Quick Failover Algorithm in SCTP

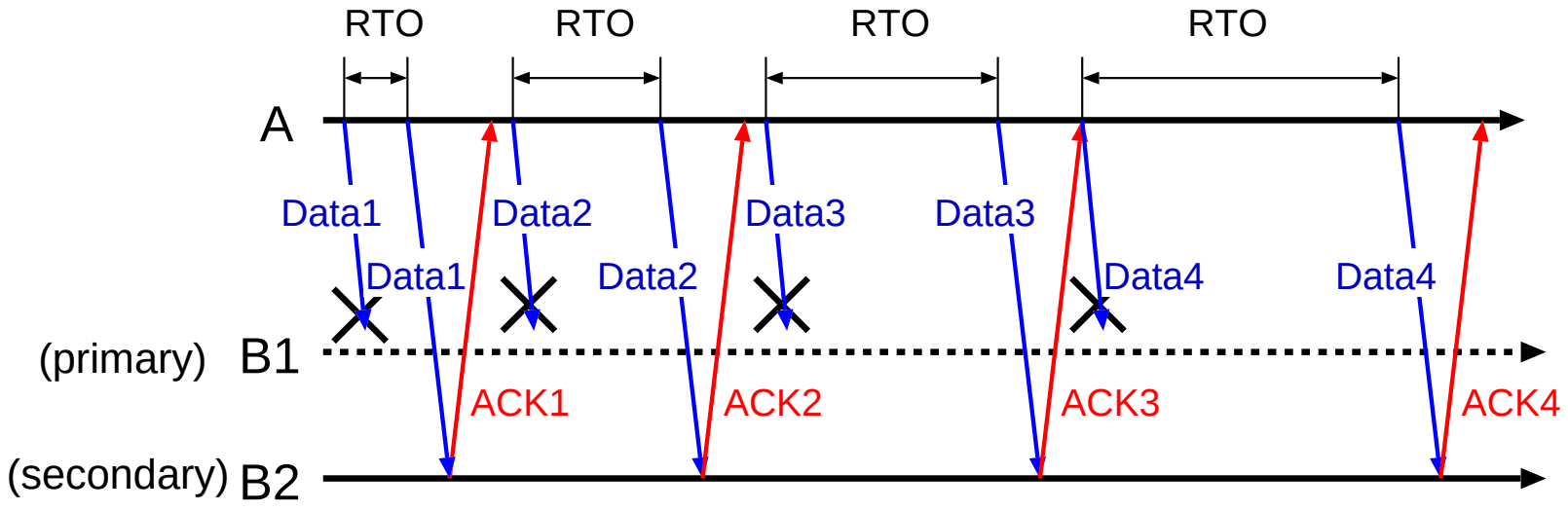
Yoshifumi Nishida, WIDE Project
Preethi Natarajan, CISCO systems

Motivations

- Multihoming is a major feature of SCTP
  - SCTP can migrate to secondary paths when primary path becomes unavailable
- But, SCTP needs 30-60 secs to failover in standard settings
- Describing remedies for this issue makes SCTP more useful and attractive

