802.1CB
Another Reset Proposal

Johannes Specht johannes.specht AT uni-due.de **Univ. of Duisburg-Essen**

Problems …

Initial situation in 802.1CB

- http://www.ieee802.org/1/files/public/docs2015/new-tsn-specht-cb-failure-modes-0521-v1.pdf shows a couple of failure modes, not covered by the underlying 802.1CB draft
- Excluding cut-through issues, the essence of all failure modes are different flavors of rouge packets on one of the redundant paths (e.g. by broken bridges)
- Proposed robust solution was „don‘t do reset“ – which the author believes is the most robust thing we can do for fail silent applications

Problem with this …

- Having no reset kills stream forever once they are down (if there‘s no management operation)
- While acceptable for fail silent applications (e.g. current automotive systems), no automatic reset can be an issue for other use cases
- Additional management protocols bloat the standard, need to be safe themselves, etc

Solutions

Solution Space

http://www.ieee802.org/1/files/public/docs2015/cb-nfinn-sequence-recovery-0715-v01.pdf summarizes variations and their issues of protocol-based solutions:

- Carried in the protocol, which makes interworking with other protocols (HSR, pseudowires) much more difficult;
- Separate best-effort packets, which means they may not follow the same path through the network as the Stream packets, so they may be out of order with the Stream, or not even arrive; or
- Separate in-stream packets, which means they must be accounted for in the Stream bandwidth calculation.

Current Solution

- A timer counting discarded packets. If enough packets were discarded in series, a reset is performed.
- Given that only one of two redundant paths is allowed to fail anyway, the other (fault free) path would sent enough non-discarded packets, thus preventing timeout
- Drawbacks (cmp. http://www.ieee802.org/1/files/public/docs2015/cb-nfinn-sequence-recovery-0715-v01.pdf):
  - Explicit: A reset cannot be performed at arbitrary times and arbitrary fast
  - Timer configuration
  - Implicit: We need a timer (state)

Content

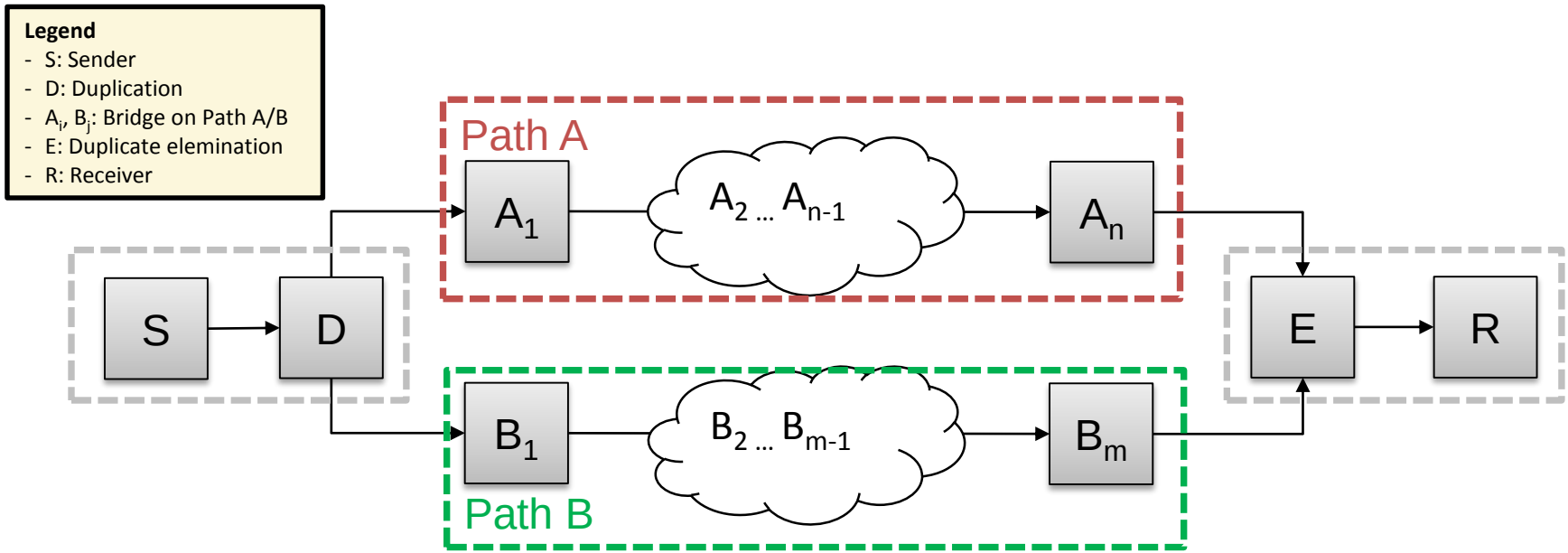
What is in this slide set?

