LOAD SPREADING FOR SPB

DAVID ALLAN
JANOS FARKAS
SCOTT MANSFIELD

SYNOPSIS

- This presentation proposes an overall direction and an exemplar algorithm whereby an arbitrary SPB network can automatically seek to average the number of pairwise Ethernet Switched Paths (ESPs) that transit each link
  - Assumption being that this will universally maximize the headroom across the network for burst and slosh tolerance and automatically avoid the creation of “hot spots”
- For simplicity, the exemplar presented assumes all ESPs are equal, and all links are equal capacity
  - It is possible to envision variations of both, and is required in any practical instantiation, but would unnecessarily complicate presenting the basic concept
- It does this in the context of edge assignment of load to SPT Set
  - No changes to the relay function

PROBLEM STATEMENT

- SPB requires a mechanism to more effectively utilize all the links in a given network
  - In particular in scenarios where there are multiple equal cost paths between any two points in the network
- The technique of selecting low-high bookends in tie breaking works well where there are typically two diverse paths between any node pair, but the techniques for going beyond two paths has limitations
  - Tie breaking on the basis of lexicographic ordering of node IDs produces significant diversity between the bookends of any given selection but…
  - Repeated selection via algorithmic manipulation of node IDs and re-tie breaking only produces pseudo random results as each selection is performed independently
    - Significant dilation required to ensure all links utilized which also results in excessive amounts of state
    - Even amongst the utilized links, the distribution of load is uneven

A technique is required to achieve uniform link utilization in networks with very dense mesh

RETHINKING THE PROBLEM

- We require a mechanism such that path selection seeks to explicitly avoid paths already selected in previous selection steps
  - Make avoiding paths with many ESPs explicit, not random
- We are making a mistake only concerning ourselves with path diversity between arbitrary node pairs, we need to consider the overall distribution of ESPs in the network
  - Else we have the same situation as with algorithmic manipulation of node IDs, just on a larger scale, individual iterations produce good results in isolation but the sum of the parts is unsatisfactory because the components are selected independently
- We are overly afraid of using computation despite the fact that it is the real innovation in 802.1aq
  - Commodity technology and a few very cool bits of computing
- So we will postulate a criteria of goodness as minimizing the co-efficient of variation (CV) of the number of ESPs that transit each link for a given network
  - Try to use each link equally
- And propose an algorithm that actively seeks to minimize the CV across the network *without* being afraid of computational complexity
  - We can optimize it once we figure out where we want to go…

THE PROPOSED ALGORITHM

- Is quite simple….
- A first pass through the database generates the low-high ECT set as is currently described in 802.1aq
- During this process, the number of shortest paths transiting each link is recorded…
  - The cumulative number of pairwise Ethernet Switched Paths (ESPs)
- Two-stage tie breaking is used for equal cost paths in subsequent passes
  - The sum of per link ESPs for each equal cost path is generated
  - If there is a path with a uniquely lowest sum of ESPs, then it is selected
  - If there are multiple paths with an equal lowest sum, then tie breaking among them reverts to the lexicographic sort of node IDs

