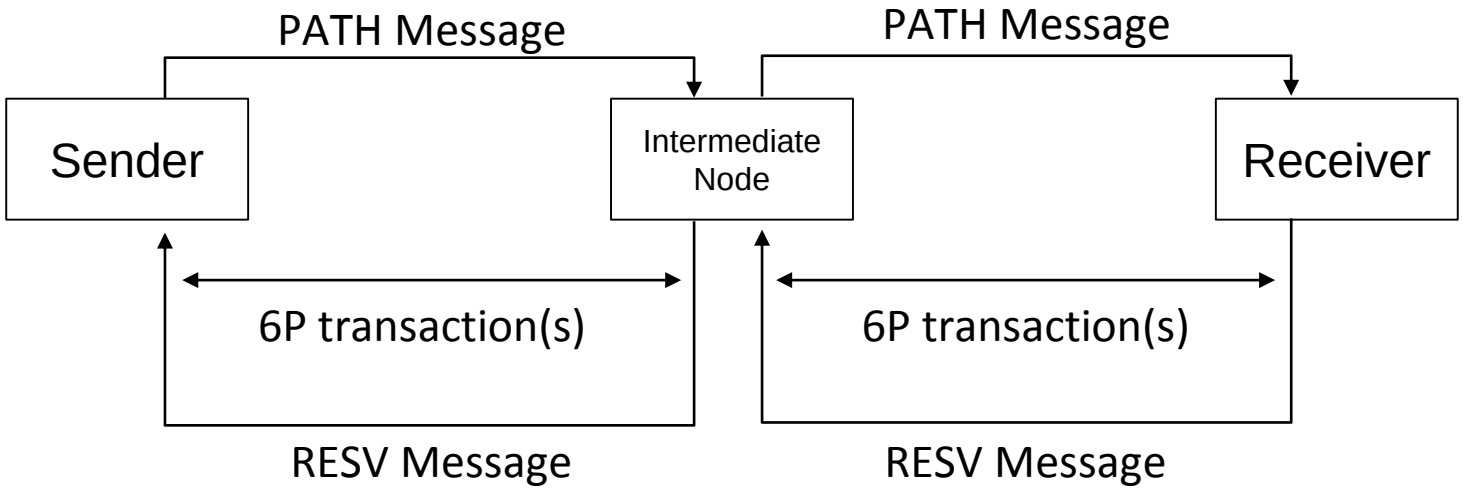
draft-satish-6tisch-6top-sf1-04

Satish Anamalamudi
Bing (Remy) Liu
Mingui (Martin) Zhang
Abdur Rashid Sangi
Charles E. Perkins
S.V.R Anand

Introduction to SF1

- Objectives
  - Reserve a track to a destination multiple hops away
  - Fulfill bandwidth and QoS (e.g., priority, time-critical) requirements
  - Hop-by-hop, distributed
- Combination of RSVP-TE and 6top protocol (6P)
  - RSVP-TE: end-to-end resource reservation
  - 6P: cell negotiation between two neighbors
- When to trigger SF1?
  - The Sender has an outgoing bandwidth requirement for a **new instance** to transmit data to Receiver.
  - The Sender has a **new outgoing bandwidth requirement** for an existing instance to transmit data to Receiver.

End-to-end operation



Step 1: The sender sends a PATH message downstream hop-by-hop to the receiver.

Step 2: The receiver initiates a 6P transaction to verify if there are enough cells to fulfill the requirements. If so, the receiver sends the RESV message to reserve these cells.

Step 3: Upon receiving a RESV message, an intermediate node reserves the cells between its upstream neighbor in the same way as in Step 2.

Step 4: When RESV arrives at the sender before end-to-end timeout, a track from sender to receiver is built.

PATH and RESV messages

```
<PATH Message> ::=    <Common Header> [ … ]
                      <SESSION> <RSVP_HOP>
                      <TIME_VALUES>
                      [ … ]
                      <LABEL_REQUEST>
                      [ <SF1 OPERATION REQUEST> ]
                      [ <6P OPERATION REQUEST> ]
                      [ … ]
                      <sender descriptor>
<sender descriptor> ::=  <SENDER_TEMPLATE> <SENDER_TSPEC>
                              [ <ADSPEC> ]
                              [ <RECORD_ROUTE> ]
```

