Dynamic Transport Selection

Shachar Hasson, UCX | UCF Conference Dec 2023

Agenda

- Problem Statement
- Design Goals
- Solution
- Benchmarks

Problem statement

- To optimize latency and message rate in large clusters, we aim to identify heavily used EPs and switch them to RC transport.
- Due to limited RC resources, we'd like to prioritize selecting RC for the most active EPs.
- Support varying traffic pattern and switch between RC and DC if needed.

Design Goals

- **Main goals:**
  - Avoid frequent switching
  - Don’t impact fast path performance
  - Low memory usage
  - Keep protocols semantics
- There are some tradeoffs between the different goals.

  For example, two methods for EP activity tracking:
  - Add extra bits in UCP_EP to store EP id, which will be used to access EP array (better for performance).
  - Use a hash table where the EP address is used as a key, only for highly used endpoints (better for memory)

Solution overview

Main components

- Monitoring/Selection - Identify highly used connections by packet count.
- Negotiation - Reach network agreement regarding which connections will be switched.
- Switching - How to switch transport under traffic.

Monitoring And Selection

Potential Naïve Solution

- Evaluate each connection by counting how much data is sent through it.
  - Add a counter per connection.
  - A counter is incremented per each send operation.
  - Identify the subset of connections with the largest count values.
  - Perform transport switch for the selected subset.
- 2 primary issues arise from this solution:
  - **Counter storage**
    - Storing a counter for each connection is not scalable.
  - **Counter update**
    - Adding an update operation in fast path will impact performance.
- **Due to memory scalability issues and fast path overhead, another approach is needed.**

Monitoring And Selection

Counter Update Design Approaches

- Counting can be performed using several methods:
  - **Timing**
    - When a send operation starts.
    - When a send operation completes.
  - **Layer**
    - UCT/UCP
- UCT
  - **Pros**
    - CQE Moderation can be utilized to coalesce several send operations into a single “counter update” operation.
  - **Cons**
    - Lack of such coalescing in non-IB transports.
    - Coalesce size is inconsistent between transports.
- UCP
  - Simpler implementation.
  - The lack of coalesce capability affects performance.
- **Intermediate approach**
  - Counting is done in UCP layer but toggled on and off alternately.
  - An extra branch instruction is still needed on fast path per each send operation.
  - It can be solved by modifying an existing branch operand value.

Monitoring And Selection

Counter Storage Solution - Multi-Stage Statistical sampling

- Maintain LRU structure which tracks recently used EPs and updates per each send completion.
- Statistically if we take a snapshot, highly active EPs will most likely be on the top.
- To prevent momentary peaks from influencing the selection decision, multiple samples over time are needed.
- Periodically sample LRU results and aggregate them into an **exponential decay** (ED) score table with a single entry per connection.
  - This stage is required to filter out noise
- The list of most active connections is defined by the subset of connections with the highest ED scores

Monitoring And Selection

Counter Storage Design Approaches

- **Moving window approach**
  - FIFO based approach which considers the most recent LRU samples.
  - High memory consumption.
- **Jumping window approach**
  - Hit counter based score.
  - No need to maintain a FIFO.
  - Extra stage adds complexity.
- **Exponential Decay**
  - Reward heavily used connections by raising their score each time they are sampled.
  - Idle connections score is lowered, as no new data are sent.
  - Older connections will be harder to replace, as we give weight to history when calculating score.
  - Update equation: current_score = A * current_score + B
    - A – decay coefficient.
    - B – new sample value.

