Link Aggregation Setup and Maintenance

- **Initial Setup**
- **Dynamic Behavior**
- **Administrative Controls**
- **“Flush” Protocol**
- **Other Issues**

Initial Setup

- **Exchange “Hellos” to automatically configure link aggregation**
- **Converge within at most a very few seconds**
- **Reliably determine the aggregation possibilities**
- **Allow connection (without aggregation) to aggregation-unaware devices**
  - **Within reason, reliability and connectivity to unaware devices are more important than millisecond initial setup.**
- **Verify the correctness of preconfigured connections**
  - **This should fall out of the more general protocol; it is not worth doing two protocols.**

Initial Setup (2)

- **Accommodate differing hardware/software constraints on aggregation capabilities**
  - **Existing hardware or software may have arbitrary constraints on its ability to form aggregations.**
- **Minimize the cost of "yet another Hello" protocol**
  - **Use fast start-up after hardware connection is made, slow down Hellos if no response received.**
  - **Can slow down Hellos after aggregation is made; this is a trade-off against response to link failures not detected by hardware.**
- **This is not a general topology discovery protocol**
  - **There is overlap with topology discovery, but trying to address discovery problems will de-focus this effort.**

Initial Setup (3)

- **Detect uni-directional links**
  - **This capability falls out of the exchange of Hellos**
  - **This capability may require administration; it conflicts with the desire to connect to devices unaware of link aggregation or which are unidirectional in nature (e.g. a “sniffer” port).**
  - **Aggregation does not come up if no links are bi-directional.**
  - **Uni-directional link is not allowed to join an aggregation.**

Dynamic Behavior

- **Use hardware indicators to detect failed links**
  - **Where available, the hardware provides the quickest possible detection of a failed link.**
  - **Link failures must be detected quickly, as frames will be black-holed until the distributor detects the failure and redirects the flow(s) on that physical link.**
- **Use protocol packet exchange as a backup to detect failed links not indicated by hardware**
  - **Some insurance against hubs and/or failed hardware is needed.**
  - **Using the timing of Hellos after the aggregation has been formed, one can trade responsiveness for bandwidth wastage by Hellos.**

Dynamic Behavior (2)

- **Use same protocol as initial setup to allow links to join an existing aggregation**
  - **Reconnection is less time-critical than failure detection.**
  - **Reliability of connection verification is just as important as at initial setup time.**
- **The joining and leaving of links can be indicated by speed changes of the aggregate**

Administrative Controls

- **Allow administrative constraints on connectivity**
  - **The administrator must be able to limit which physical links may aggregate together, and to which mux ports they may aggregate.**
  - **We may decide to *not* define an administrator's ability to specify limits as to which devices an aggregation may connect; this would be a new capability not available to non-aggregated links, and quite separable from link aggregation.**
- **Control whether or not link joins/leaves are propagated as link speed changes to higher layers**
  - **In order to minimize spanning tree disruptions, we may want to inhibit changing the link speed on join/leave.**

Administrative Controls (2)

- **Accept/deny non-aggregated connections with devices which do not run the trunk establishment protocol.**
- **Report the creation/destruction of aggregates as mux interface up/down traps.**
- **Separate the manual control of aggregation from the control of the automatic protocol.**
  - **This allows the implementation of aggregation with or without automatic control.**

Other Issues

- **We may wish to detect multi-point connections, though not to aggregate them**
- **There are classes of existing devices which cannot meet the strict requirements for the distributor or collector as defined, but which we may wish to allow some degree of meaningful participation**
  - **E.g. a hardware bridge which cannot ignore the physical source of a packet is perfectly able to receive packets distributed solely on the basis of source MAC address.**

“Flush” Protocol

- **Switch “A” connects to switch “B” with three aggregated links 1-3; a fourth link 4 joins the group.**
- **Presumably, some flows will switch from, say, link 1 to link 4.**
- **Switching a flow may cause out-of-order delivery of packets in that flow.**
- **We can provide an optional provision to avoid out-of-order delivery in this case:**
  - **“A” sends a “flush” message down link 1.**
  - **“B” (in software) returns a “flush” reply to “A”.**
  - **Until “A” receives the reply, it holds or discards all frames in flows that are switching from 1 to 4.**

“Flush” Protocol (2)

- **Sending “flushes” should be optional**
  - **Not all distribution algorithms require it.**
  - **Some implementations may elect to suffer the consequences of out-of-order delivery.**
- **Sending “flushes” should be possible**
  - **Many good distribution algorithms require it.**
  - **It is probably not acceptable to standardize something that prohibits meeting this basic bridge requirement.**
- **Unfortunately, the echoing of “flush” packets must be mandatory if “flushes” are allowed**
  - **“A” cannot depend on using “flushes” to ensure ordering if “B” does not echo them back to “A”.**