All key properties of the original tie breaking algorithm are preserved

THIS IS COMPUTATIONALLY EXPENSIVE

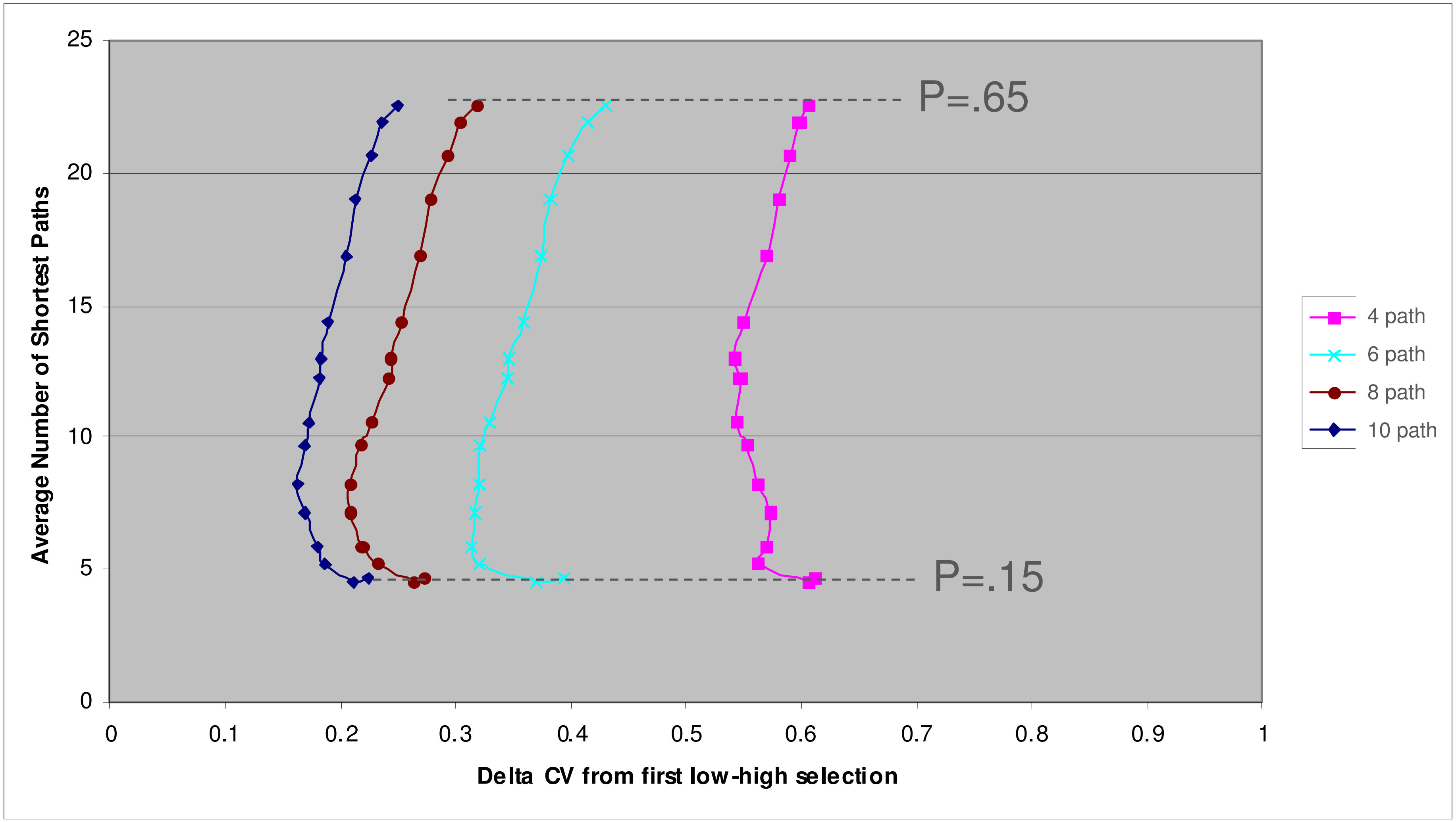
- And if it took a number of passes through the database to produce a useful result the technique would be useless…

- However the majority of the benefit accrues quickly…
  - And gives us a good compute vs. state trade off

- Graph on the next chart illustrates the benefits of the technique…
  - Based on Erdős-Rényi model of 150 node network
    - Probability of an adjacency ranges from .15 to .65
  - Illustrates the delta in CV of ESPs transiting each link (ratio of how much the STD DEV of ESPs changed) for subsequent passes after the initial pass performed as per 802.1aq today
    - Also assumes low/high ranking ECT selection on each pass
      - Selecting a single path per iteration is more efficient of state, but is computationally more complex