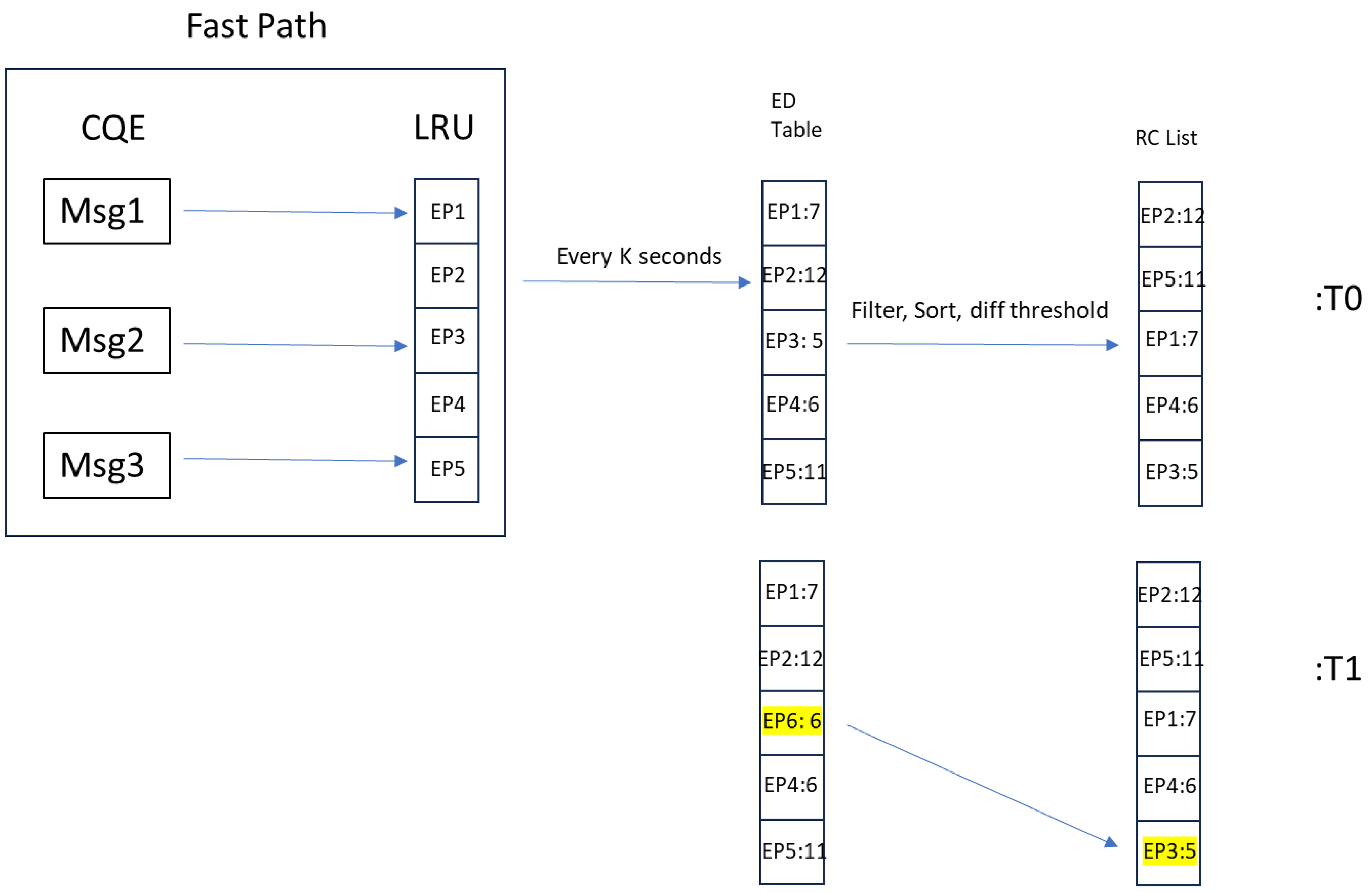
Monitoring And Selection

Usage Tracker

- Track and prioritize connections according to their usage.
- Generic UCS data structure, independent of particular transports.
- Avoid storing data per connection, by only tracking a small number of connections.
  - Total memory footprint is constant, rather than O(n) of total EPs number.
- **Main Terms**
  - Promotion – transition of a connection to a “highly used” connection.
  - Demotion – the opposite of promotion.
- **High-level implementation**
  - Maintain a connection table which corresponds to highly used connections.
  - On each “progress” operation, the LRU cache is flushed into the table to produce updated scores.
- **Entry points**
  - usage_tracker_touch
    - Touches the connection entry for each new packet send operation.
  - usage_tracker_progress
    - Updates the connection table with new scores and adjusts it if required.
    - Called from UCX periodic callback context.
- **Output**
  - Callback notification of promotion and demotion events.
- Asymmetric bidirectional connections can be updated according to remote side.