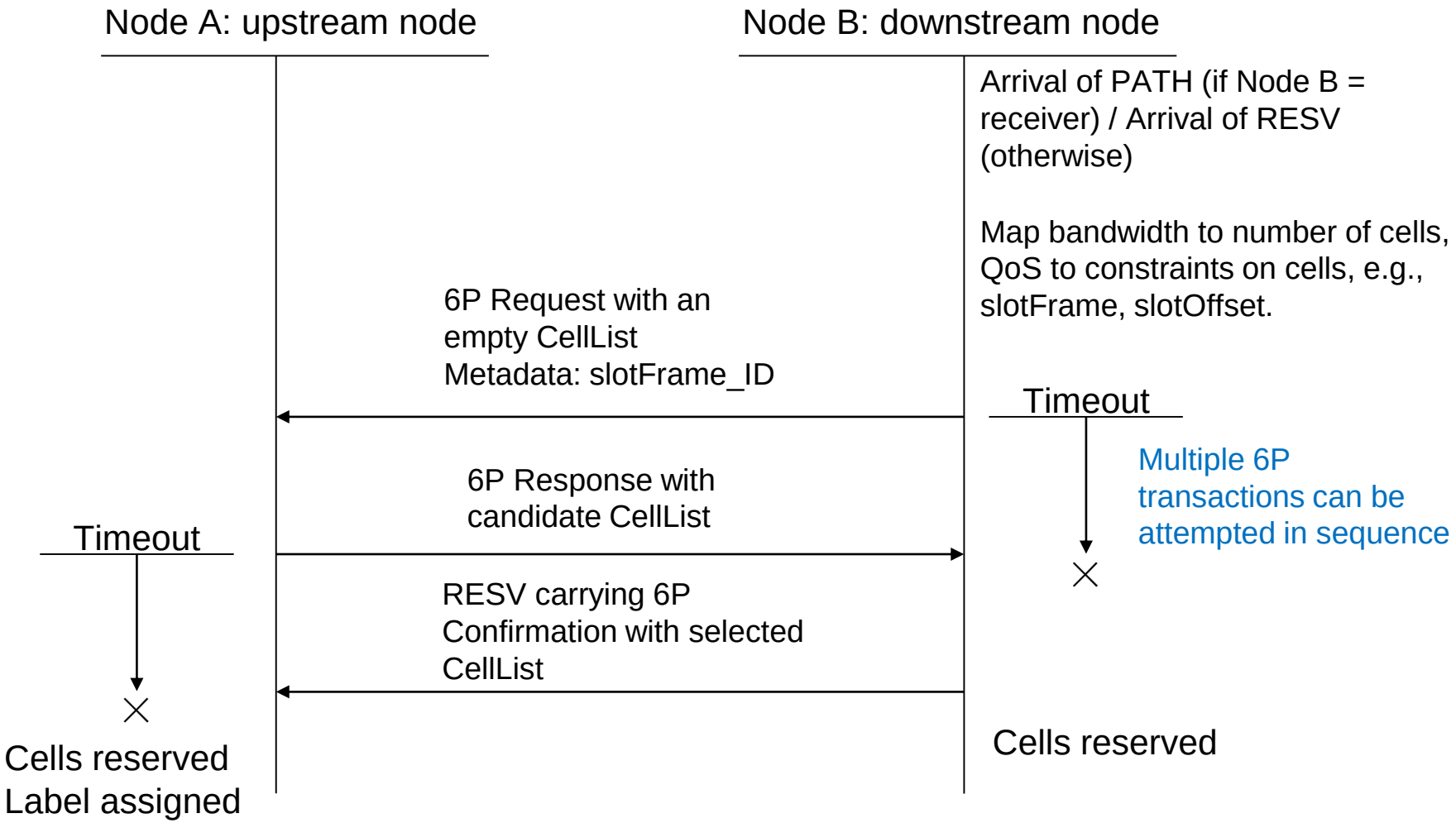
- PATH: describe the flow (SENDER_TSPEC) and collect path properties (ADSPEC).
- The three request objects: used to verify if the nodes along the route have the requested capabilities.
- The LABEL_REQUEST is set to TSC (timeslot switching capability).

```
<RESV Message> ::= <Common Header> [ … ]
                   <SESSION> <RSVP_HOP>
                   <TIME_VALUES>
                   [ <6P OPERATION> ]
                   [ … ]
                   <STYLE> <flow descriptor list>
<flow descriptor list> ::= <FF flow descriptor>
<FF flow descriptor> ::= [ <FLOWSPEC> ] <FILTER_SPEC>
                         <LABEL>[ <RECORD_ROUTE> ]
```

- RESV: describe the bandwidth and QoS requirements (FLOWSPEC), assign label to the upstream node (LABEL).
- Parameters in FLOWSPEC are calculated according to SENDER_TSPEC and ADSPEC.
- In case of 3-step transaction, 6P confirmation is encapsulated in 6P OPERATION.