EXAMPLE RESULTS

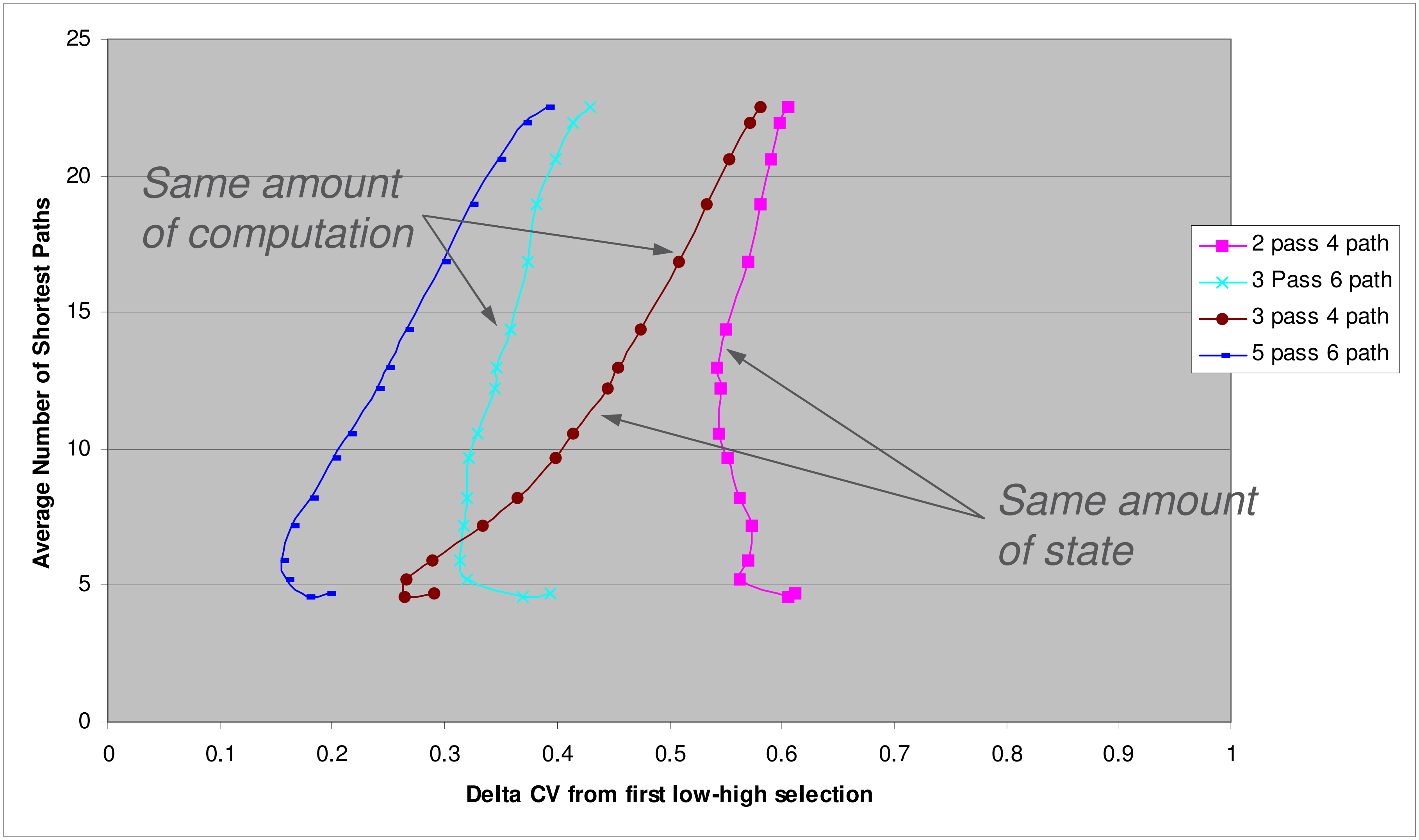
SELECTING 2 PATHS AT A TIME



Average of 5 randomly generated 150 node networks for each of 16 different mesh densities

