Auto-configuring Aggregate Links

Objectives
&
A functional partition

Mick Seaman
mick_seaman@3com.com

Auto-configuring Aggregate Links

Presentation Goals

- Share some ideas
- Introduce some objectives

Presentation Non-goals

- To claim unique/best perspective

Overview

- Auto-configuration Philosophy
- A functional partitioning
- Objectives to take away

Auto-configuration Philosophy (1)

- There is no magic t=0 for LAN switches
  - Links can be added
  - ……………….. removed
  - ……………….. aggregated
  - ……………….. disaggregated
  - ……………….. fail
  - ………...and recover
  - without switches being powered down, reinitialized ……… just as part of normal network operation

Auto-configuration Philosophy (2)

- If it doesn’t auto-configure we may as well use routing with equal cost load sharing
  - – switches are distinctly different because of ease-of-administration
  - – different economic approach is required if two competing solutions are to survive
  - – short-term alternatives are not worth the standardization time and hassle

Auto-configuration Philosophy (3)

- Swift and Sure

  Major obstacles ….

  - excessively chatty protocols, particularly after major network events (neither swift or sure)
  - not using low-level indicators (not swift)
  - relying on low level indicators (not sure)
  - not continuous (when am I finally sure?)

Auto-configuring Aggregate Links

- Aggregate Link Identify
  - identifies likely candidates for an aggregate link
- Aggregate Link Verify, Initiate, Maintain
  - verifies all links connect the same two systems
  - synchronize start use of links as an aggregate
  - add and remove links to/from existing aggregate
- Link Failure Detect
  - identify failed links rapidly

Aggregate Link Identification (1)

- Potentially an additional repetitive message on every link in the network
  - many links may not be aggregated - pure waste
  - existing protocols already identify opportunities
    - see later example
  - media specific opportunities?
  - manual “hints” for single ended configuration
  - brute force as a fall back
- Keep this separate
  - even if available in the aggregation tool kit

Aggregate Link Identification (2)

- Links identified as candidates for aggregation within 1 minute of adding or physically rewiring link

Verify, Initiate, Maintain (1)

- Verify connectivity (two systems, point-to-point)
- Verify both systems can aggregate the links
- Initiate use of links as aggregate
- Signal ‘single-ended’ failure/out of use of link
- Remove link from aggregate
- Add link to aggregate
- Guard against missing physical link indicator and one-way connectivity

Verify, Initiate, Maintain

- Verification, Initiate, and Add not time critical
- Eliminate duplication risk on initiation and addition
- Minimize loss risk on initiation and planned Removal
- Minimize loss window for single ended failure
  - < 1 second, allow < 50 milliseconds
  - < any periodic message generation

Failure Detection

Minimize loss window, within 50 milliseconds

Link Identification : Example

- Spanning Tree identifies potential aggregates
- … bridge ports with same Designated Port
- … Root Port and/or Alternates
- … not limited to links “in the Spanning Tree”
- … identified at one end of the link(s)

Conclusion : Set Objectives for ...

- No absolute requirement for additional protocol on links which will never be aggregated
- Continuous switch operation
- Timing, loss, duplication, message frequency for verify, initiate, add, remove, failure, failure detection
- Deterministic outcomes