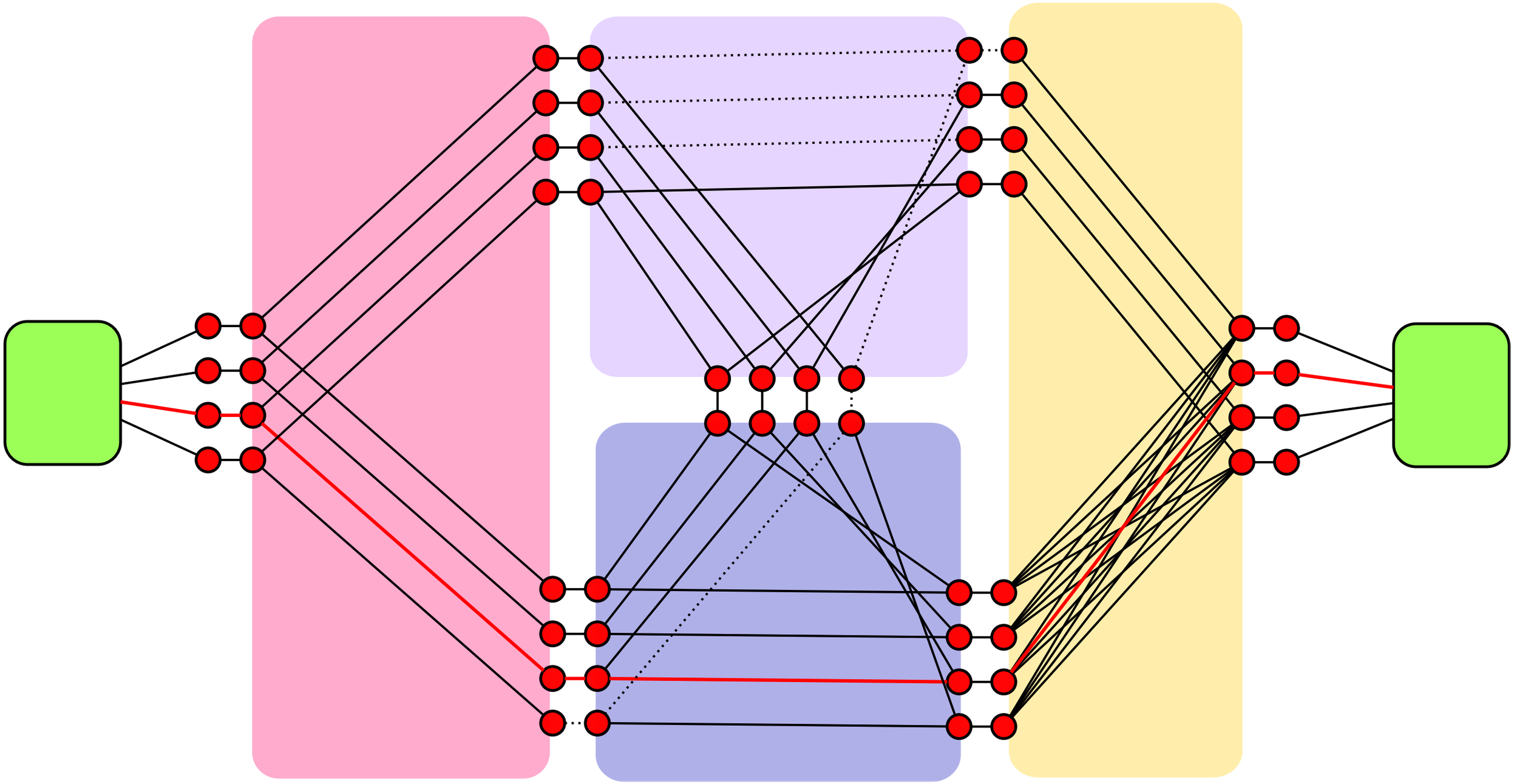
Practical Pathfinding



Paul Boven, Fedde Bloemhof (JIVE) for NSI-WG,OGF.

Theoretical approach:

- Assume we have a (static) topology available
- Perform Dijkstra's to find shortest path (resource usage optimalization)
- Try to reserve shortest path through NSI
- On failure, excise failed link from topology, and rerun Dijkstra's.

Practical problems:

- It takes 1.5s (SURFnet) or 0.7s (NORDUnet) to process an NSI reservation (fail or succes)

- The most likely failure is lack of permissions or lack of bandwidth, not lack of ‘label space’.

- We must however exhaustively search the (huge) label space before attepting a ‘longer’ path.

- Suggestion: return reason for reservationFailed
- Alternative: find path first, then find labels to use.

It gets worse:

- With VLAN retagging, the number of label-combinations to test for a path will be N^2 ... N^X This can cause a huge number or try-reservations.

- We need to hold reservations for some legs of a path while trying to find a complete path, wasting capacity.

- Race conditions occur when users do simultaneous pathfinding, resulting in non-optimal routes and possible failures.

Policy issues:

- Try the shortest/cheapest path first (resource usage)
- No width-first path-trying (reserving more links than you will actually use)
- Minimize the number of NSI transactions
- A special state for ‘pathfinding reservation’, indicating a retry might be useful ?

Suggested actions:

- Add length (km) as weight for pathfinding
- Return the cause of an error to speed up path-trying
- Alternatively, reserve path first, then find labels
- Mark reservations as ‘path-trying’ or ‘committed’
- Speed up response by NRMs
- Set policy for NSI load caused by path-trying.