Example Results

Selecting 1 path at a time vs. selecting 2 at a time



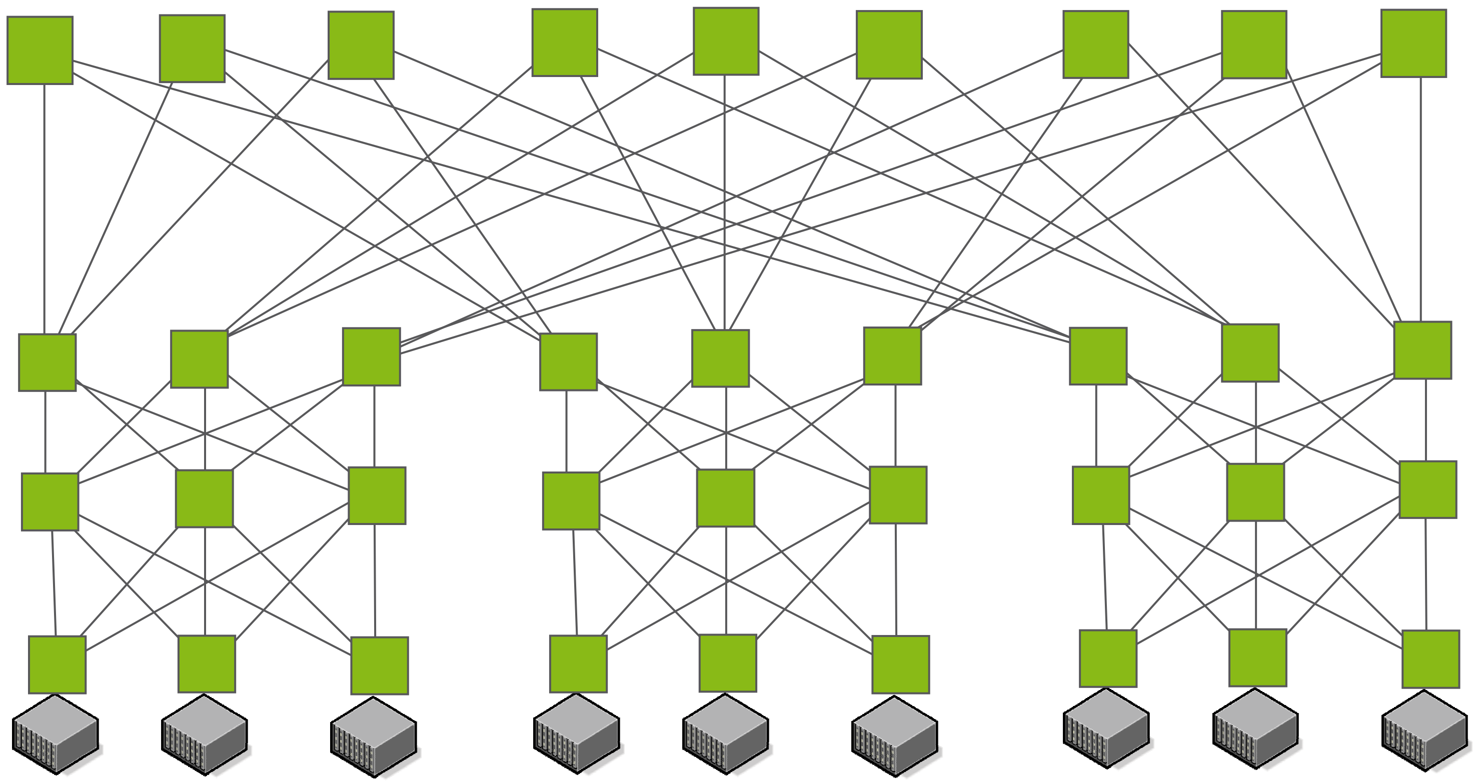
Illustrates the Compute Complexity vs. State Tradeoff

OTHER INTERESTING THINGS WE CAN DO WITH THIS….

- If we have a “hot link” in the network we can administratively bias it with an artificial count of ESPs in order to move some load off it
  - We have a knob we never had before of great subtlety
    - It does not change metrics, or hops, just affects ***how often*** the link is selected in equal cost tie breaking, and traffic displaced by the technique tends to be diffused in the rest of the network
      - It does a very graceful job of squeezing the balloon….
- This provides SPB with the reactive tool for automated traffic engineering that to date it has been missing

BUT THIS LOOKS EXPENSIVE FOR LOTS OF ECTS?

- A few ECTs is likely OK for your average metro/enterprise network
- Datacenters tend to by clos or “fat tree” architectures…very densely meshed but very regular…
  - e.g. GFT(3,3,3) below which optimally needs 9 sets of ECTs



