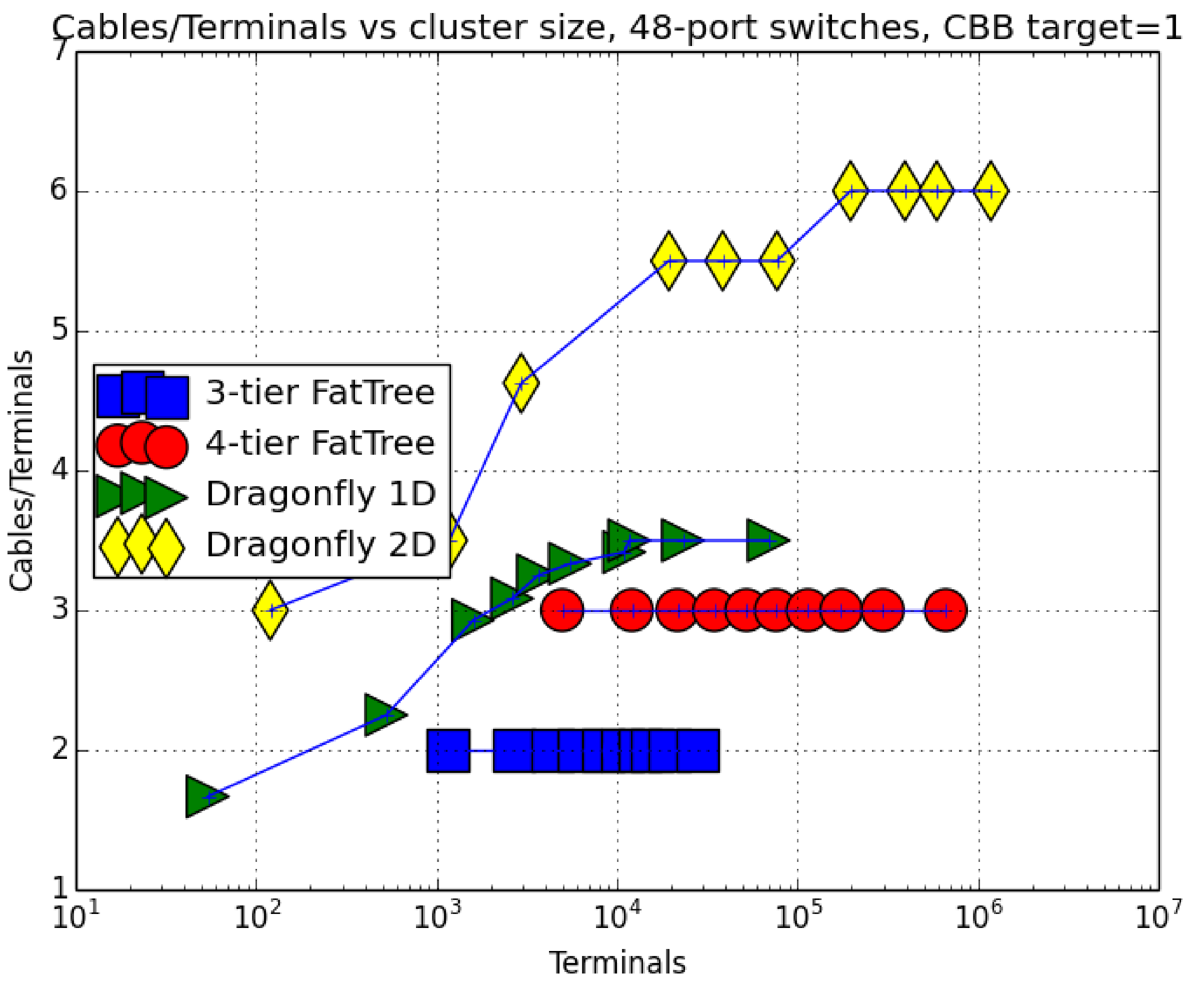
Slimming the top

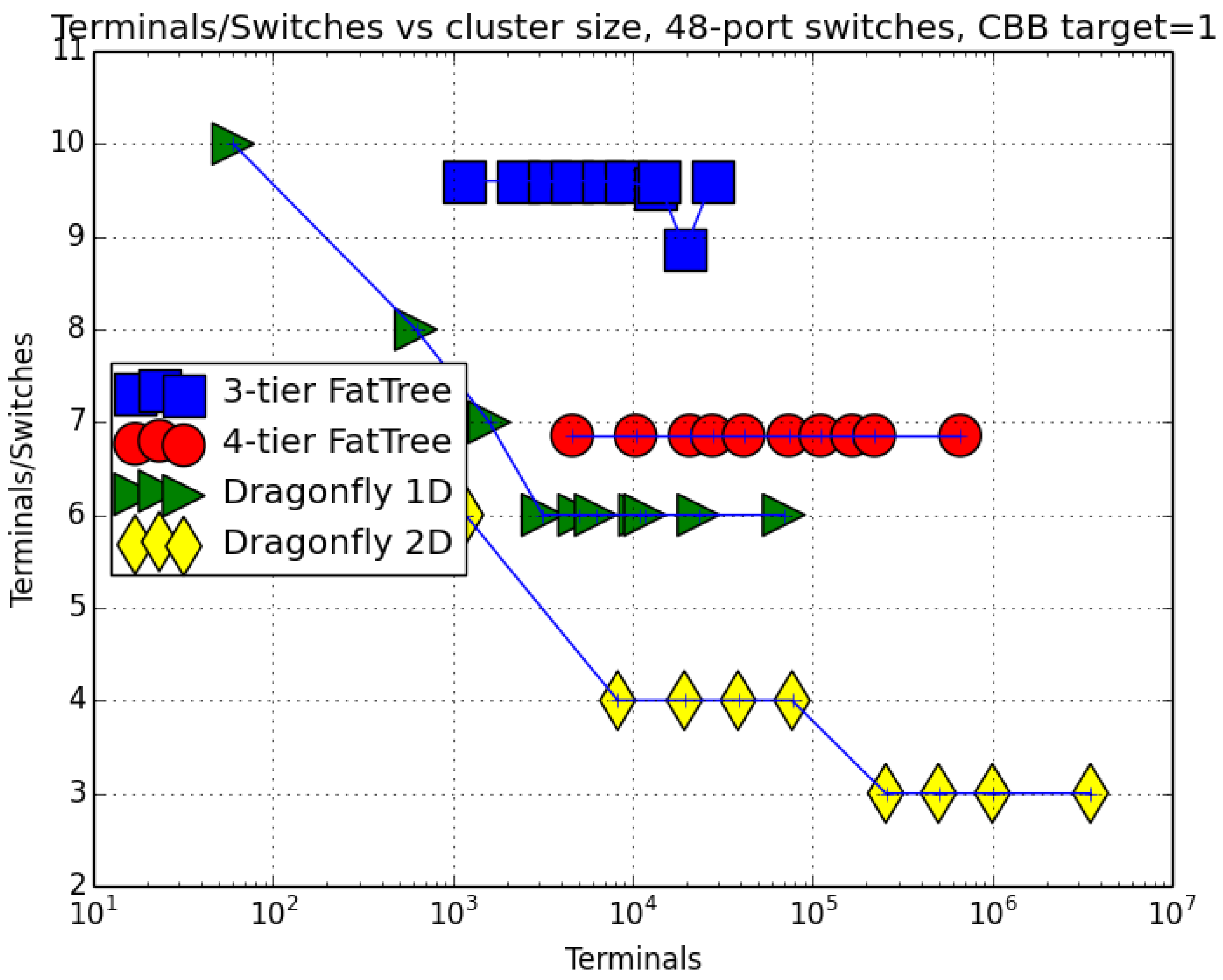


Slimming everything

For CBB=3 the table has turned in favour of the dragonfly for any topology size, even when slimming everything.