Issues in SCTP Failover

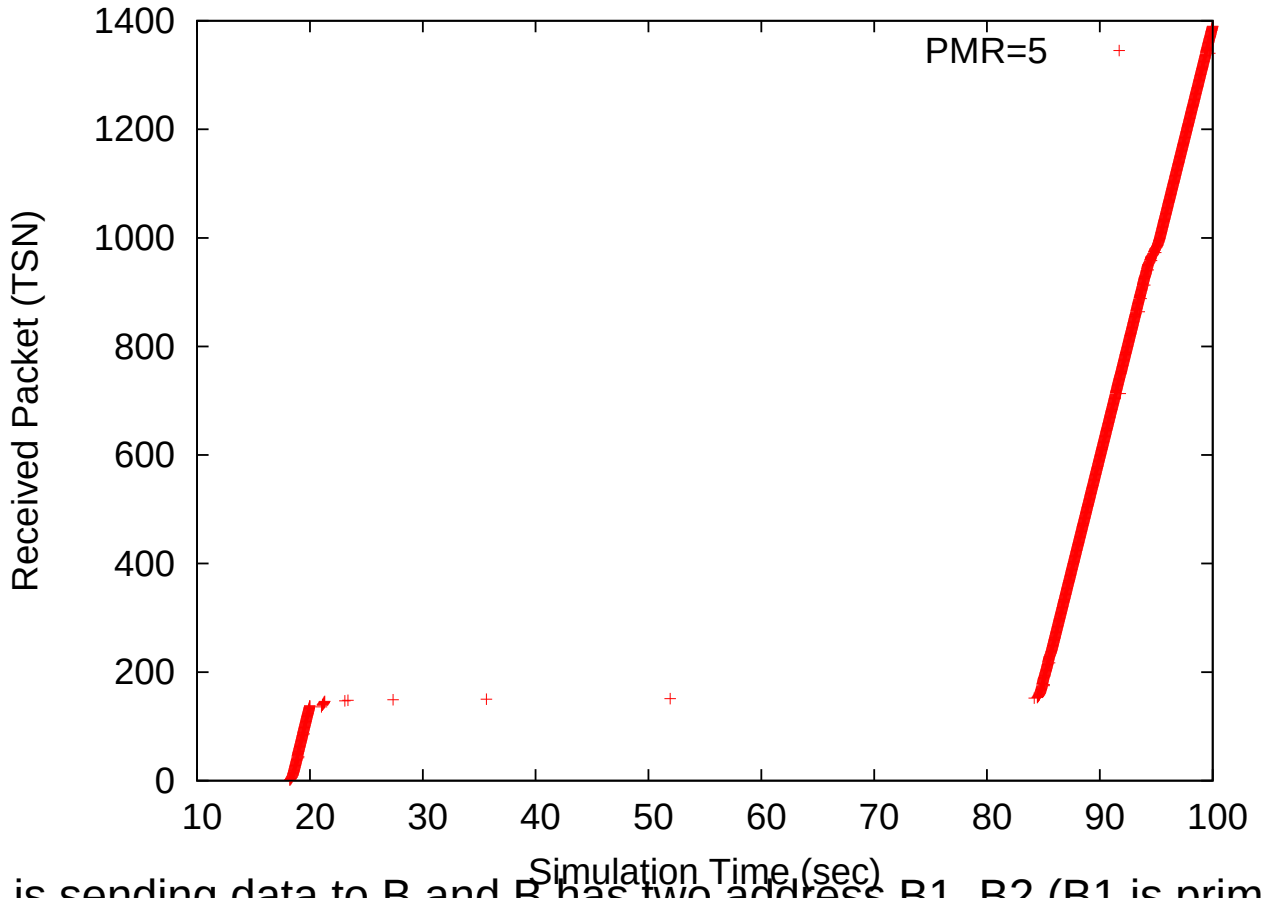
- Path.Max.Retrans is recommended to be 5 in standard
  - SCTP needs 6 consecutive timeouts before failover
  - RTO is doubled on each timeout
  - Only retransmitted packets can reach the receiver during failover process



A is sending data to B and B has two address B1, B2.
when B1 becomes unavailable, SCTP keep using B1 until 6 timeouts

An Example for SCTP Failover

- Simulation result using ns-2.34



A is sending data to B and B has two address B1, B2 (B1 is primary) when primary becomes unavailable at 20 sec, it takes 60 secs to restart data transmission. (Path.Max.Retrans = 5)

Possible Solution (1)

