PUBLISH and SUBSCRIBE to Tracks in a Namespace

IETF in Bangkok, March 2025

Ian Swett, Cullen Jennings

Problems / Key Use Cases

Conferencing:

- Two people want to join a call, meeting, or gaming session where they can talk to each other
- User experience is one person joins first, and as second one joins, they see the media of the first person and say “Hi”.
- Do not want to clip or loose the “Hi.”

Video Ingestion:

- Publisher wants to connect to and immediately start sending video

Relevant Issues

Video ingestion is awkward and slow #778

- A client can't initiate publishing

No way to Subscribe to all Tracks in a Namespace #779

Support wildcard Subscribes #615

No one's sure what to make of Track Alias (#350, etc)

Original Wildcard Subscribe Issue #253 was closed when SUBSCRIBE_ANNOUNCES was added, but didn't fix it.

Observation from HTTP, QUIC, etc

HTTP has GET and POST

QUIC has Client and Server initiated streams

MoQ has HEAD (Track Status) and two GET variants,
But no POST

Part 1: Add PUBLISH

Almost identical to SUBSCRIBE, but sent by Publisher

Because they're almost identical once established,

- SUBSCRIBE_UPDATE and other messages are used
- Only adds PUBLISH and PUBLISH_OK messages

Part 2: Variant of Alan's Track Alias Proposal

1. Remove the Track Alias field from SUBSCRIBE and remove the RETRY_TRACK_ALIAS error code
2. Replace Track Alias with Publisher ID, which is odd, and require Subscribe ID to be even
3. Replace the Track Alias field in Object and Subscribe ID field in control messages with `Track Identifier`. A Track Identifier can be either a Subscribe ID or a Publisher ID, determined by LSB.
4. If a subscriber receives an Object with an unknown Track Identifier, it MAY drop it.
5. A Publisher ID MUST NOT be reused within a Session, otherwise no restrictions are needed.

PUBLISH re-creates a past issue #145

Objects can be received with an unknown ID

But sessions and streams have flow control and implementations buffer 0-RTT packets today

Alan's Track Alias proposal enables using either ID

- Subscribe ID is even, Publisher ID is odd (ie: QUIC)

Possible Pub-Before-Sub (Alan's slides)

```
PUBLISH
{
  Type (i) = 0xTBD1,
  Length (i),
  Publisher ID (i),
  Track Namespace (tuple),
  Track Name Length (i),
  Track Name (..),
  Group Order (8),
  ContentExists (8),
    [Largest Group ID (i)],
    [Largest Object ID (i)],
  Number of Parameters (i),
  Subscribe Parameters (..) ...
}
```

```
PUBLISH_OK
{
  Type (i) = 0xTBD2,
  Length (i),
  Publisher ID (i)
  Subscribe ID (i),
  Subscriber Priority (8),
  Number of Parameters (i),
  Subscribe Parameters (..) ...
}
```

Part 3: SUBSCRIBE to a Namespace (Wildcard)

If the ‘Track Name’ field is empty, Subscribes to all active Tracks in the Namespace, resulting in PUBLISH(es)

- Use a new message type if the WG prefers

If trackname is empty, requests a PUBLISH for all Tracks that exactly match the specified Namespace.

Cancelling the Namespace (Wildcard) Subscription cancels all corresponding publisher initiated Subscriptions

Individual publisher initiated Subscriptions can be updated, cancelled, etc using **either** the Publisher ID or Subscribe ID

Wildcard was requested 2 years ago, Why now?