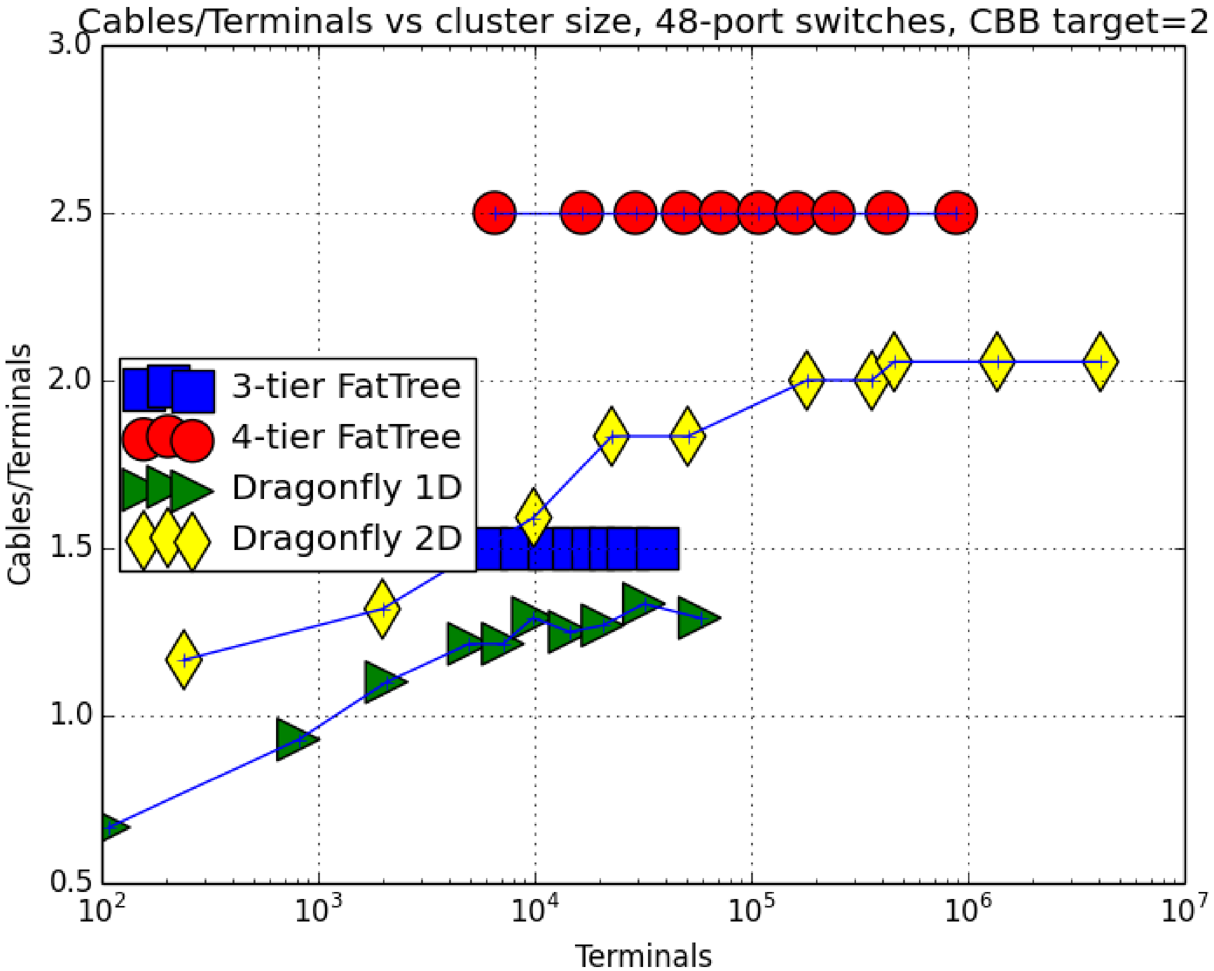
OTHER METRICS CBB = 1



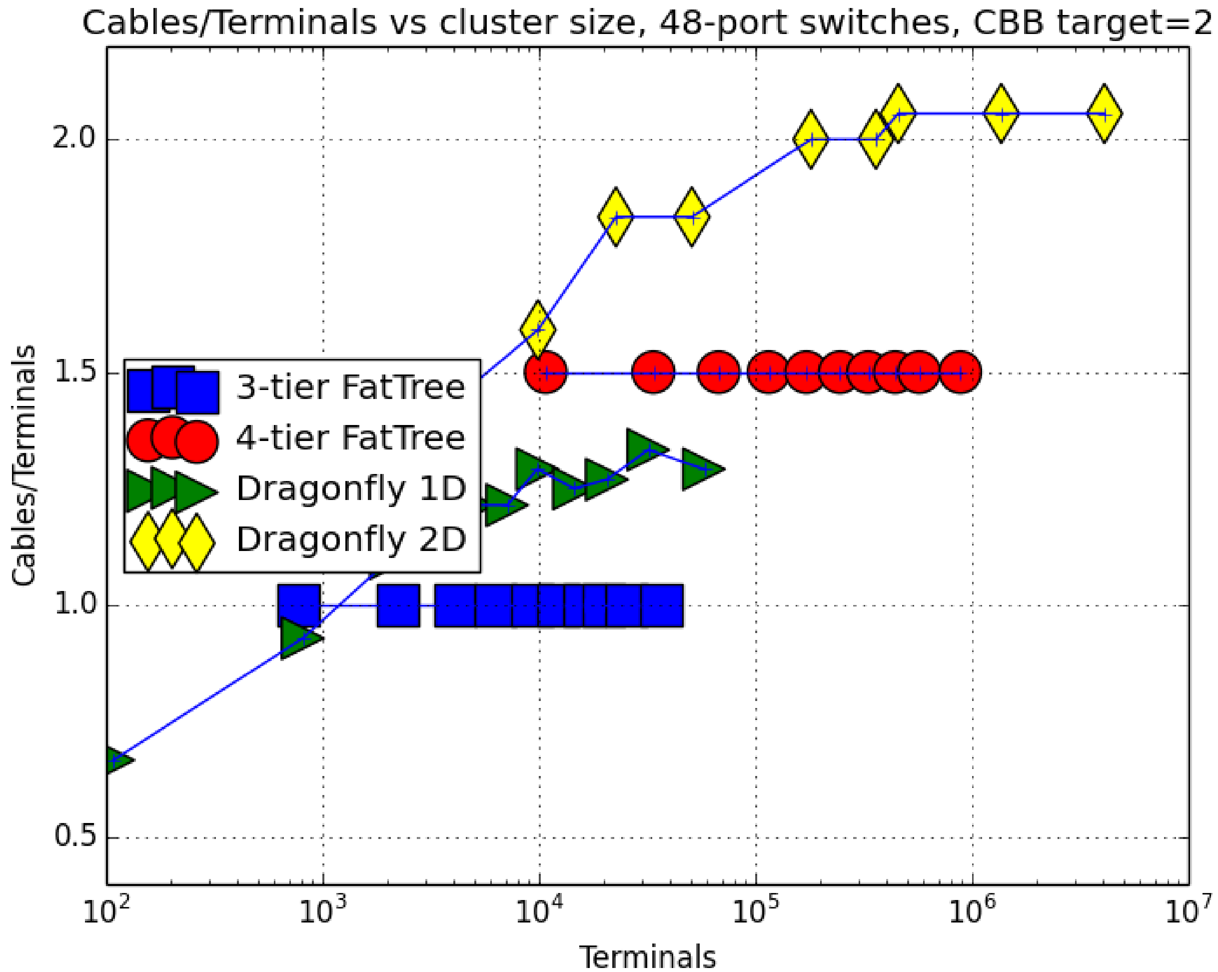


The fat trees have a much higher nonblocking efficiency in terms of cables per terminal and terminals per switch

