Link Aggregation Control Protocol

Presentation to the Link Aggregation Study Group, April 1998

Tony Jeffree

Basic assumptions/objectives

- If aggregation is possible, it will happen automatically
- Determinism
- Rapid convergence
- Base-line is non-aggregated behaviour
- Low risk of mis-configuration
- Low risk of duplication or misordering

Identifying link characteristics

- Many characteristics that contribute
  - Standardised in .3: Link speed, duplex/non-duplex…etc
  - Other characteristics…e.g., administrative, non-standardised
- -> concept of a *Capability* identifier
- *Capability Group:* All Links in a system that share the same capability
- Null capability: This link is not capabale
- Simplifying assumption: A link needs only one Capability ID

Identifying Link Aggregation Groups

- System ID plus Capability provides a global identifier for a Capability Group
- The set of links in an aggregation are identified by concatenating the Capability Group identifiers at each end of the link
- Hence, for Systems S and T, who use C and D as the Capability ID for a set of aggregated links, the LAG ID would be {SC, TD}…(which is the same identifier as {TD, SC})
-

Detecting Aggregation possibility

- Aggregation possibility can be detected simply by exchanging global Capability Group Ids across a link; each system can then see whether any other Links exist with the same {SC,TD} value.
- If other links in a system exist with the same {SC, TD} then they can all be added to the same Link Aggregation Group
- Simplifying asumption: no limit on aggregation size - allocate more capabilities if it is necessary to impose such a limit.

Prevention of Duplication/Reordering

- Don’t Distribute on a Link until you know that the other end is Collecting
- Stop Distribution/Collection on a Link prior to moving it to a new group
- Need to “flush” other links if Conversations are re-allocated as a result of adding/removing links (see Paul Congdon’s comment - this needs to be added to the document)

Protocol basics

- If in doubt, say everything twice
- Communicate state, not commands
- Need to Know/Need to Tell concept
  - Need to Know if not confident of your information at hand
  - Need to Tell if local state has changed
- Tell the other party what you know. When you are agreed - aggregate.

Information communicated

- Your SC;
- What you think their TD is;
- Whether you are Collecting on that link;
- Whether you Need to Know;
- Your Link ID (may be useful for debugging)
- (Flush ID - not yet added to the document)

Basic protocol operation

- NK becomes asserted if Refresh timer fires
- Receiving inconsistent information, or information with NK asserted, causes NT to be asserted
- Local state change causes NT to be asserted
- NK and NT cleared once 2 messages sent
- NK cleared on receipt of a response
-
  –

Flush protocol operation something like...

- Flush ID plus NK sent (along with normal message content). Sender chooses ID value.
- Recipient’s NT is asserted by receipt of NK; Flush ID saved by recipient & sent in subsequent messages till NK/NT no longer asserted.
- Note: Does not fix the case of a link failing.
-

Link States w.r.t. Aggregation (1): State variables

- D: Capability of this system w.r.t. this link (0=Null)
- C: Capability of other system w.r.t. this link (0=Null or don’t know)
- K/0: Collecting/not collecting
- J/0: Distributing/not distributing

Link States w.r.t. Aggregation (2): Valid states are...

- 0000: Neither system believes that this link can aggregate
- D000: Local system believes the link can aggregate, remote does not
- 0C00: Remote system believes the link can aggregate, local does not
- DC00: Both systems can aggregate, collection/distribution yet to be started
- DCK0: Collection started
- DCKJ: Distribution started (implies remote collection is enabled)

Link States w.r.t. Aggregation (3):

- 0000, D000, 0C00 represent states in which the Link can only be operated as a normal, non-aggregated link, unless re-configuration of the link and/or its capabilities takes place
- Remaining states show various states of progress towards aggregation. Note that DC0J is an illegal state (Distributing on the link before Collection is enabled).

Progression towards aggregation

- Collection can be enabled as soon as agreement has been reached between the ends of the link
- Distribution is enabled only after:
  - It is known that the far end is Collecting; and
  - Any “flushing” required by near end re-configuration has completed.

“MUX” state

- Receive enabled if any of its Links are Collecting
- Transmit enabled if any of its Links are Distributing