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

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

© 2006-2007. The Copyright in this presentation belongs to Aria Networks Ltd.
MPLS-TE Background

MPLS-TE used to build “pipes”
- Direct traffic away from shortest paths
  - Make best use of network resources
- Group traffic for common treatment
  - Pseudowires, L3VPNs, scalability
- Quality guarantees through resource reservation
- Network repair and protection
  - Fast Reroute (FRR)
  - End-to-end protection

Signalled using RSVP-TE

Traffic Engineering Database (TED)
- Built from information distributed by the routing protocols
- 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

Path Computation Element

"An entity (component, application, or network node) that is capable of computing a network path or route based on a network graph and applying computational constraints" - RFC 4655

- What’s new?
  - Nothing!
  - A formalisation of the functional architecture
  - The ability to perform path computation as a (remote) service
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?

![Virtual Node aggregation hides internal connectivity issues](image1)

![Virtual Link aggregation needs compromises and frequent updates](image2)

**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; GIJLNPQ...V cost 8
- HIJLNPR...V cost 7; HIGMQ...V cost 6; HIJLNPR...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 AFHGMQT cost 8

Advanced Computation Issues

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


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
- 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