OTHER METRICS CBB = 2



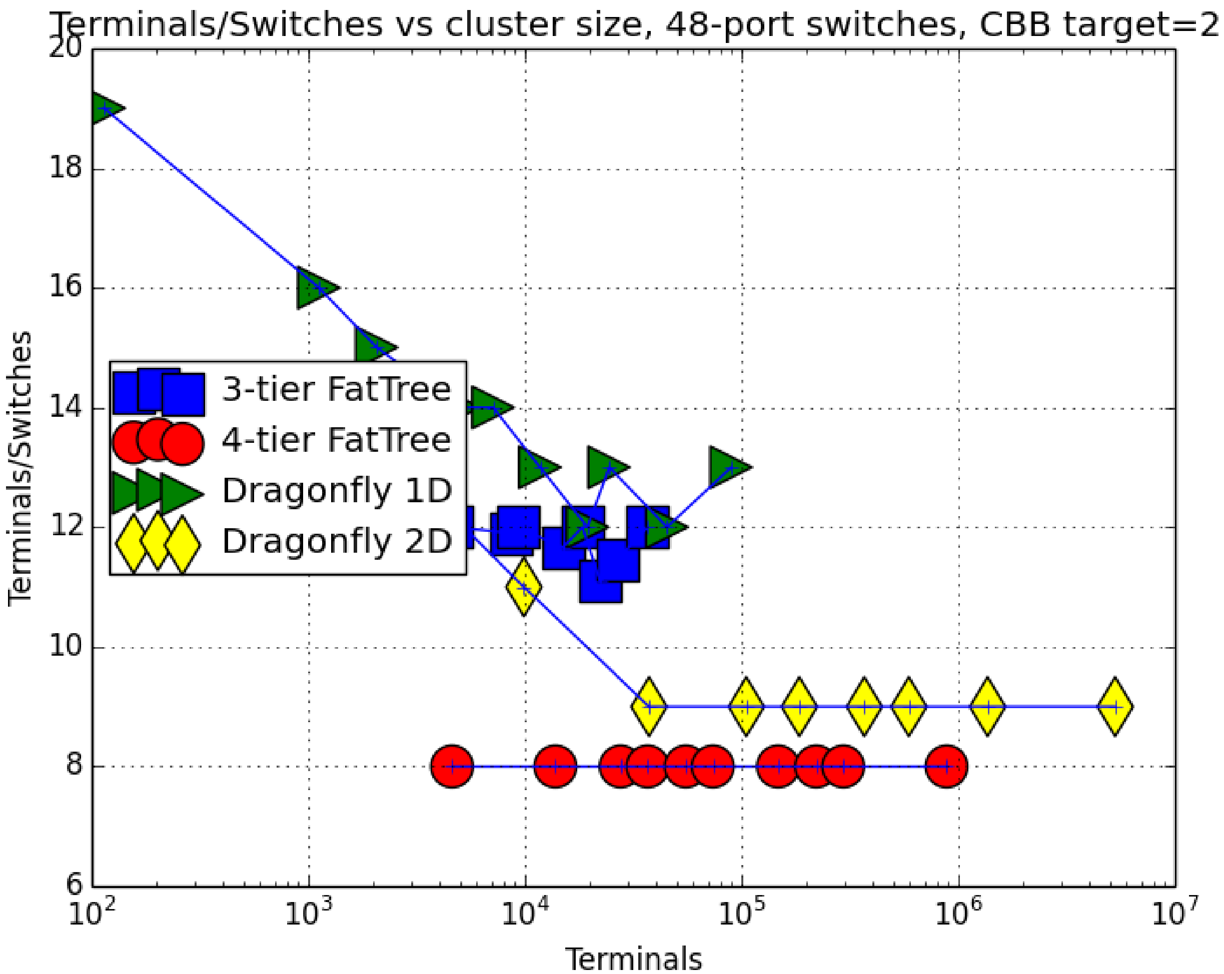
Slimming the top



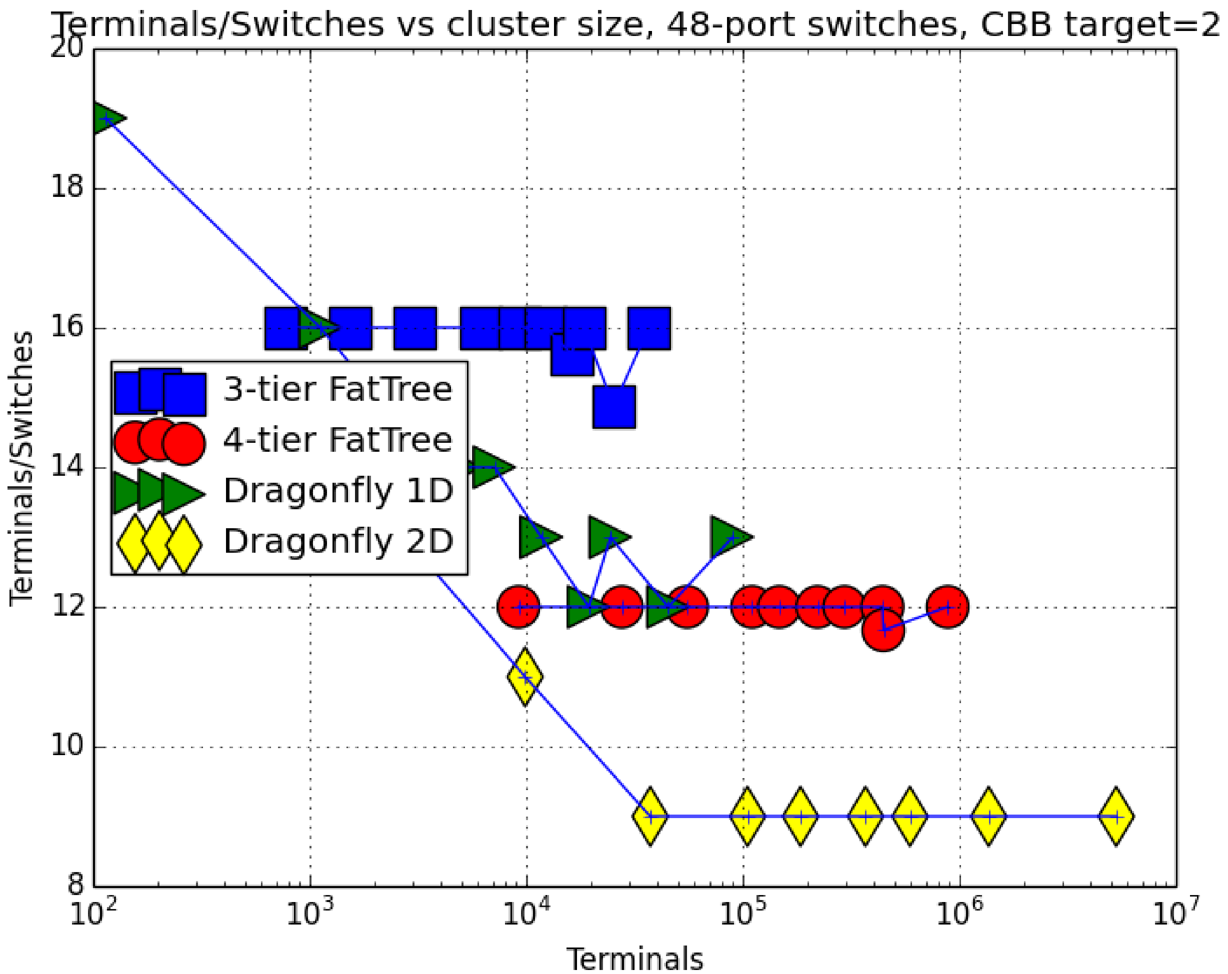
Slimming everything

With increasing CBB ratio the cost improvement of the dragonfly over the fat tree comes to a large extent from the reduction of the number of long links

