
Multi-domain MPLS-TE

Latest developments in techniques for
computing inter-area and inter-domain
paths for traffic engineered MPLS

Adrian Farrel
CTO

Aria Networks Limited
adrian.farrel@aria-networks.com

Future-Net 2007





Agenda

- ➊ MPLS-TE Background
- ➋ What are Domains and Why Cross Them?
- ➌ Techniques for End-to-end Connectivity
- ➍ The Path Computation Element (PCE)
- ➎ Per-Domain Path Computation
- ➏ Crankback Routing
- ➐ TE Aggregation is bad!
- ➑ Backwards Recursive Path Computation
- ➒ Advanced Issues



MPLS-TE Background

- ④ MPLS-TE used to build “pipes”
 - ④ Direct traffic away from shortest paths
 - ④ Make best use of network resources
 - ④ Group traffic for common treatment
 - ④ Pseudowires, L3VPNs, scalability
 - ④ Quality guarantees through resource reservation
 - ④ Network repair and protection
 - ④ Fast Reroute (FRR)
 - ④ End-to-end protection
- ④ Signalled using RSVP-TE
- ④ Traffic Engineering Database (TED)
 - ④ Built from information distributed by the routing protocols
 - ④ Used to compute end-to-end paths



Network Domains

“A domain is considered to be any collection of network elements within a common sphere of address management or path computational responsibility.” - RFC 4726

- IGP areas
- Autonomous Systems
- Network layers
- Client/server networks

• Why cross domains?

- Because source and destination are not in the same domain!
 - Multi-area and multi-AS networks, virtual POP, etc.
- Because one domain provides connectivity for another domain
 - Client/server, multi-layer, VPN, etc.

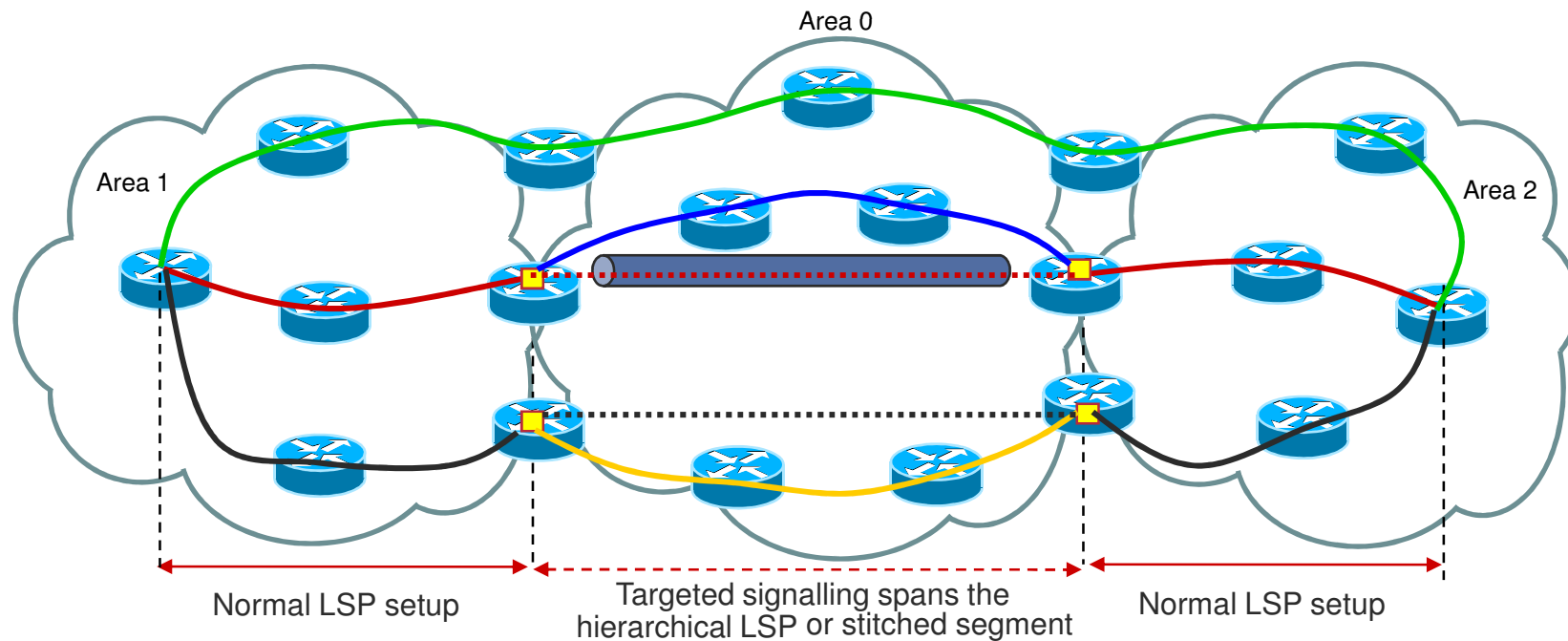
• How do we do it now?