APPLYING THE ALGORITHM TO FAT TREES

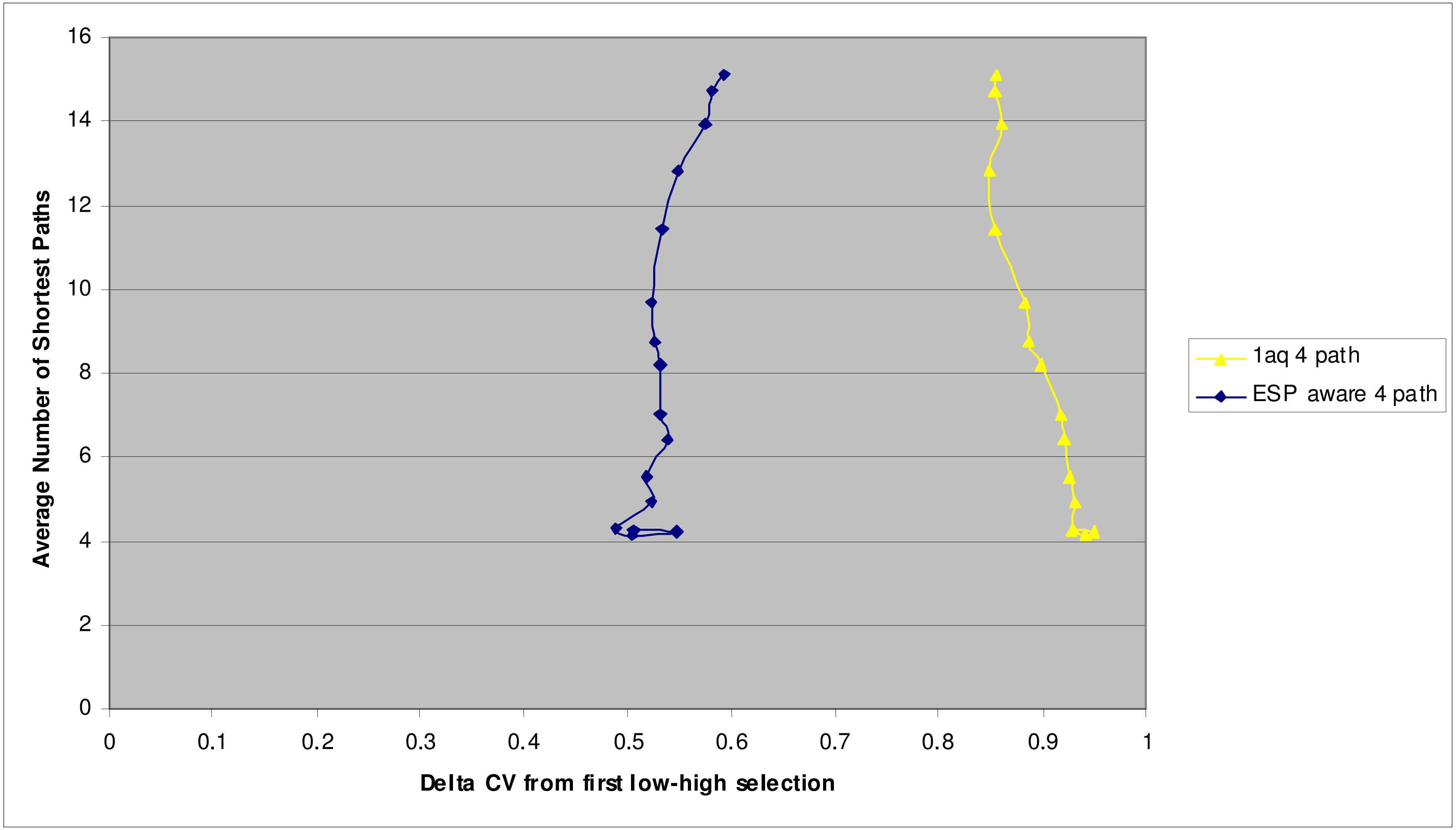
- Multiple spanning trees administratively rooted on the top tier nodes are an excellent fit with the architecture…
  - And far more computationally tractable than multiple “all pairs”
- The issue is maximizing the routing diversity between the spanning trees
- The algorithm works just fine in this scenario
  - Where you have a regular number of links at each level of the hierarchy, corresponding to the number of roots you’ll get perfect diversity of paths
  - Each iteration of generating a spanning tree will explicitly avoid the links selected in previous iterations
  - And when a failure occurs, it will still “seek” the unloaded paths to minimize the perturbation of the traffic matrix
  - Spanning tree is O(roots*N*log N) complexity, radical reduction in overall required computation
- Loss of a root reverts to “ESP aware all pairs” for that B-VID such that the load is diffused across the remaining roots
  - Move computation from specialized to general case