OTHER METRICS CBB = 2



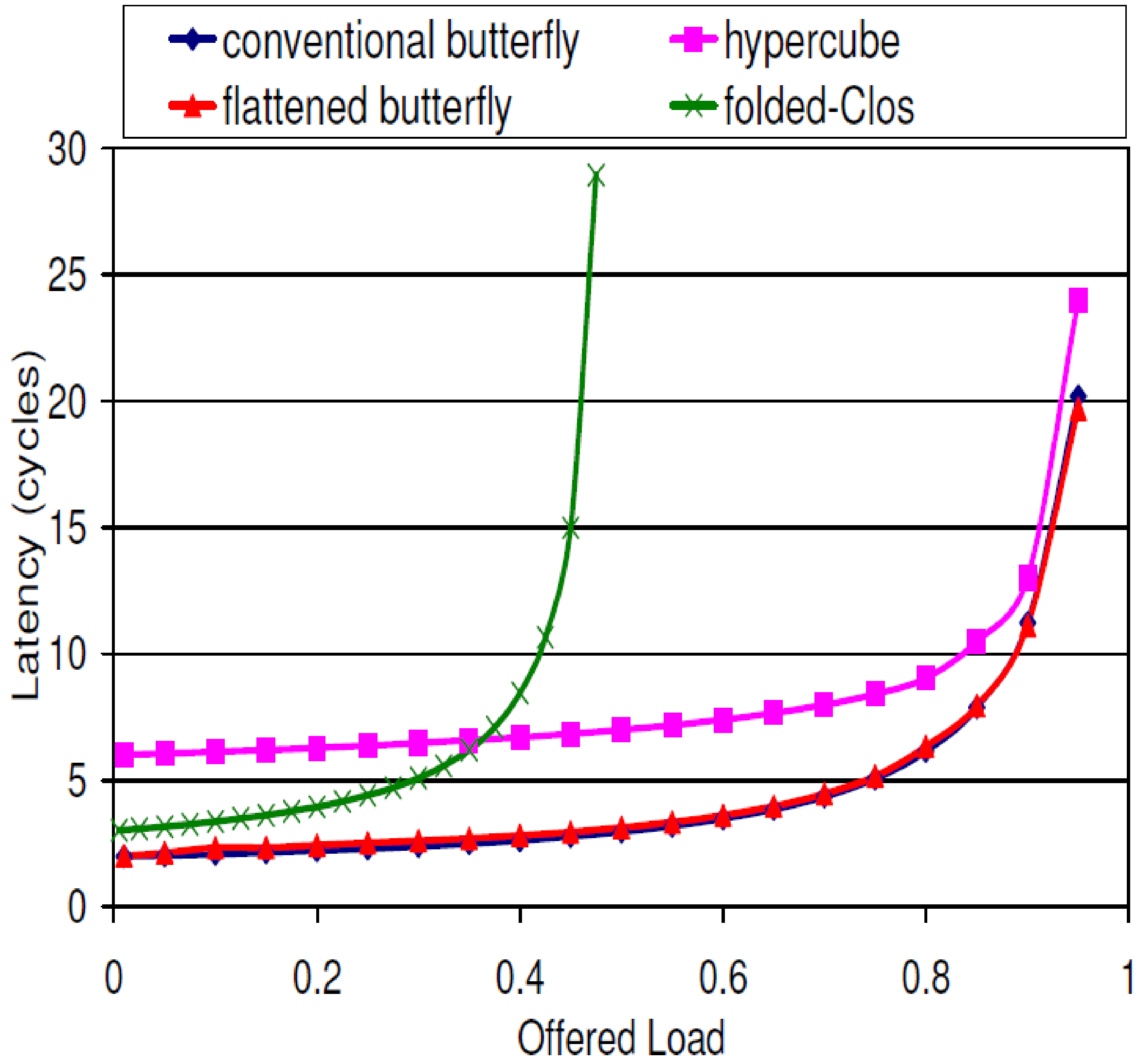
Slimming the top



Slimming everything

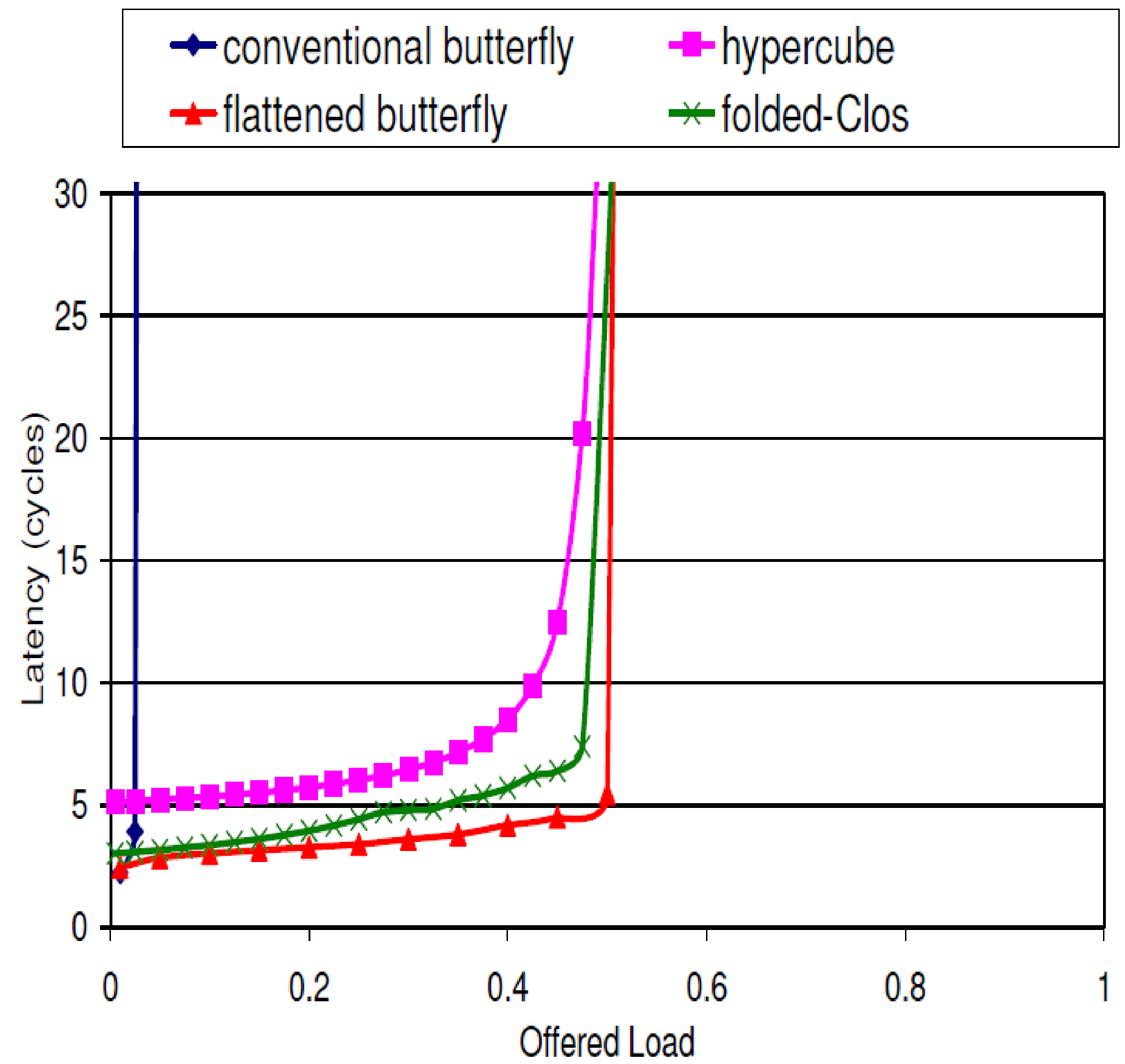
With increasing CBB ratio the cost improvement