- Manual stitching at domain boundaries
- Tunnel termination and reclassification of traffic at domain boundaries
- Careful off-line planning and management (e.g., FRR at domain borders)



Techniques for End-to-End Connectivity

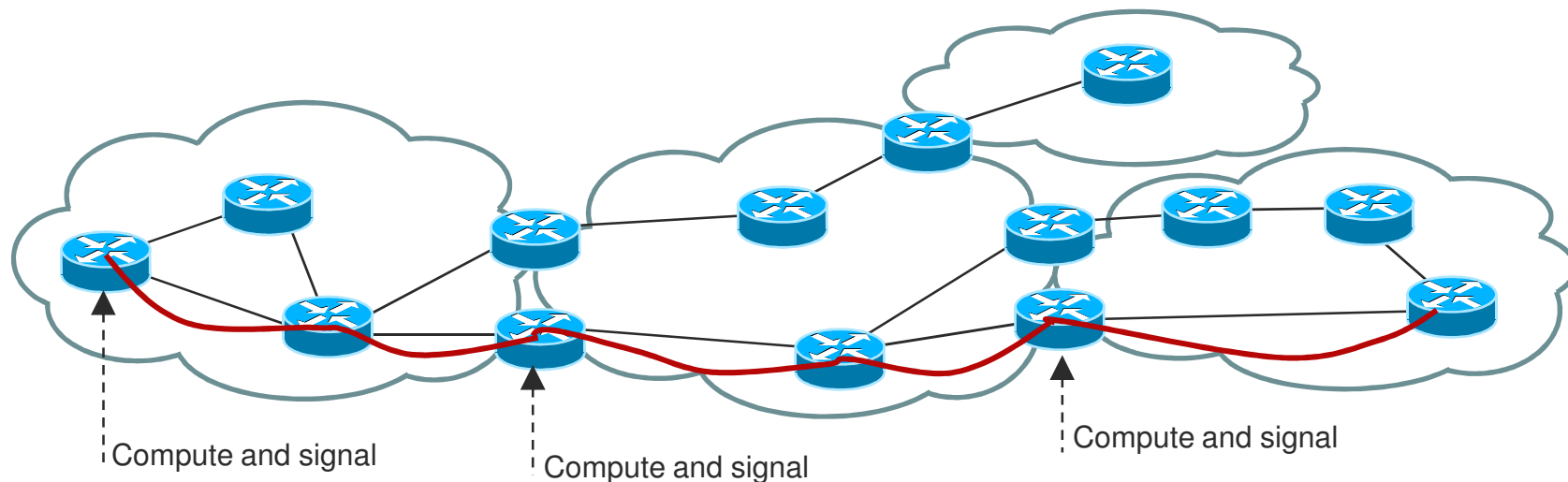
- Three techniques: contiguous, hierarchical, or stitched
- Trade-offs
 - Conceptual simplicity
 - Administrative boundaries
 - Data plane simplicity
 - Reoptimisation and protection
- Unanswered issues
 - How to compute end-to-end paths
 - How to select domain border nodes





