L2TPEXT Presentation

Tom Mistretta

Mahesh Kelkar

Paul Howard

Topics

- Failover (draft-ietf-l2tpext-failover-04.txt)
- RADIUS-EXT / info-msg / SESINFO
- Tunnel-switching

Failover

- Long understood problem
- Each L2TP peer has configured and dynamic (stateful) information
- Stateful information changes with the processing of protocol packets
  - Control connections and sessions
  - Sequence numbers for control channel and data channels

Model

- Each active endpoint has backup capabilities
  - Redundant hardware, parallel system, fast-reboot, etc.
  - View is that backup can mirror some state but this is performance-sensitive and lossy in nature
- Upon a failover, stateful information needs to be reconciled between failed endpoint and non-failed endpoint

Choices for recovering state

- Complete synchronization between active and backup
  - Adds significant complexity and performance impact. Unrealistic in most scenarios, especially those of high scale.
- Re-signal all control connections and sessions
  - Possible for low-scale scenarios
  - Subject of galtzur draft

draft-ietf-l2tpext-failover

- L2TP protocol extensions that allow for re-synchronization of stateful information
- Three phases:
  - Pre-failover
  - Failover recovery
  - Session state synchronization

Protocol Overview

- Pre-failover during control channel setup
  - Specify failover capabilities (control / data) and parameters (timeout of control channel)
- Failover recovery
  - Failed node establishes a recovery tunnel for each failed control connection
    - Non-failed node can reject if out-of-sync
  - Resynchronize sequence numbers
- Session state recovery
  - Reconcile any differences between sessions on control connection through use of session query/response messages

Status

- draft-ietf-l2tpext-failover in good shape, ready for working group last call
- Concern that there are two drafts addressing this problem (failover and galtzur drafts)
- Our view is that they both are valid for different circumstances
  - Failover draft good for high-scale scenarios but requires more code changes
  - Galtzur draft applicable for low-scale scenarios and has the advantage of re-using existing L2TP protocols

Moving Forward

1. Attempt to merge drafts
2. Choose one draft over the other
3. Allow both to move forward independently

- Due to the drafts distinctly different application areas we propose that L2TPEXT allow both drafts to move forward independently
  - Failover ready for last call

- **Discussion and comments?**

RADIUS-EXT, Infomsg, SESINFO

- History
  - No standard way of propagating NAS-port information between LAC and LNS and out to RADIUS
    - Slot/port/vpi/vci/vlan/etc.
    - Critical for authentication, billing, etc
  - Debug messages
  - Tunnel-switch behavior undefined

History, cont’d

- SESINFO draft addressed NAS-port to some extent (extension of L2TP physical channel ID, node-name list, etc) however didn’t go far enough (i.e., no standard encoding)
- Infomsg talked about debug/information strings but not about NAS-port info

RADIUS-EXT

- Debug/information strings
- NAS-port definitions
- Integration with RADIUS
- Tunnel-switch behavior

Moving forward

- RADIUS-EXT only draft
  - Forget about infomsg and sesinfo
- Move tunnel switch behavior to newly-resurrected tunnel-switching draft (more on this later).
- New version of RADIUS-EXT
  - Expires in December
- Go to last call on RADIUS-EXT

- **Discussion and comments?**

Tunnel-Switching

- Common application
  - Intermediary between LAC and LNS performs control channel aggregation, implements policy, etc.
- Independent, proprietary implementations
  - Interoperability woes
  - Some technical problems
    - Attribute propagation
    - Loop detection (mostly academic)
    - Congestion control
- Older draft long expired

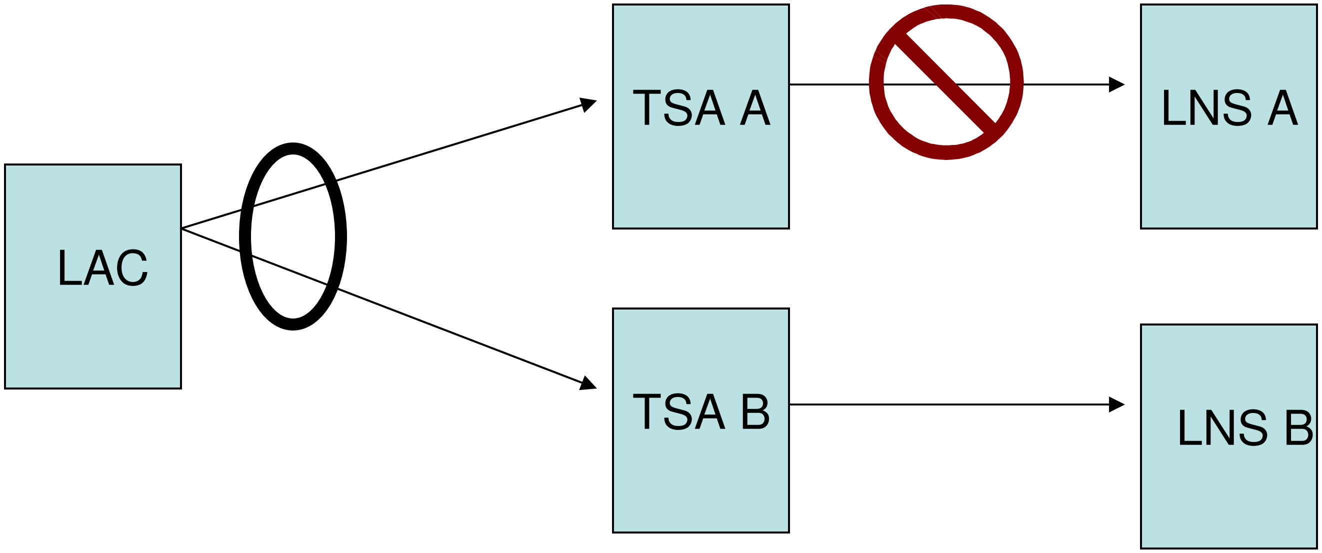
What to do

- Aforementioned problems still exist
- Would like to re-open tunnel-switching draft
  - L2TPv3 updates
  - Move AVP propagation definitions from RADIUS-EXT
  - Re-visit congestion problem
  - Loop detection?
  - Deliver new draft to mailing list by early next year
    - Plenty of time before next meeting
  - Resolve issues and move to last call quickly

- **Discussion and comments?**

Brief Focus on Congestion Problem

- Problem: congestion on Tunnel Switch Aggregator (TSA) not propagated back to LAC



Simple Proposal

- Existing draft has a complex solution that specifies capacities and meta-information (“profile” or “domain”) back to LAC
- We believe this problem can be solved by simply defining new disconnect codes
- Local policy on LAC runs the rest

- **Discussion and comments?**