of the dragonfly over the fat tree comes to a large extent from the reduction of the number of long links

Topology performance



(a)

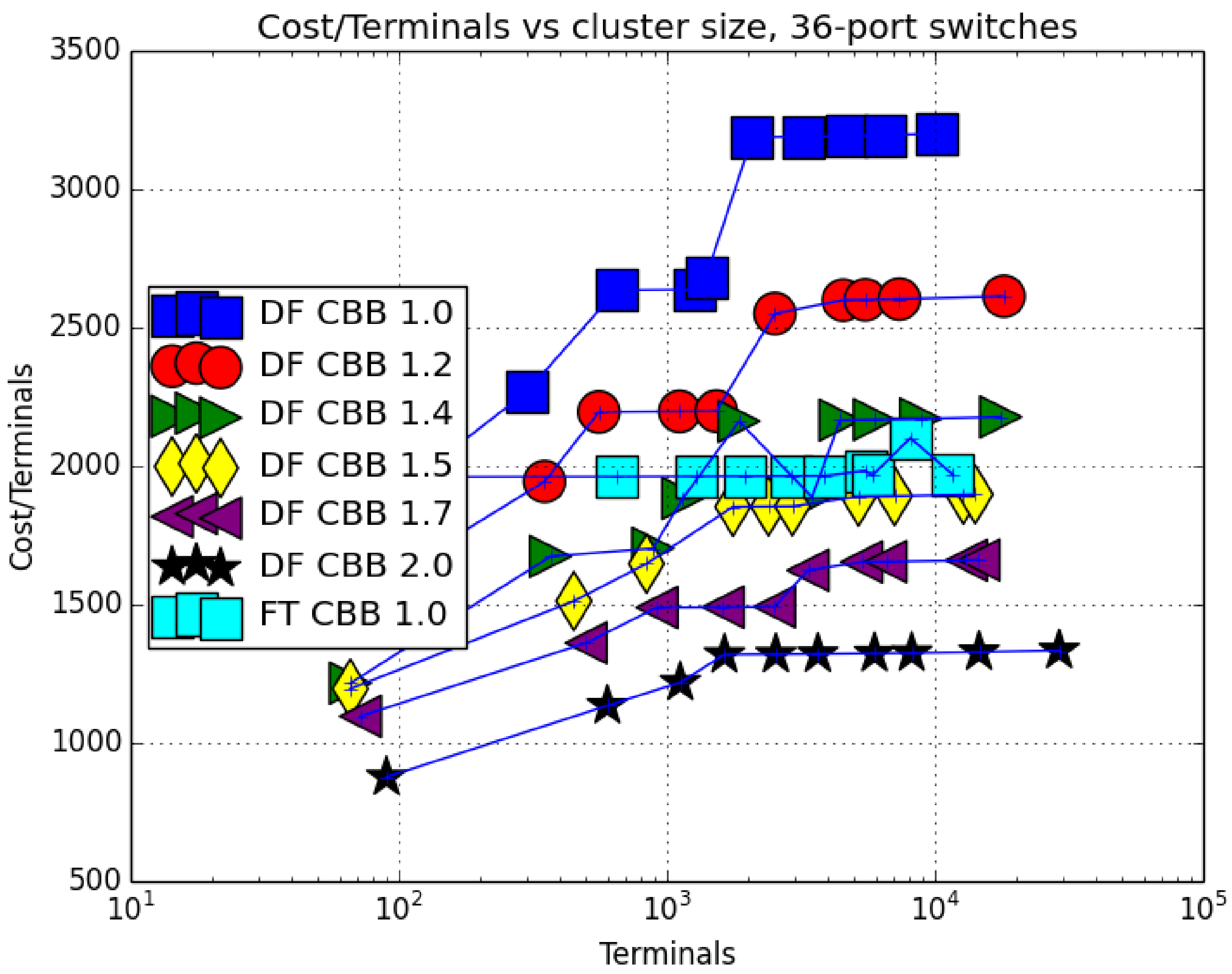
Uniform



(b)