Per-Domain Path Computation

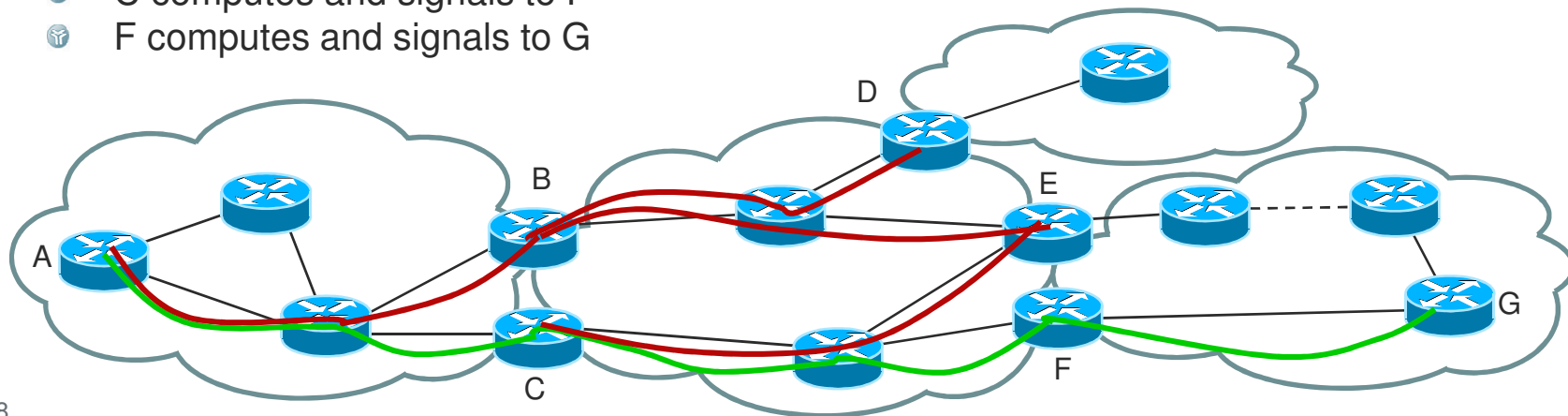
- Computational responsibility rests with domain entry point
- Path is computed across domain (or to destination)
- Works for contiguous, hierarchical, or stitched LSPs
- Which domain exit to choose for connectivity?
 - Follow IP routing? First approximation in IP/MPLS networks
 - Sequence of domains may be “known”
- Which domain exit to choose for optimality?





Crankback Routing

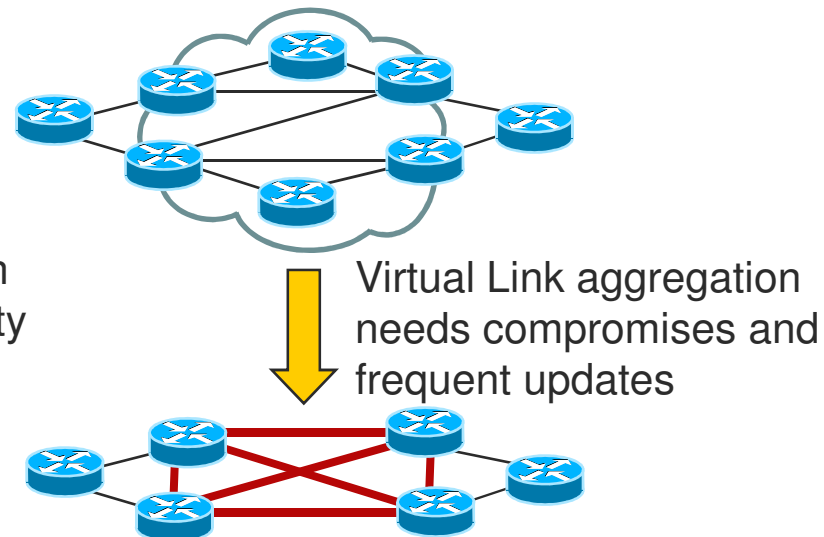
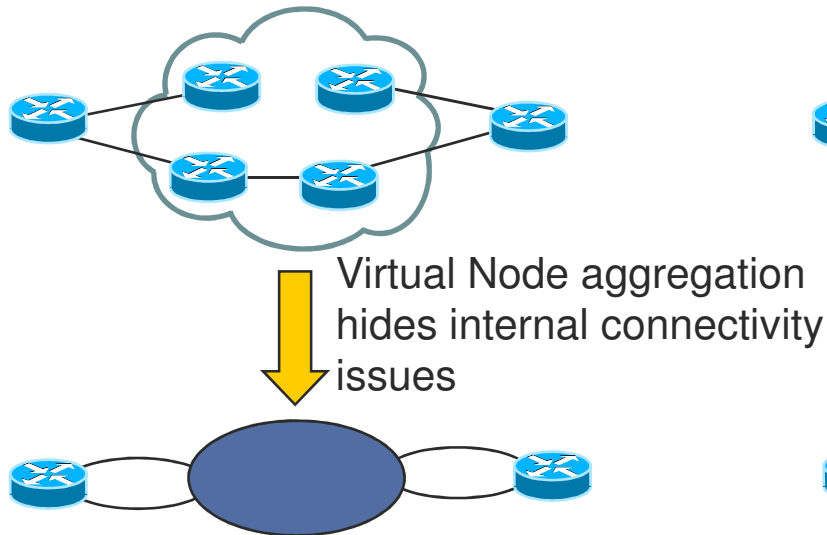
- ⊕ A cure for connectivity, but not for optimality
- ⊕ “Connectivity” means TE connectivity
 - ⊕ May have IP connectivity, but insufficient resources
- ⊕ May be painfully slow! “Informed random walk with wasted signalling”
 - ⊕ A computes and signals to B
 - ⊕ B computes and signals to D
 - ⊕ D fails to compute and reports failure to B
 - ⊕ B computes and signals to E
 - ⊕ E computes to G, but no resources. Reports failure to B
 - ⊕ B reports failure to A
 - ⊕ A computes and signals to C
 - ⊕ C computes and signals to E (can be avoided if E’s previous report is passed around)
 - ⊕ E computes to G, but no resources. Reports failure to C
 - ⊕ C computes and signals to F
 - ⊕ F computes and signals to G





TE Aggregation is Not a Solution!