In Conclusion

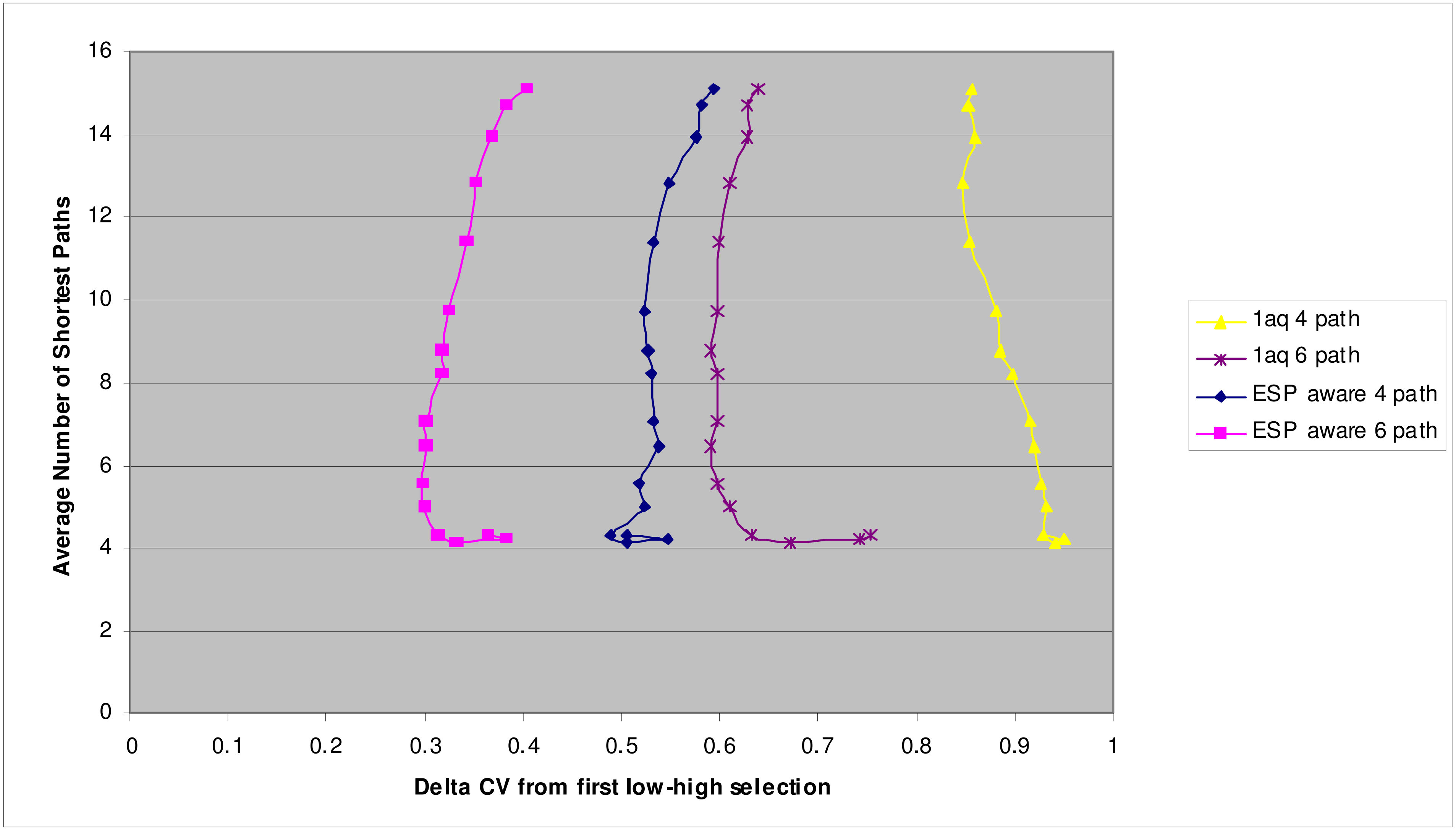
1. Using the ESP count resulting from previous passes of the database as an input to tie breaking for subsequent ECT generation significantly evens the load in the network
   - And the majority of benefit occurs in very few iterations of ECT generation
2. Administrative biasing of the ESP count for a link when used as an input to tie breaking adds a reactive automated traffic engineering tool to 802.1aq
   - Not what this activity was looking for, it is simply serendipitous
3. Highly regular datacenter architectures are best served with algorithms simpler than the generalized “all pairs” in 802.1aq
   - And these can take advantage of the technique of ESP aware tie breaking
4. All of this can be achieved purely with computation and within the bounds of the existing Ethernet technology base
   - No changes to the frame format, no changes to the silicon, no changes to the OAM
5. …and we’re not actually done yet… this is simply an indication of what can be achieved
   - Although further improvements are more “polish” than “profound”