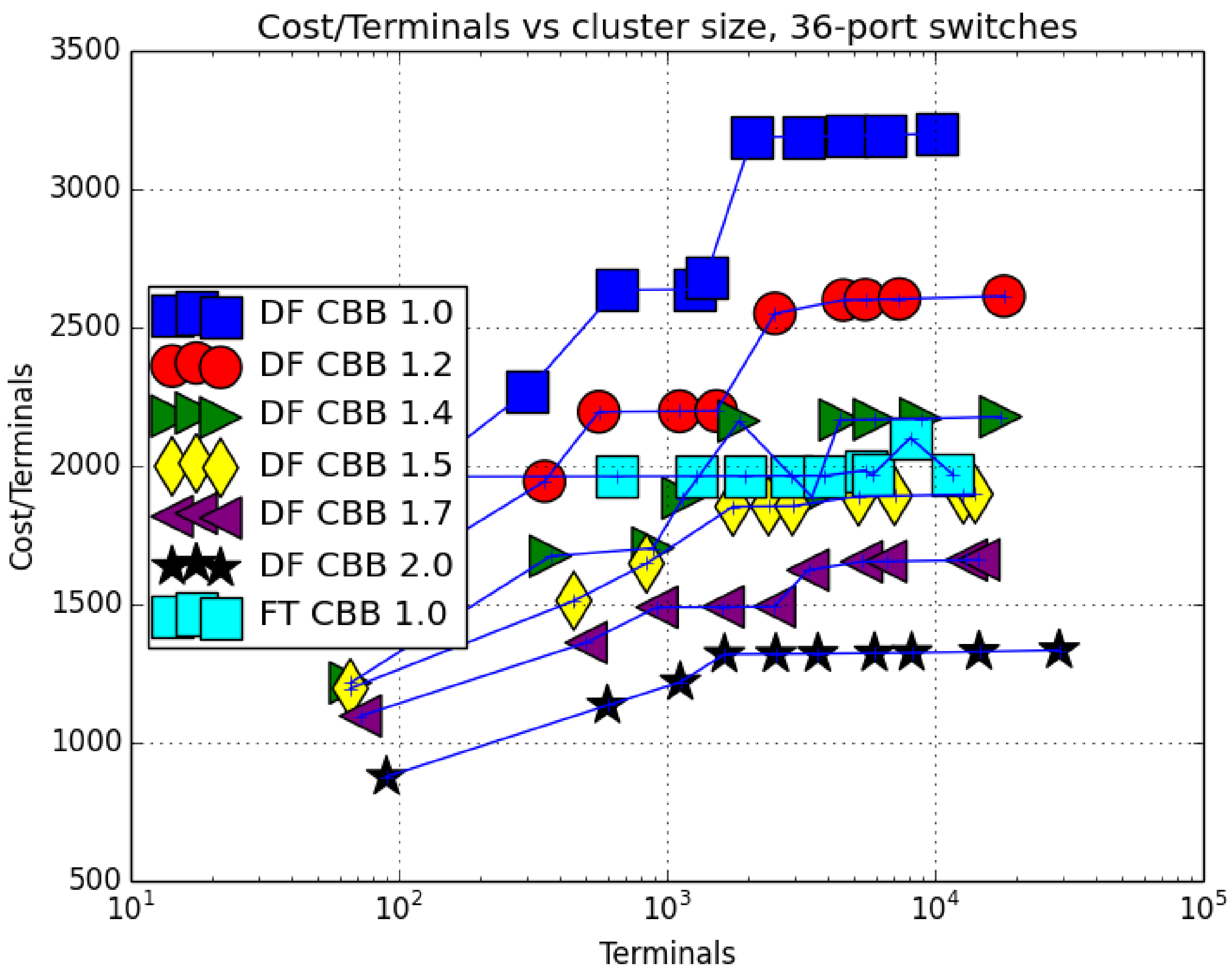
Worst case

THE CROSSING POINT



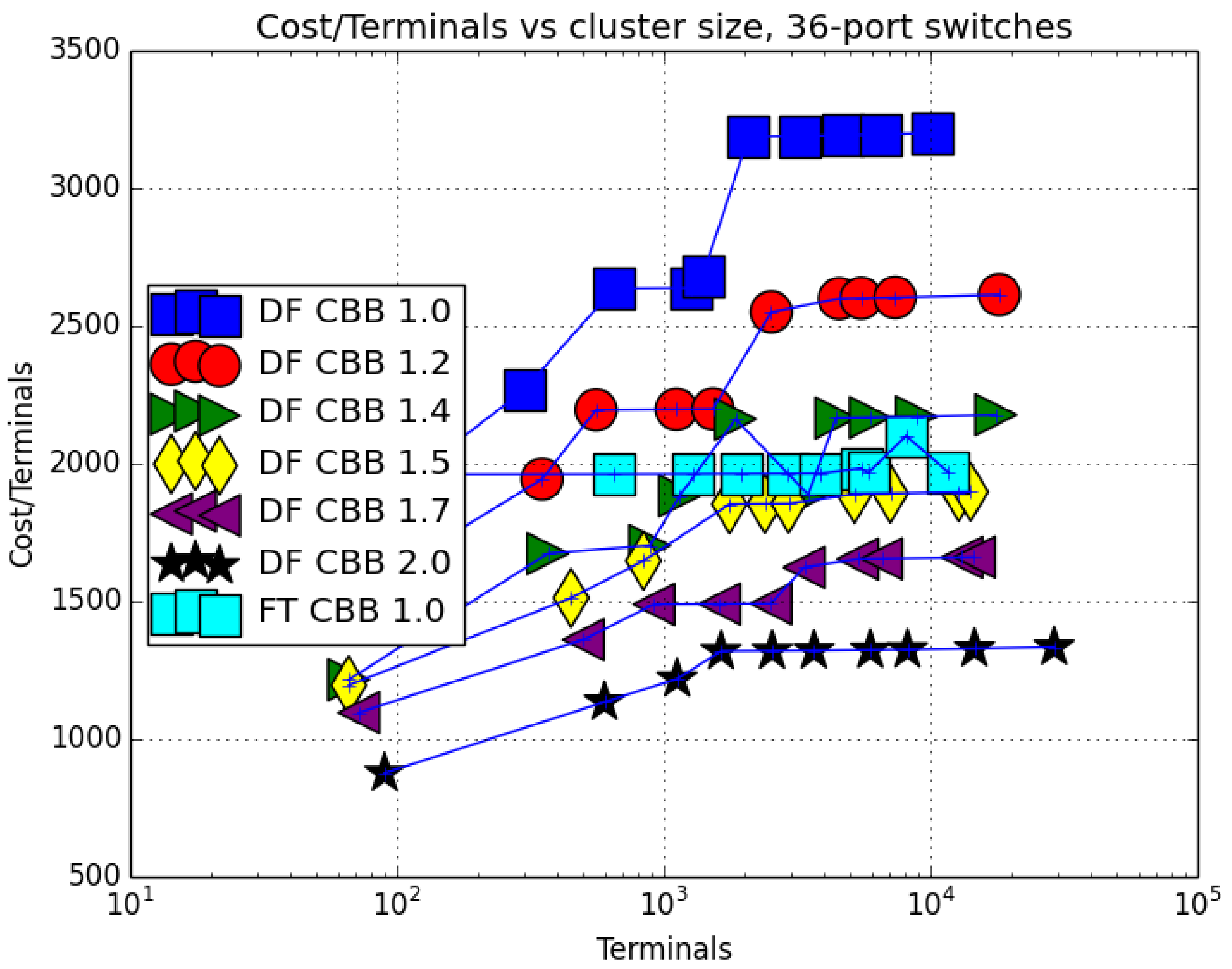
- Uniform traffic means that 50% of the traffic crosses the bisection
- Worst-case traffic means that 100% of the traffic crosses bisection
- CBB = 1.5 supports 75% of the traffic crossing the bisection