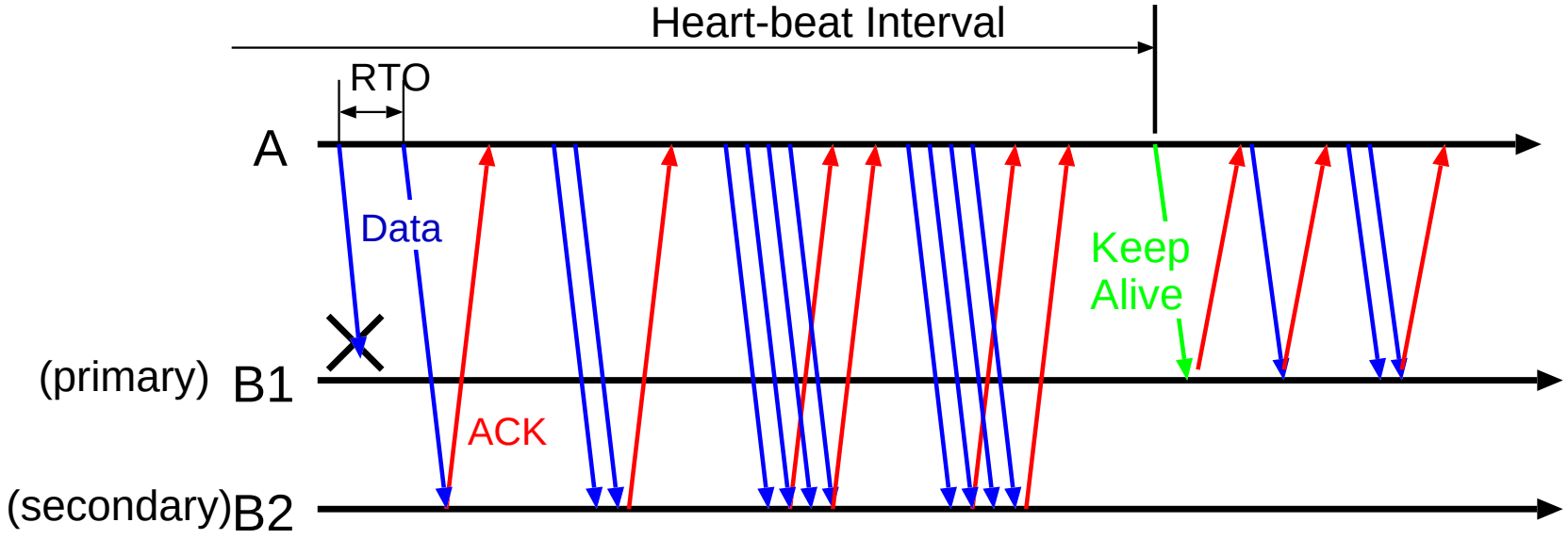
- Adjust RTO related parameters
  - The more RTO is small, the more SCTP can failover quickly
    - Using smaller value for RTO.max
    - Using smaller RTO.initial or RTO.min will also be effective
- Pros
  - Simple, no need to modify kernel
- Cons
  - Need to have enough knowledge about path
    - Otherwise, it can cause adverse effects

Possible Solution (2)

- Reduce Path.Max.Retrans
  - If Path.Max.Retrans = 0, SCTP switches to secondary on a single timeout
- Pros
  - Simple, no need to modify kernel
- Cons
  - A small violation of RFC (recommended PMR is 5)
  - Need to consider Spurious failover
  - Need to consider Asoc.Max.Retrans

Spurious Failover Issue

- If PMR is small, minor congestion can trigger failover
  - Once failover happens, it will take long to back to the primary
    - Recommended interval for heart-beat is 30 seconds