BACKUP

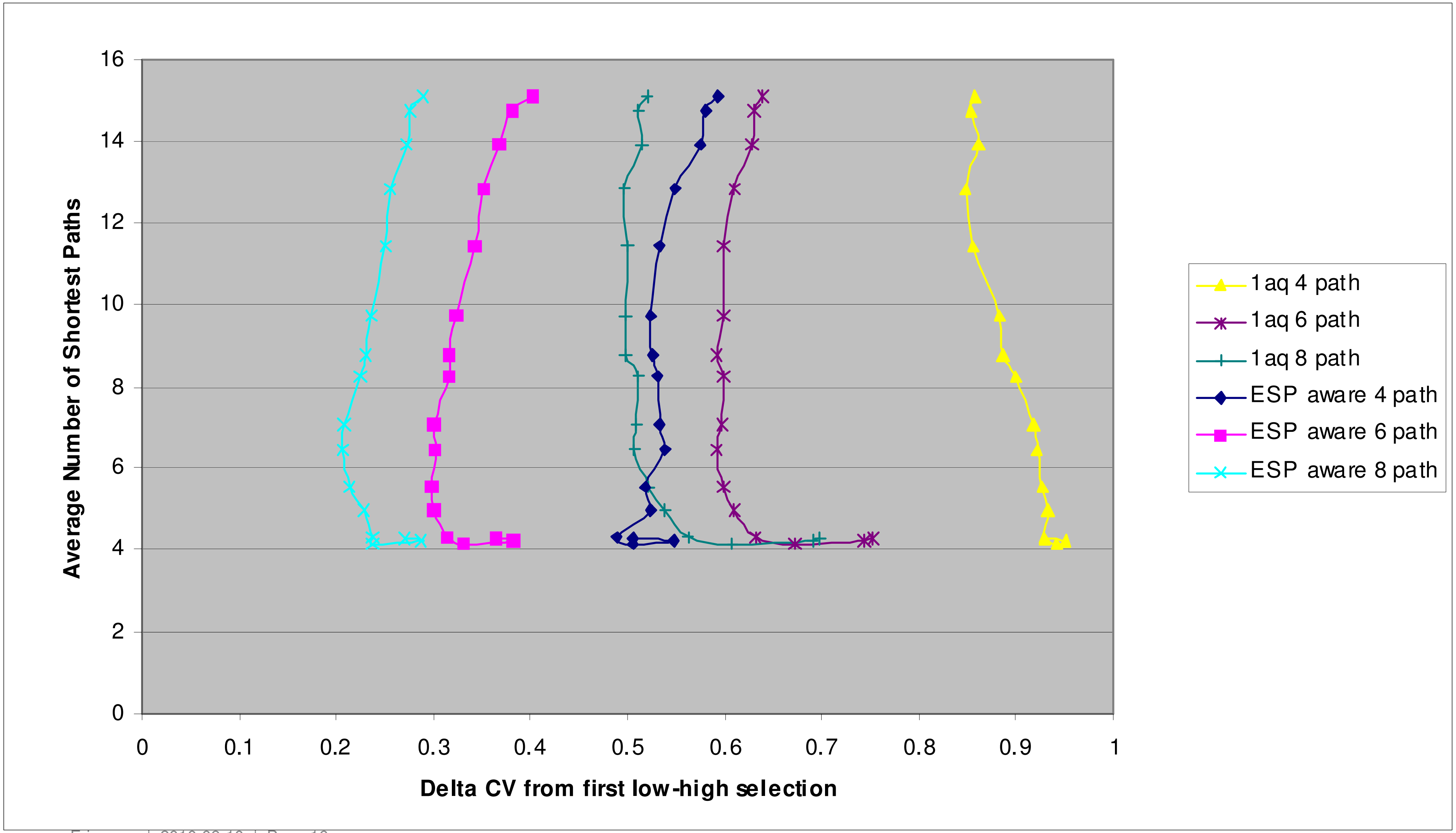
COMPARISON OF ESP AWARE VS CURRENT .1AQ (ER100)