SUBSCRIBE is better defined, Priorities are defined

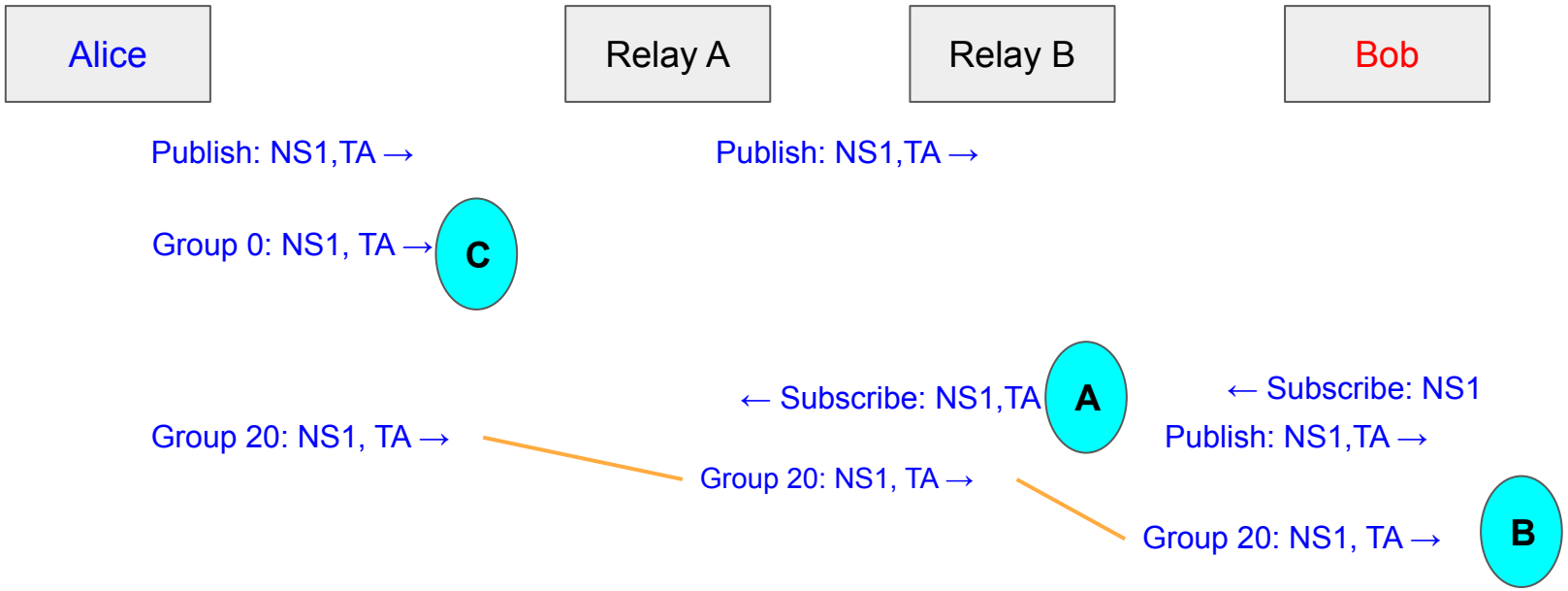
Better understanding of compelling use cases

Alan's even/odd Subscribe ID/Publisher ID enables either peer to initiate a Subscription

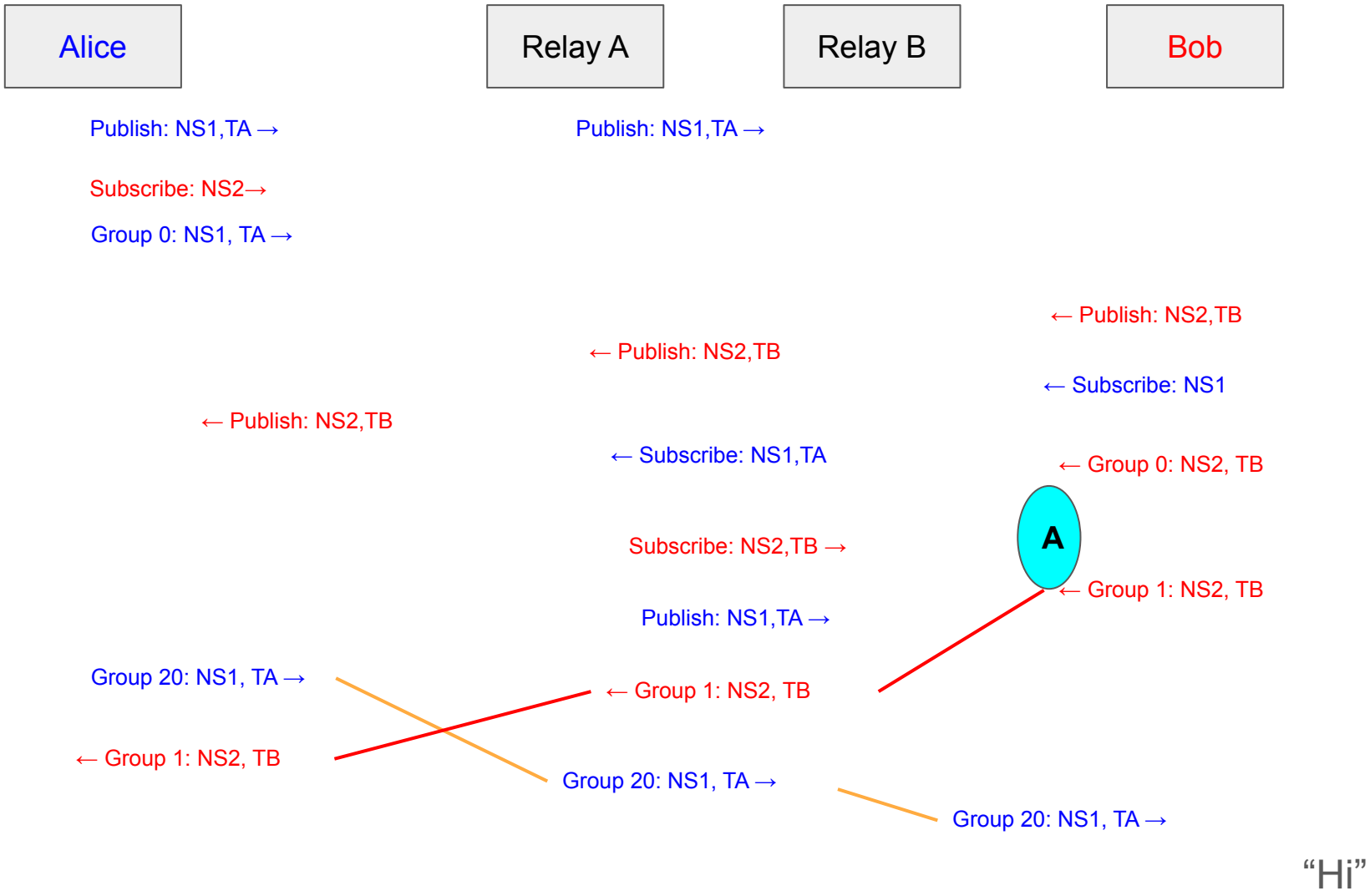
The Namespace Tuple provides a structure for wildcards