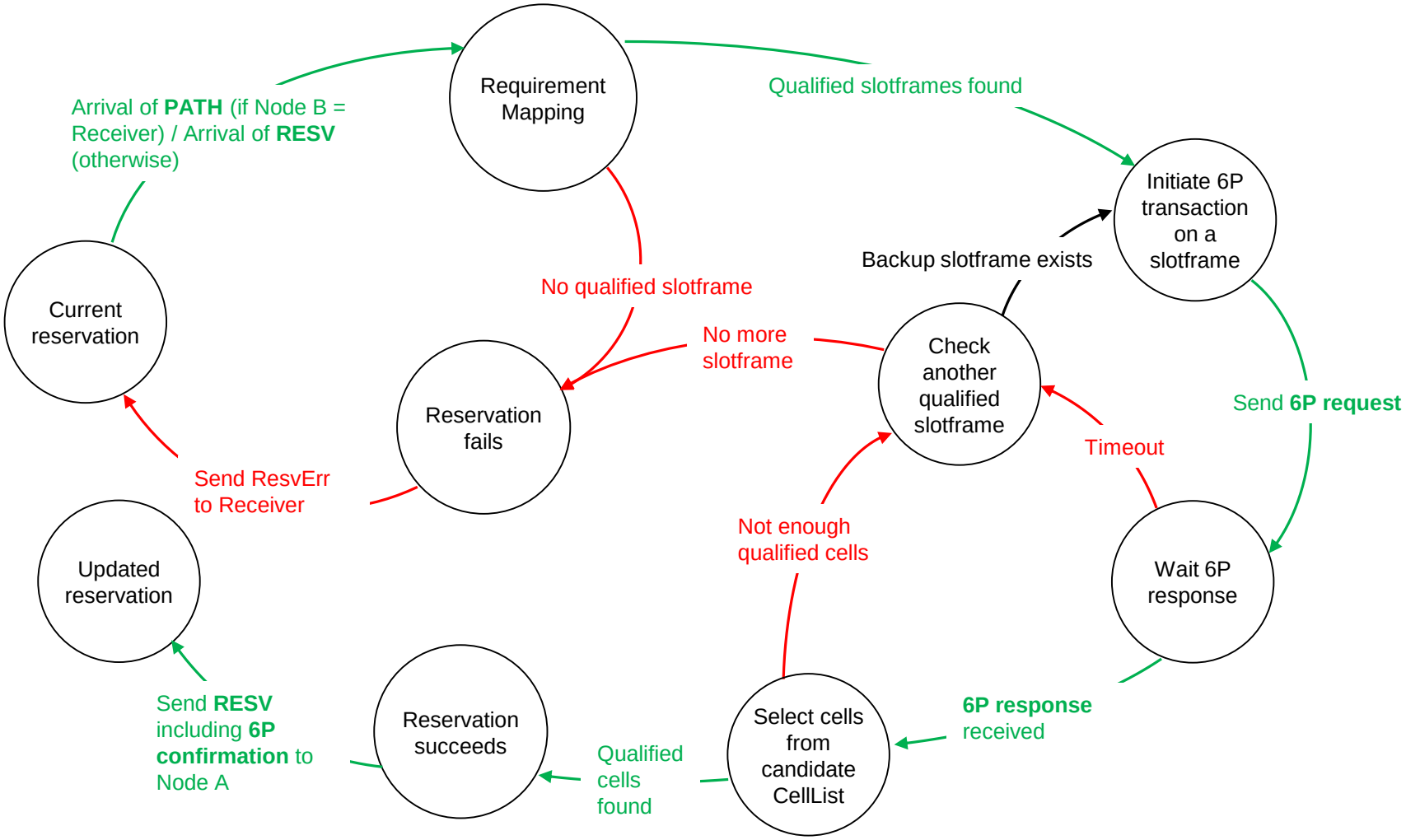
One-hop operation using 3-step trans. (updated)

Note that the 2-step trans. can also be used.



Node B State Transition Diagram

Node A: upstream node
Node B: downstream node



TrackID, Label and Bundle

- TrackID in SF1 (updated definition)
  - 16 bits identifier, assigned by the sender
  - Mapped from (source & destination IP address, RPLInstance)
  - Encapsulated in SENDER_TEMPLATE of PATH and FILTER_SPEC of RESV
- Label (updated definition)
  - 32 bits, mapped from a bundle between two neighbors
  - Encapsulated in the LABEL object of RESV
  - Locally valid between two neighbors, assigned by the downstream node
  - Associated to a track
- Bundle between two neighbors
  - A group of equivalent scheduled cells (slotFrame_ID, CellList)

Next steps

- Complete the definition
  - 6P request, SF1 request, 6P operation, Teardown message, etc.
- Mapping the traffic requirements to cells
- As suggested in 6top, the following requirements need to be covered
  - Error Handling (more detailed error code)
  - Specify the SF behavior of a node when it boots
  - Security considerations
  - Examples
- Implementation: simulation and hardware deployment

Thank you!

Track Forwarding

- Not in the scope of SF1, just for the completeness of the story
- **A track can be seen as an LSP using bundles as implicit labels**
- The sender identifies which track a packet should follow based on “sender/receiver IP address, RPLInstance”.
- Then G-MPLS is used
  - The sender pushes the first label to the packet.
  - The label is swapped at each intermediate node
  - The label is popped out at the penultimate Hop or at the receiver
  - At each hop, the packet is forwarded using the bundle associated to the label.