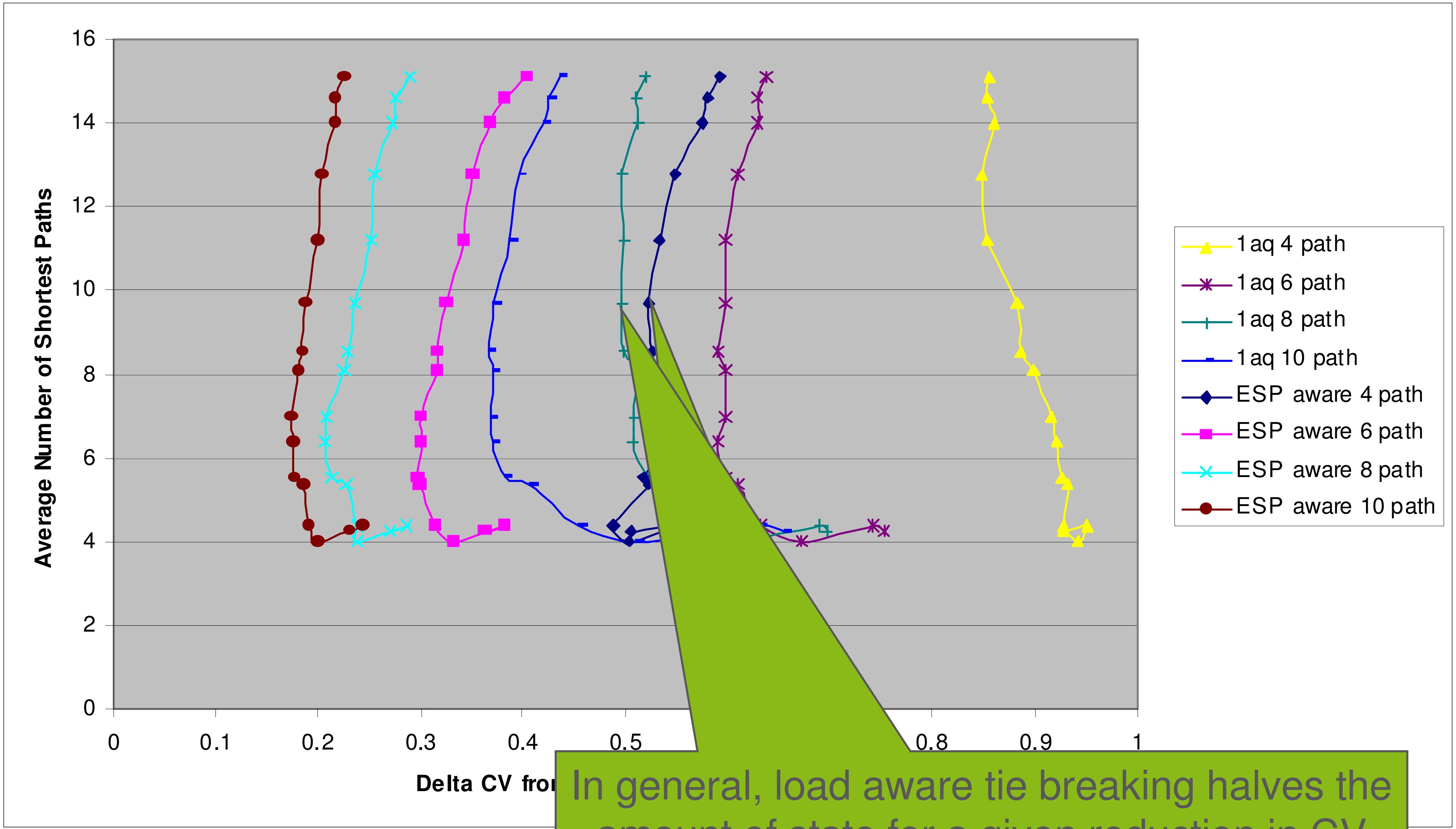
COMPARISON OF ESP AWARE VS CURRENT .1AQ (ER100)



COMPARISON OF ESP AWARE VS CURRENT .1AQ (ER100)



Comparison of ESP aware vs current .1aq (ER100)



In general, load aware tie breaking halves the amount of state for a given reduction in CV

ERICSSON