… an alternative solution for discussion, very roughly described:

- Lives without an additional protocol/additional packet data (☺)
- No timer configuration needed
- Less constrained reset (… plenty of resets) when needed while a faulty path cannot issue reset
- Should have higher coverage w.r.t. fault scenarios
- Drawbacks: Also needs state, but probably less – more specific: *2+ceil(log$_2$(window size))* bit per path in the extended version (read on…); less to implement the basic idea

* Note: The herein presented proposal was not yet checked against all error cases nor does this slide set include all details packet sequence diagrams, etc.

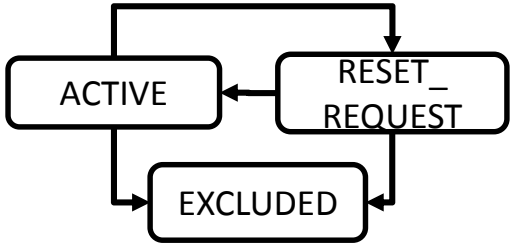
Proposal: Group Confirmed Reset



Brief Description of the Basic Idea

- If a reset is performed, it is done based on sequence number jumps observed by E
- ***all*** paths in the group (2 in the above example) have to confirm reset!
- E.g., if a babbling bridge on the faulty path emits bad sequence numbers interpreted as reset request, than no reset is executed – the fault free path does not request reset
- **In the extended version**: faulty paths are excluded from the decision, thus re-enabling reset for the remaining fault free path

Extended Version ...