Monitoring And Selection

Usage Tracker data structures



Transport Negotiation Protocol

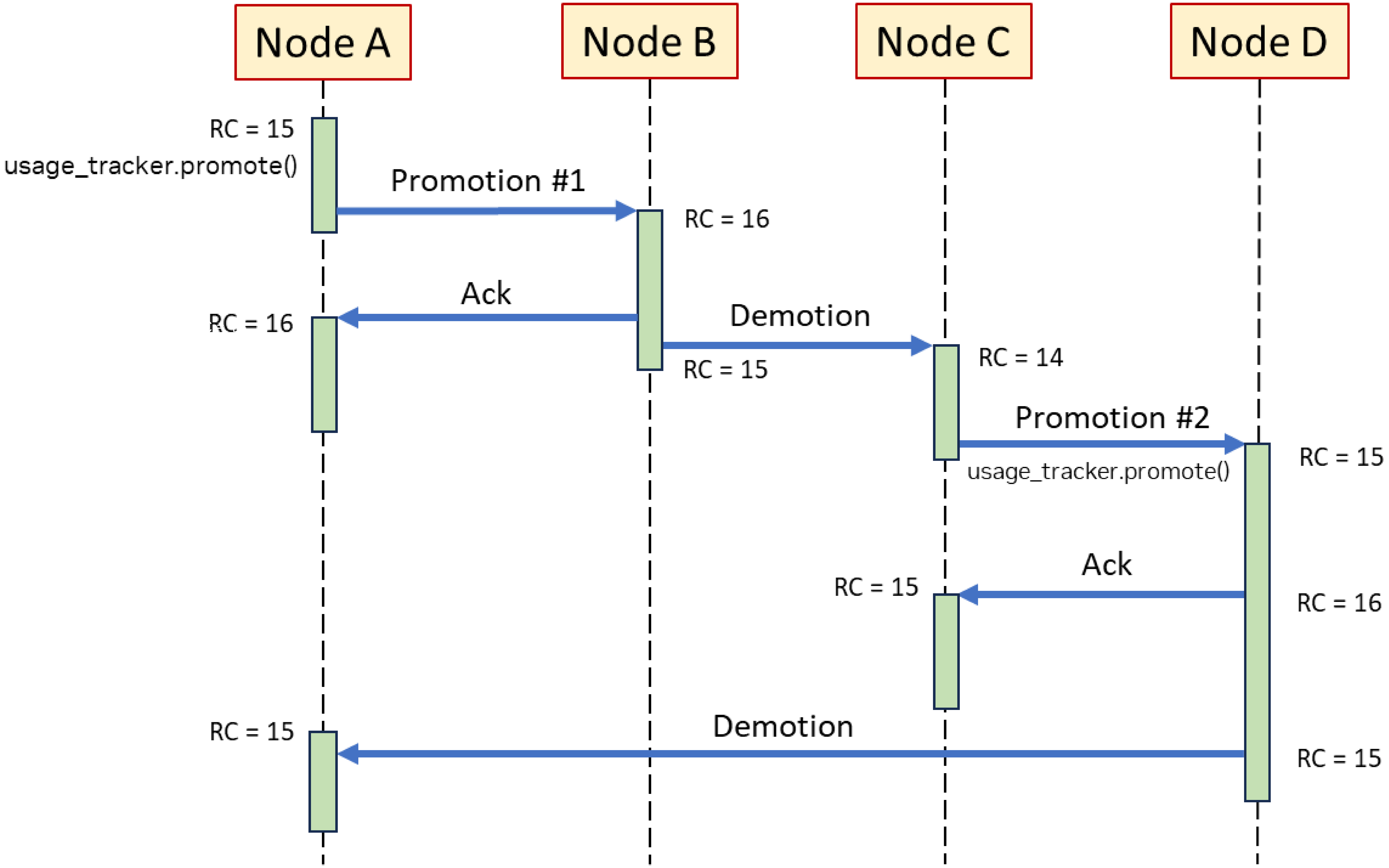
Network perspective

- As each EP involves 2 nodes, they may have different views regarding traffic amount relative to other EPs.
- Furthermore, a node has no knowledge of its neighbors' RC capacity (a remote node may have exhausted its RC resources)
- Thus, a new protocol is needed to ensure all nodes agree on EPs transports to be used
- The most efficient allocation would require looking at the whole cluster "from above" and having full information about all connections
- As it is not practical, a "close enough" approximation is made instead.
- The new protocol must ensure consensus and avoid infinite loops caused by cyclic switching patterns.