- The solution is “full TE visibility” but this does not scale
- TE aggregation looks very promising
 - Provide enough information to compute, but still scale
 - But aggregation reduces available information so optimality is in doubt
 - May hide connectivity issues
 - May cause confusing aggregation of information
 - May need frequent updates as internal information changes
- TE reachability also sounds good
 - Just provide information about which destinations can be reached
 - What does “reachability” actually mean?



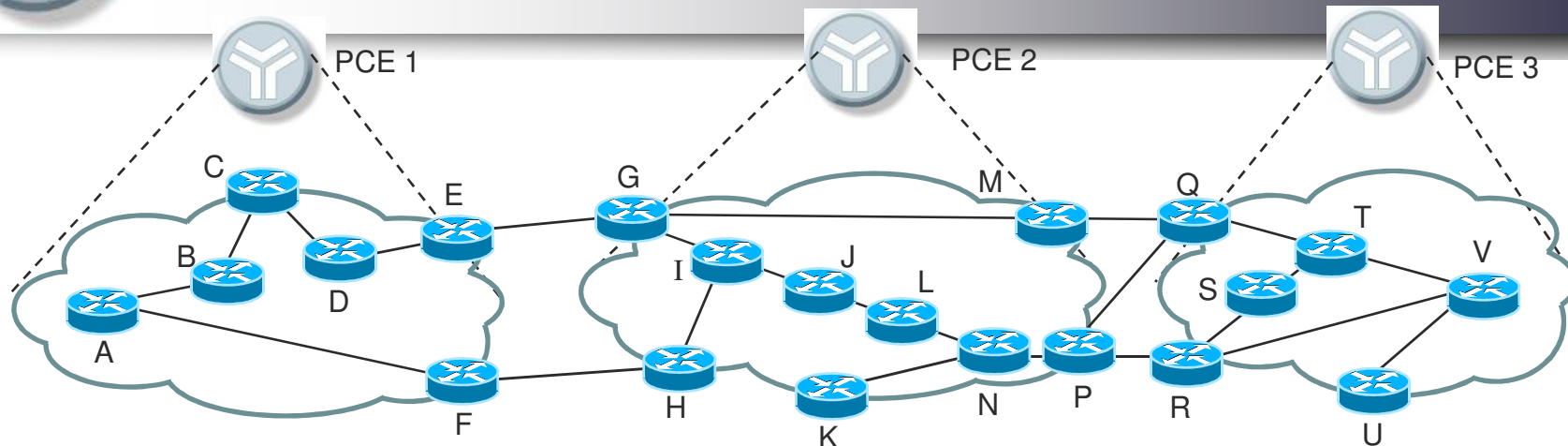


Backward Recursive Path Computation

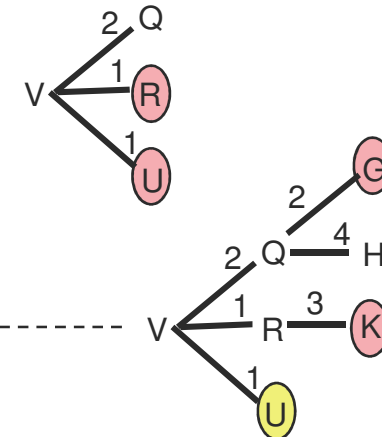
- Ⓜ PCE cooperation
 - Ⓜ Can achieve optimality without full visibility
 - Ⓜ “Crankback at computation time”
- Ⓜ Backward Recursive Path Computation is one mechanism
 - Ⓜ Assumes each PCE can compute any path across a domain
 - Ⓜ Assumes each PCE knows a PCE for the neighbouring domains
 - Ⓜ Assumes destination domain is known
- Ⓜ Start at the destination domain
 - Ⓜ Compute optimal path from each entry point
 - Ⓜ Pass the set of paths to the neighbouring PCEs
- Ⓜ At each PCE in turn
 - Ⓜ Compute the optimal paths from each entry point to each exit point
 - Ⓜ Build a tree of potential paths rooted at the destination
 - Ⓜ Prune out branches where there is no/inadequate reachability
- Ⓜ If the sequence of domains is “known” the procedure is neater