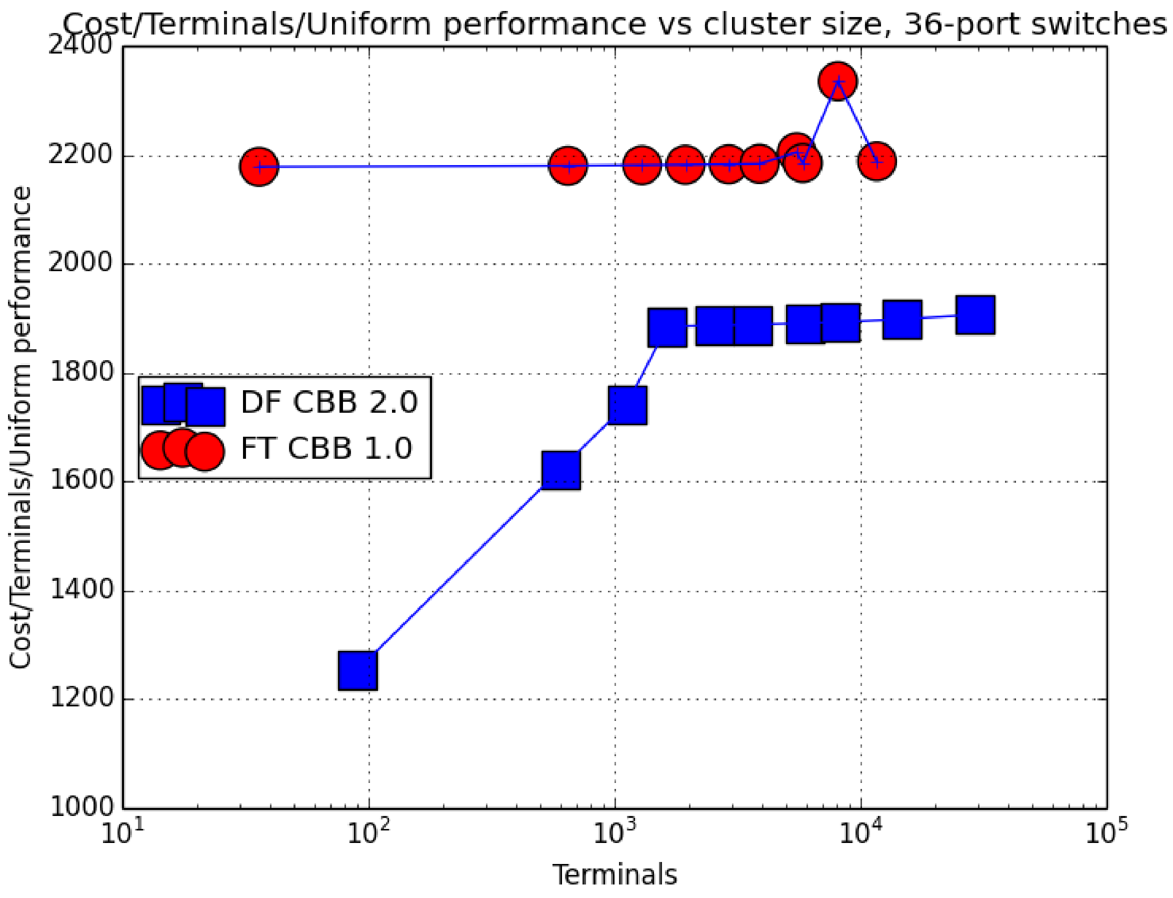
THE CROSSING POINT



- Best practical dragonfly utilisation for uniform traffic is around 70% (adaptive routing)
- Best practical fat tree utilisation per uniform traffic is around 90% (static routing)