Ian Proposal (Ingestion)

Ian Proposal (Just one way media to help understand flow)



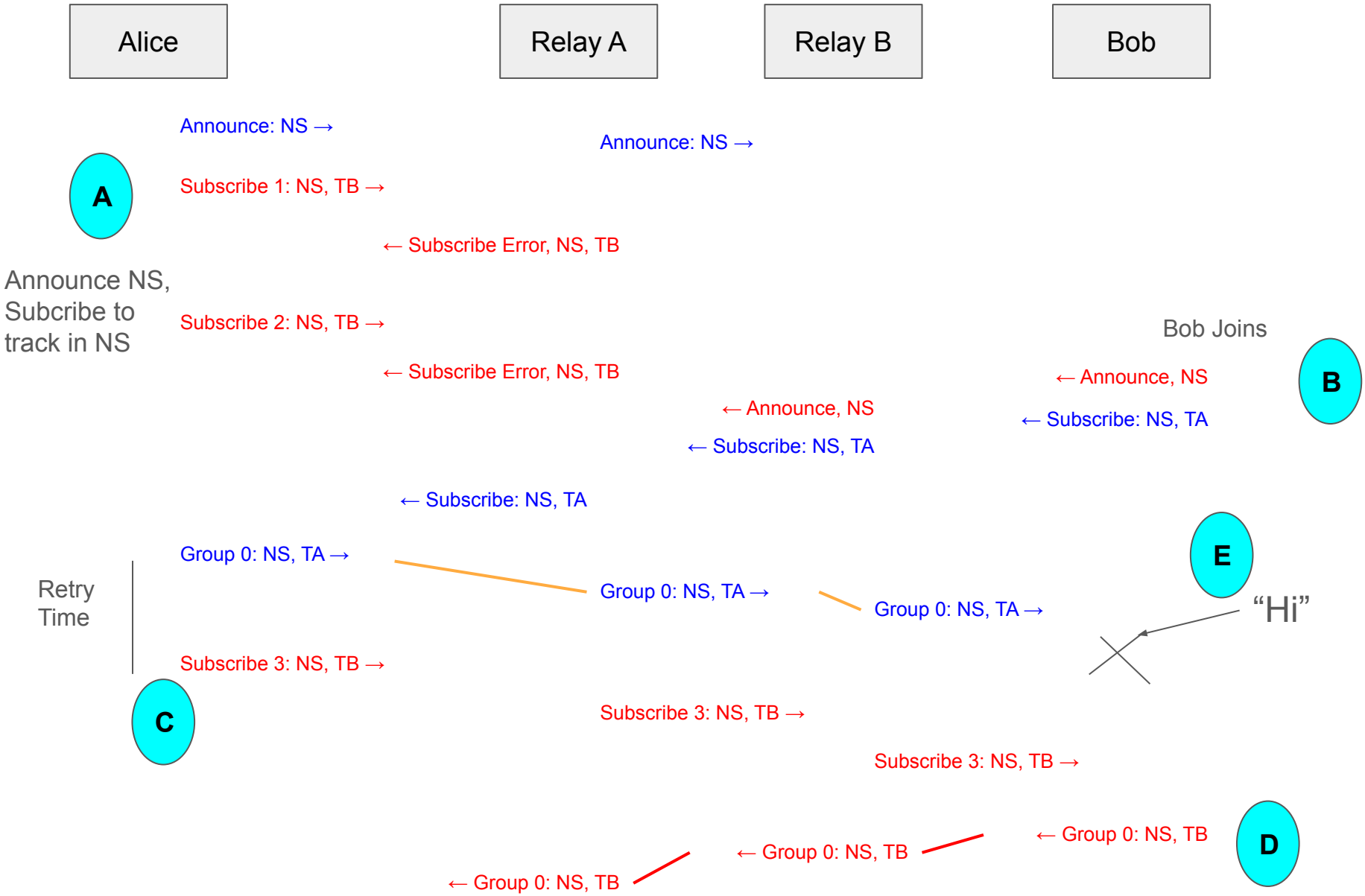
Ian Proposal (2 way media)