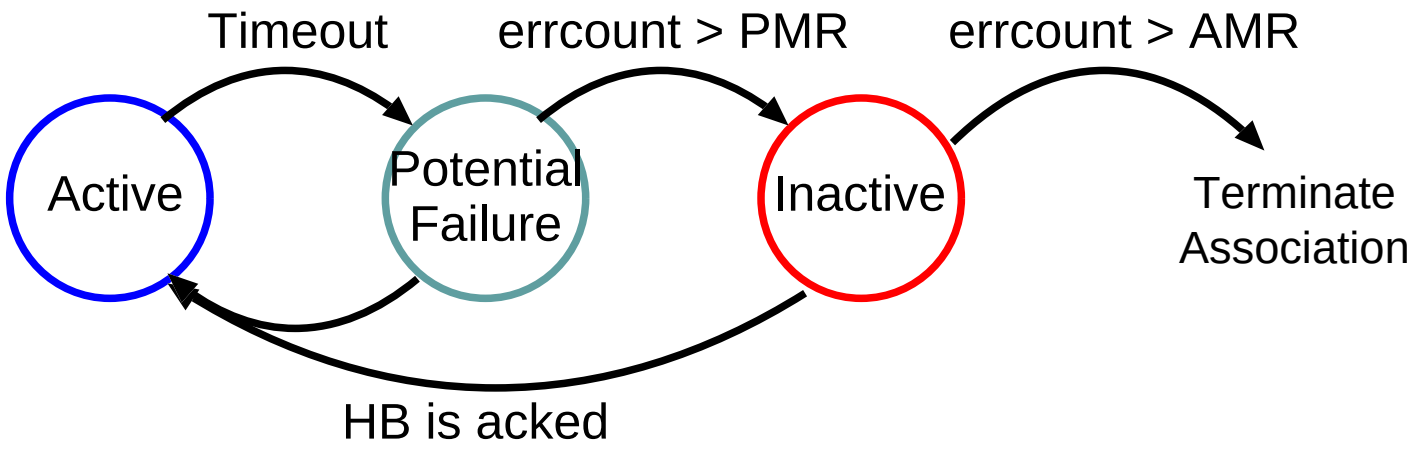
A is sending data to B and B has two address B1, B2.
when a timeout happen on B1, SCTP switches to B2
and doesn't go back to B1 until Heart-Beat is ACKed

Association.Max.Retrans

- Threshold for the total of error count for all pathes
  - If error count exceed this threshold, association will be terminated
- It shouldn’t be larger than sum of PMR of all pathes
  - Otherwise, even if all destination become inactive, endpoint still considers the peer reachable.
- But, if we reduce Assoc.Max.Retrans, association will be terminated with minor congestion

Adding New State in Path Management

- Difficulty in SCTP Path Management
  - SCTP needs to satisfy contradictory requirements
    - Respond network failure quickly
      - Need to mark path inactive as soon as failure is detected
    - Be robust against network congestions
      - Need to be conservative to mark path inactive
- One solution: Introduce an intermediate state



Possible Solution (3)

- Introduce Potential Failure (PF) State
  - Path is possibly inactive, but not confirmed yet
  - During PF state, Secondary path is used for data transmission
  - If primary respond to heart-beat, go back to the primary
    - Use new parameter PFHB.interval for heart-beat interval in PF state
      - Allow to go back to the primary quickly
- Pros
  - Use secondary path quickly
    - Go back to primary quickly when primary is active
  - No need to change PMR, AMR, HB.Interval
- Cons
  - Need to update kernel (only sender side)

Summary

- Adjust RTO related parameters
  - Simple. But not a common solution. Need to be used in limited situations
- Reduce Path.Max.Retrans
  - Simple, But, need to care about Suprious timeout issue and Assoc.Max.Retrans issue
- Potential Failure State
  - Need an extension to SCTP spec. however,
    - Algorithm is simple and easy
    - Only sender needs to be updated
    - No need to change current protocol parameters

Do We Really Need This?

- Several choices
  - Do nothing. 30-60 secs delay can be acceptable
  - Leave developers and sysadmins to solve this
    - Expect they will tune SCTP params appropriately
  - Modify default parameters in the spec
    - Some issues still remain
  - Add PF extension to the spec
    - More sophisticated solution
      - CMT draft already includes PF
- We believe
  - At least, we need to clarify the issue and document it
    - People can know the issue and its solutions
  - It would be better to have a standardized solution
    - Otherwise, implementors will try various ways for this