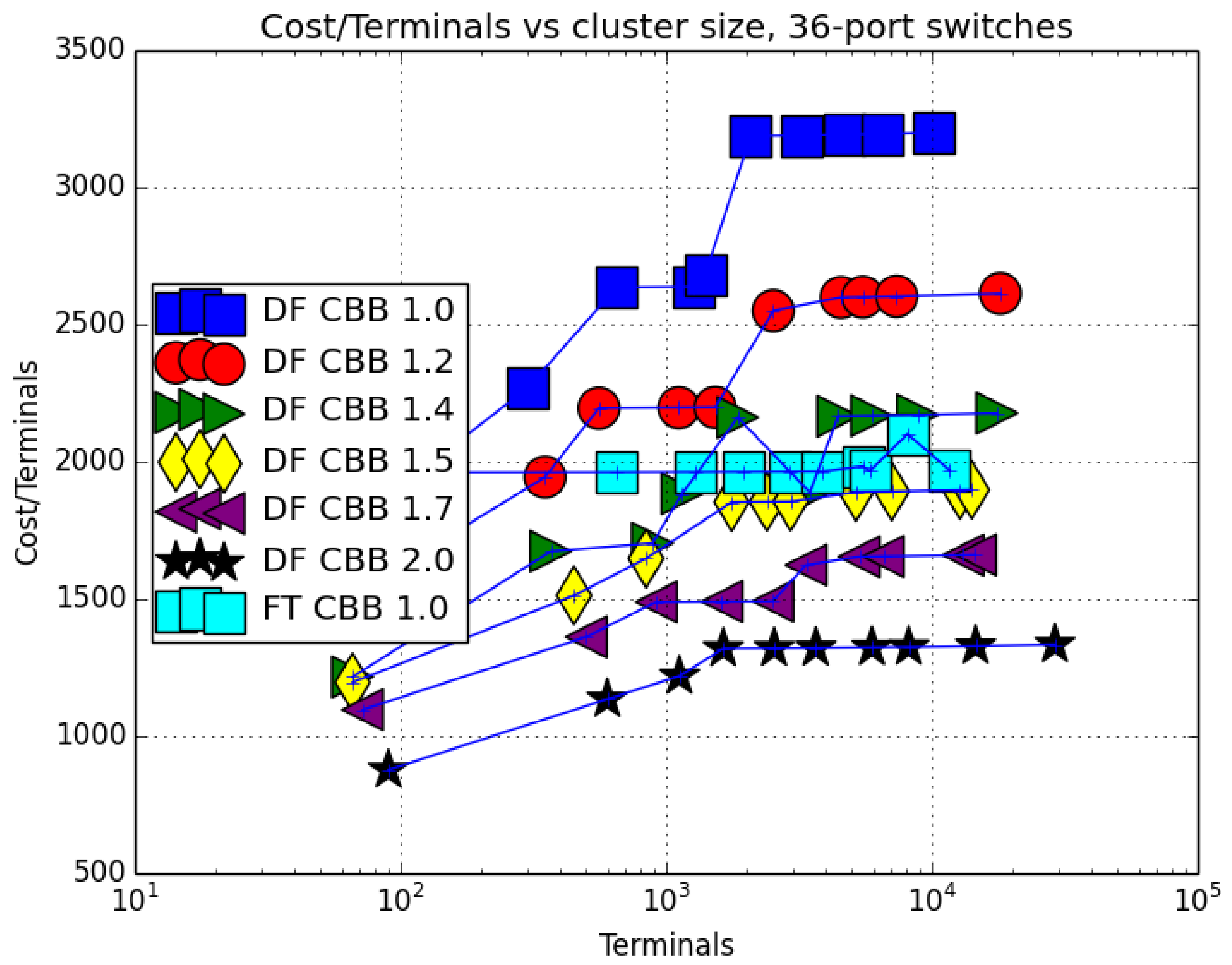
THE CROSSING POINT

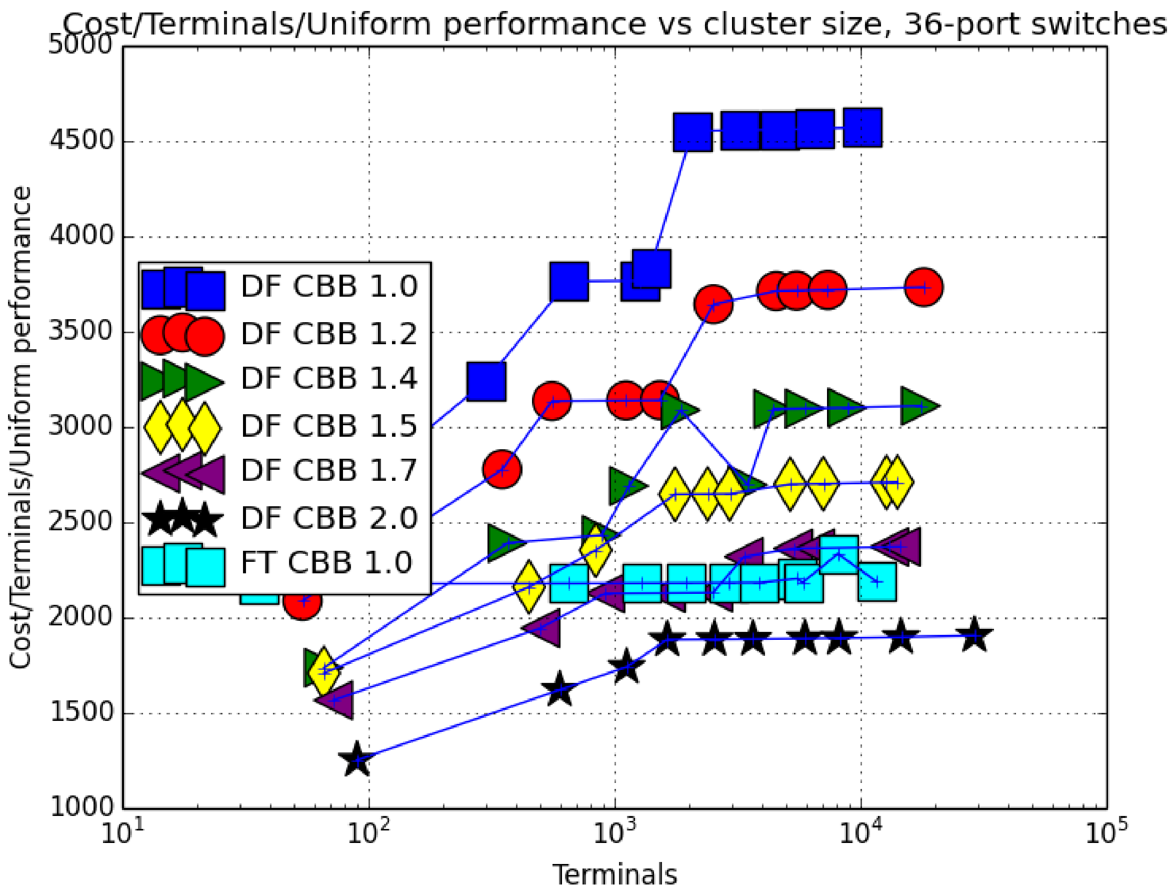




Scaling with respect to performance for uniform traffic

THE CROSSING POINT





Scaling with respect to performance for uniform traffic

CONCLUSION

Key results:

- Comparing the dragonfly topology with different group topologies to the regular fat tree topology shows that the dragonfly is the superior choice for benign traffic patterns.
- The dragonfly is better able to exploit higher CBB ratios to improve cost-efficiency
- The fat tree is the superior choice for more adverse traffic patterns, such as MPI collectives (at least with deterministic routing).
- The crossing point is somewhere around 75% of the traffic crossing the bisection (or possibly lower when considering relative topology performance).

Remember:

- The dragonfly requires support for non-minimal adaptive routing and congestion look ahead for optimal behavior, this is not supported by any existing off-the-shelf hardware, at least not with sufficient to routing performance.
- The dragonfly requires multiple virtual channels for deadlock avoidance

QUESTIONS?