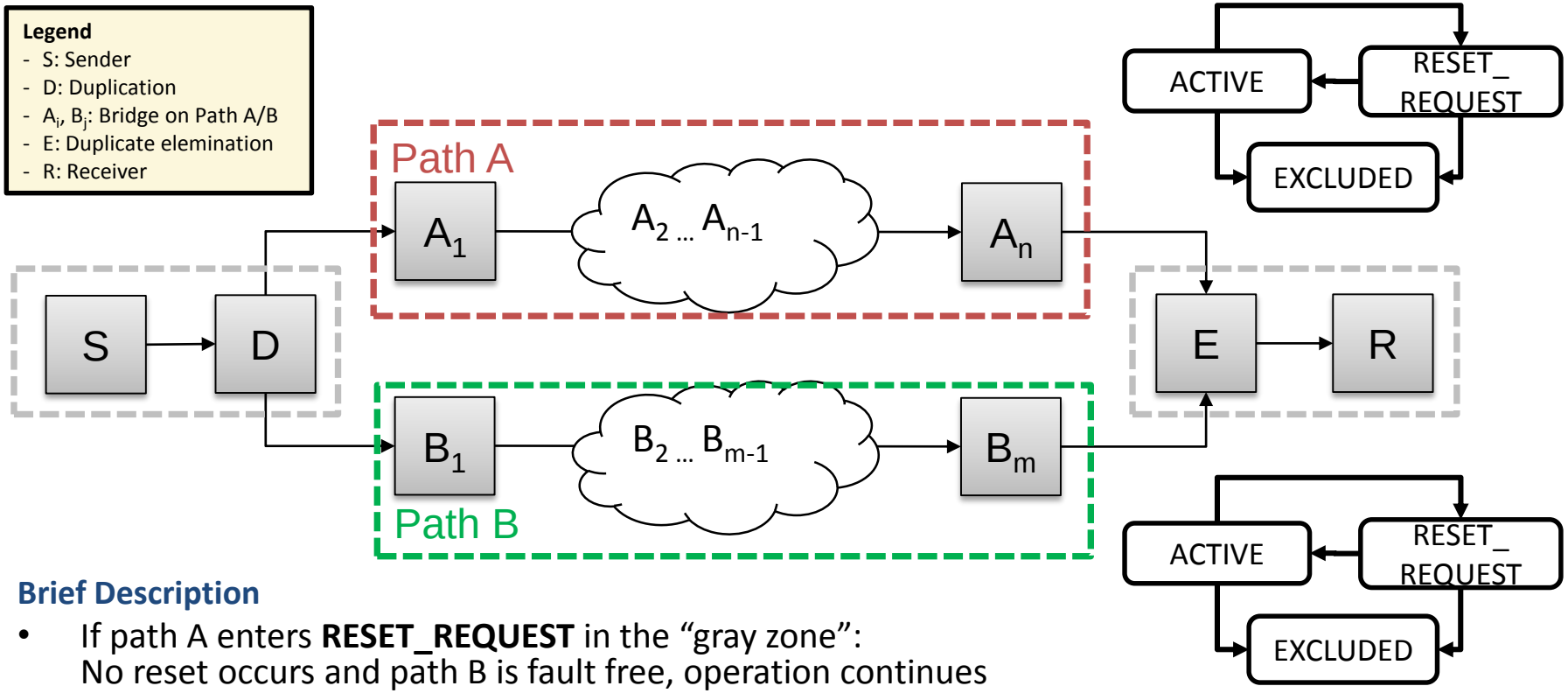
Description

- States per path:
  - **ACTIVE**: Initial state and entered from **RESET_REQUEST** after reset is executed. Packets are accepted or discarded (as-is operation based on the history window)
  - **RESET_REQUEST**: Entered from **ACTIVE** if a jump in the sequence number is observed. Packets are unconsidered and discarded*.
  - **EXCLUDED**: Packets are unconsidered and discarded, entered if no packets have been received on the path for the history window size packets - measured by the progress of other paths (that's the reason for *ceil(log_2(window size))*) - ***or*** because 802.1Qci complained about the path (stuck seq. number, etc.).
- Once a path is **EXCLUDED** there's no transition back and packets are discarded in this state. But there is at least one other path to rely on (although there may be reasonable safe criteria for transition back, but not discussed in this slide set)
- If *and only if* all paths are in **RESET_REQUEST** or **EXCLUDED** – reset is performed, i.e.:

  RESET_REQUEST → ACTIVE; **EXCLUDED → EXCLUDED** (!)

*Note: Once the last (slowest) path enters **RESET_REQUEST**, reset is performed and packets are accepted from all paths. All packets can be recovered, although out of order delivery may occur (but we have to deal with this anyway in case of failure).

Cases for Discussion



Brief Description

- If path A enters **RESET_REQUEST** in the "gray zone":
  No reset occurs and path B is fault free, operation continues
- If path A (faulty) is stuck at a sequence number:
  Qci triggers transition to EXCLUDED in path A FSM. Consecutive resets by (fault free) path B are permitted … again.
- If D goes temporarily down (lost packets) or S restarts the sequence generation:
  Reset is performed once the seq. number jump propagated to E on both paths
- If S or D emit bad sequence numbers or path A and path B are faulty:
  CB is the wrong tool anyway

Thank you for your Attention!

Questions, Opinions, Ideas?

Johannes Specht

Dipl.-Inform. (FH)

Dependability of Computing Systems
Institute for Computer Science and
Business Information Systems (ICB)
Faculty of Economics and
Business Administration
University of Duisburg-Essen

Schuetzenbahn 70
Room SH 502
45127 Essen
GERMANY
T +49 (0)201 183-3914
F +49 (0)201 183-4573

Johannes.Specht@uni-due.de
http://dc.uni-due.de