Transport Negotiation Protocol

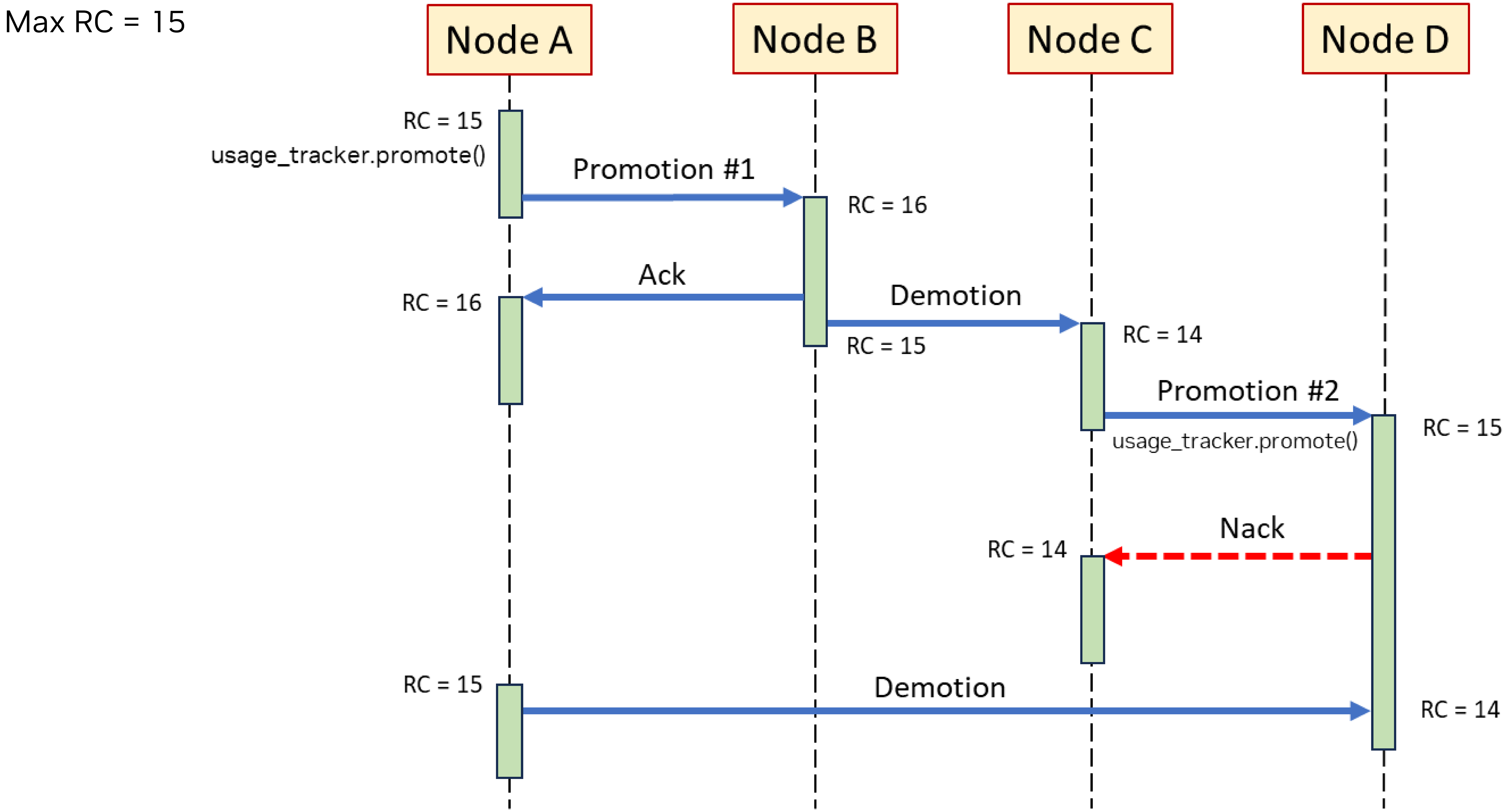
High Level Flow

Max RC = 15



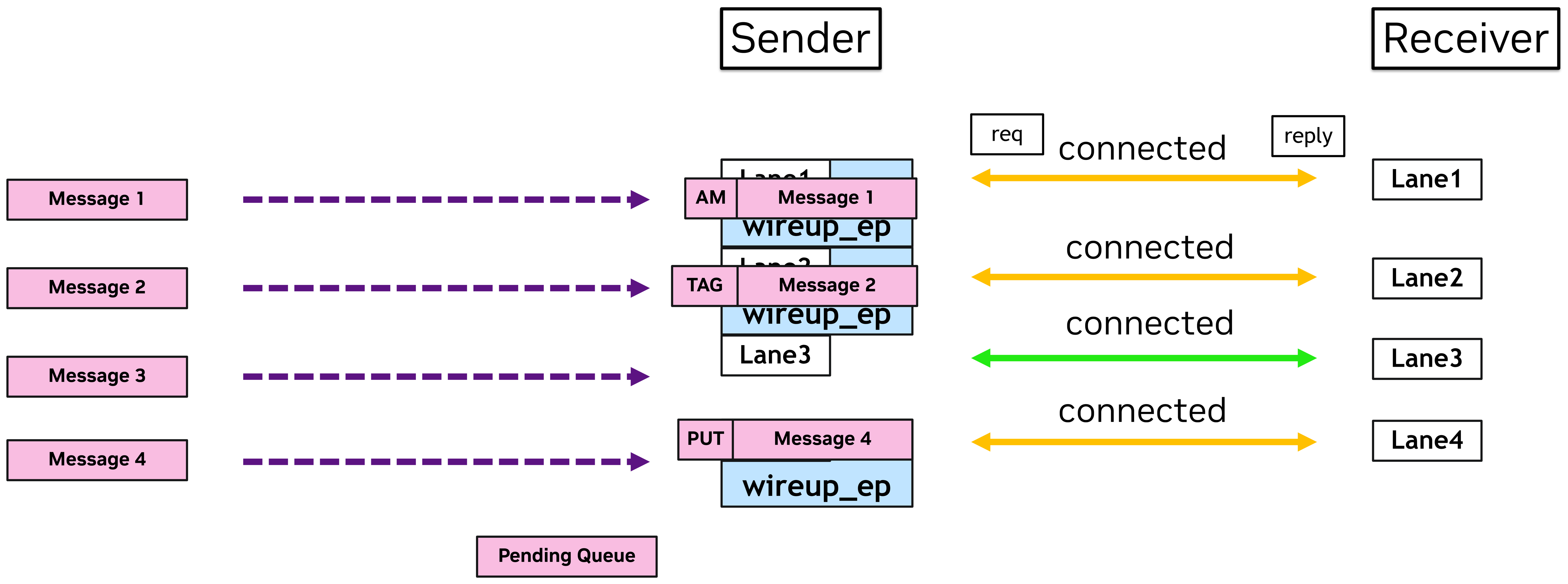
Transport Negotiation Protocol

Request Denied



Switching

Wireup Process



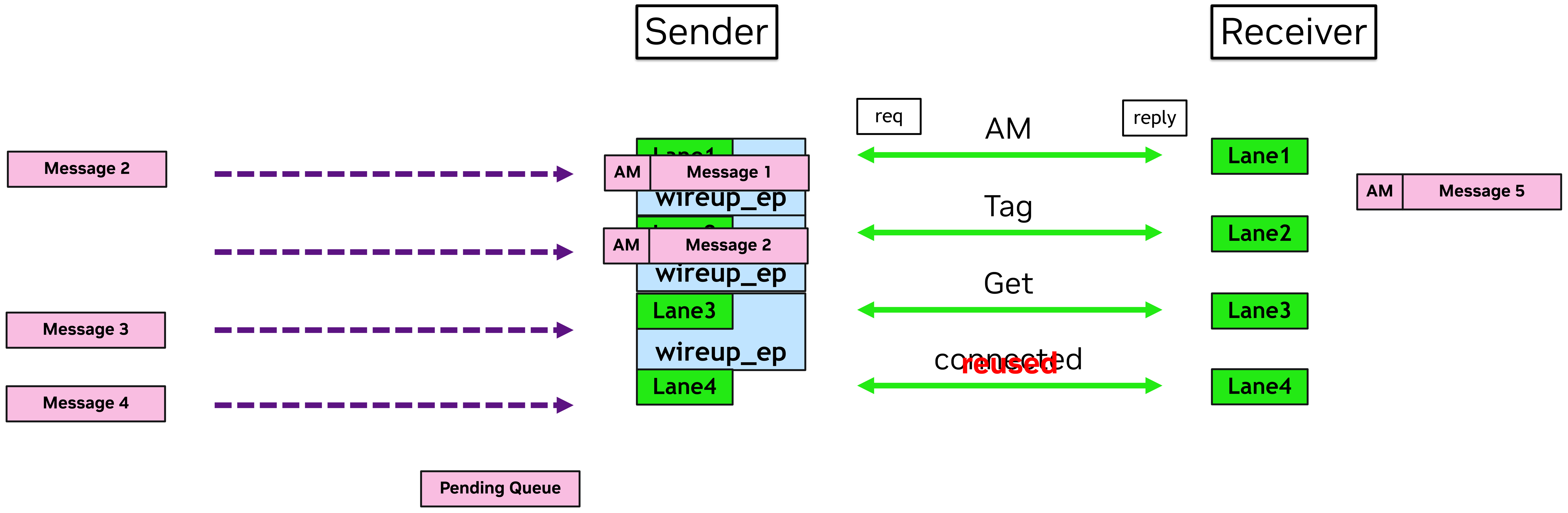
Switching

Overview

- The process of replacing the set of active connections, under traffic.
- Order should be guaranteed for active message transports.
- Reuse UCT endpoints if possible.
    - A new UCT API is implemented to determine whether a lane is connected to a remote side described by a remote address.
- Pending requests are handled by the new connection.
- Outstanding requests are flushed.
- Multi-fragment requests reset the UCP protocol.

Switching

Reconfiguration Scenario



Benchmarks

- **Basic benchmarks**
  - osu_mbw_mr
    - Few highly active EPs and a lot of unused EPs
    - Verifies switching of the correct EPs
  - osu_alltoall
    - A symmetric scenario where all EPs send a lot of data
    - Checks avoidance of excessive switching
- **Improved benchmarks**
  - osu_mbw_mr
    - An extra send operation was added to all unused EPs
    - Better simulation of real use scenarios

NVIDIA