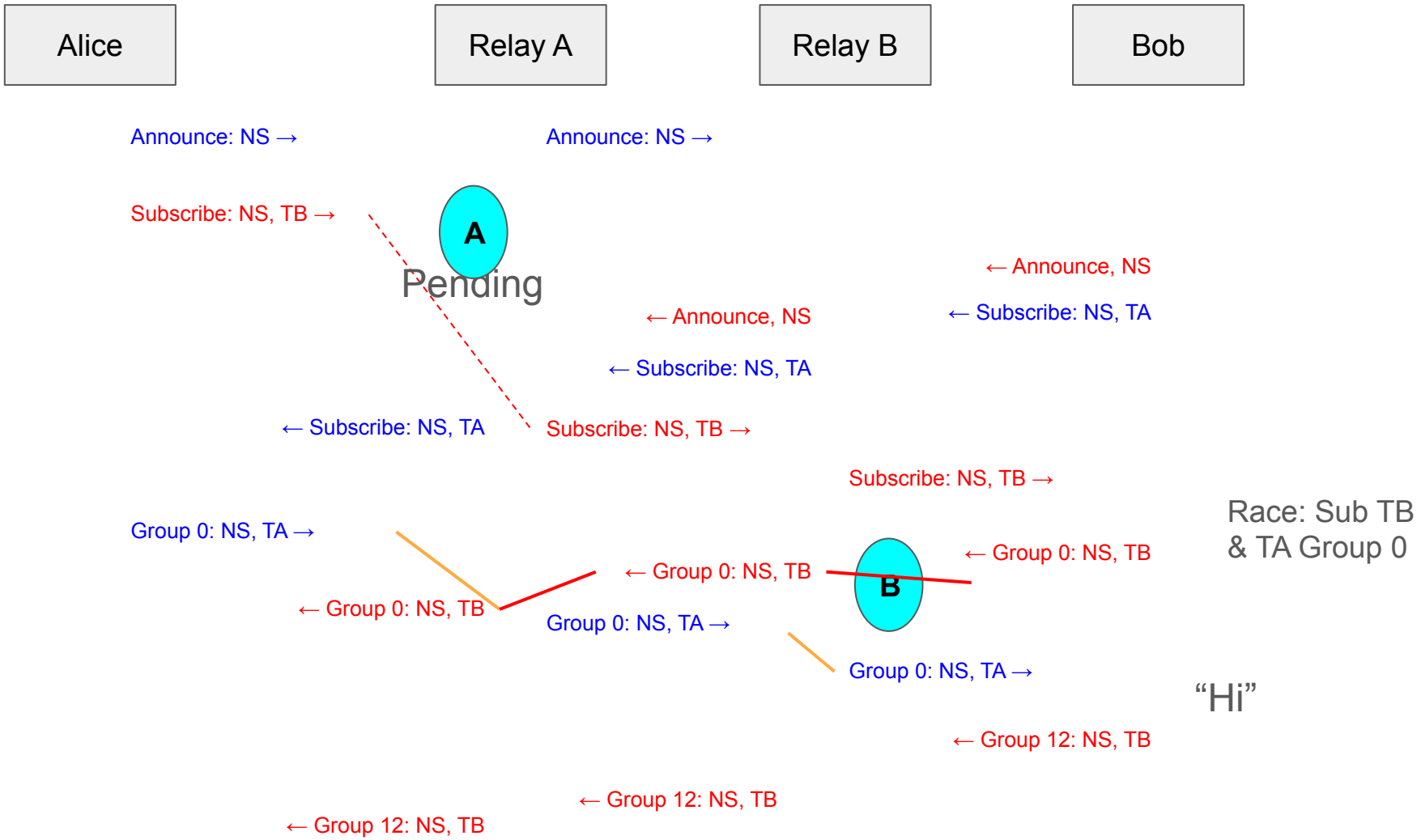
"Hi"

"Hello"

Re-Try Subscribe (aka polling)



Pending Subscribes (not in current draft)



Publish before Subscribe - Removes Latency (not in draft)