BRPC Example



- ☉ PCE 3 considers:
 - ☉ QTV cost 2; QTSRV cost 4
 - ☉ RSTV cost 3; RV cost 1
 - ☉ UV cost 1
- ☉ PCE 3 supplies PCE 2 with the tree
- ☉ PCE 2 considers
 - ☉ GMQ..V cost 4; GIJLNPR..V cost 7; GIJLN PQ..V cost 8
 - ☉ HIJLNPR..V cost 7; HIGMQ..V cost 6; HIJLN PQ..V cost 8
 - ☉ KNPR..V cost 4; KNPQ..V cost 5; KNLJIGMQ..V cost 9
- ☉ PCE 2 supplies PCE 1 with the tree
- ☉ PCE 1 considers
 - ☉ ABCDEG..V cost 9
 - ☉ AFH..V cost 8
- ☉ PCE 1 selects AFHIGMQTV cost 8





Advanced Computation Issues

- ④ Inter-domain TE link information
 - ④ For example, inter-AS links
 - ④ Needs to be part of the information within a domain
- ④ Path optimisation
 - ④ Avoidance of “traps”
 - ④ Trade-off of conflicting constraints
- ④ FRR consideration during initial LSP placement
- ④ Path diversity
 - ④ End-to-end protection, load sharing, etc.
 - ④ Link, node, domain, SRLG diversity
 - ④ Avoidance of “traps”
- ④ Reoptimisation
 - ④ End-to-end or per-domain
 - ④ “Shuffling” of deployed LSPs to free up stranded resources
 - ④ May require migration strategies
- ④ Different service types
 - ④ Point-to-multipoint



The Future of Path Computation

⊗ Holistic Path Computation

- ⊗ Solving the whole network is hard
 - ⊗ Balance conflicting constraints for different services
 - ⊗ Consider all services at once to avoid trap conditions
 - ⊗ Huge networks with thousands of services
 - ⊗ Needs to be adaptive to changes in topology and services
 - ⊗ Must be flexible to mixes of service types (P2P, P2MP, etc.)
- ⊗ Necessary for full optimisation, but can it be achieved in real time?

⊗ Non-heuristic processes

- ⊗ Conventional algorithms are deterministic and tuned to specific topologies and service types
- ⊗ Non-heuristic processes can assess the whole network and all demands at once
 - ⊗ Can handle all topologies
 - ⊗ Can manage different service types
 - ⊗ Can trade-off conflicting constraints
 - ⊗ May produce a different, but correct solution each time

⊗ Highly sophisticated planning and modelling tools

- ⊗ Multi-function
 - ⊗ Network failure analysis
 - ⊗ Capacity planning
 - ⊗ Rapid turn-around of network experiments
 - ⊗ Network re-optimisation
- ⊗ Integrated planning and activation (NMS and PCE)
- ⊗ On-line optimisation and reoptimisation
 - ⊗ Smart PCE
 - ⊗ Dynamic reconfiguration of networks with configured parameters, thresholds, and cost/risk/benefit analysis

⊗ Aria Networks Ltd. <http://www.aria-networks.com>



Standardisation Status and References

- ☉ RFC 4216: MPLS Inter-Autonomous System (AS) Traffic Engineering (TE) Requirements
- ☉ RFC 4105: Requirements for Inter-Area MPLS Traffic Engineering
- ☉ RFC 4726: A Framework for Inter-Domain Multiprotocol Label Switching Traffic Engineering
- ☉ RFC 4655: A Path Computation Element (PCE)-Based Architecture
- ☉ RFC 4206: Label Switched Paths (LSP) Hierarchy with Generalized Multi-Protocol Label Switching (GMPLS) Traffic Engineering (TE)
- ☉ draft-ietf-ccamp-lsp-stitching: LSP Stitching with Generalized MPLS TE (work in progress)
- ☉ draft-ietf-ccamp-inter-domain-pd-path-comp: A Per-domain path computation method for establishing Inter-domain Traffic Engineering (TE) Label Switched Paths (LSPs) (work in progress)
- ☉ draft-ietf-pce-brpc: A Backward Recursive PCE-based Computation (BRPC) procedure to compute shortest inter-domain Traffic Engineering Label Switched Paths (work in progress)

Questions?

adrian.farrel@aria-networks.com



Aria Networks