Packet Forwarding Related Issues

NDN Retreat
2016/11/3
Goal

• We need a specification of NDN packet forwarding behavior.
• Examine issues from past research, implementation, and operation.
• Consider other people’s alternative designs.
Selector Processing

• Since selector processing is slow, can we allow some routers skip selector processing without breaking system correctness?

• Suggestion: move selectors to be part of Interest name. (/N/S)
  • Routers that don’t process selectors can forward the interest as it is.
  • Returned Data (/N/x) needs to be encapsulated (/N/S/x).
  • A new name component type/marking for selectors.
Data/PIT lookup

• Look up an incoming data against the PIT
  • It’s not exact match since data name can be longer than interest name.
  • One data can satisfy multiple interests. In some scenarios we’d like to know which interest brought back the incoming data.

• Idea 1: Interest Digest
  • PIT stores the digest of the interest, and the data packet carries the digest.
  • How to ensure every node uses the same digest function?

• Idea 2: Use a unique index number, different at each hop.
  • The downstream attaches the index number of PIT entry to the interest pkt (in the link layer).
  • The upstream returns data together with the PIT index number (in the link layer).
What to do when the PIT is full?

• Tell the downstream to slow down, part of flow/congestion control.
• Also need to drop a PIT entry, but which one?
  • The incoming interest
  • A random PIT entry
  • The oldest PIT entry
NACK in multi-access media

• NACK is useful on p2p links, but what should be the behavior in multi-access media?
  • When there’re multiple upstream nodes, one NACK doesn’t mean anything since another upstream may be trying to retrieve the data.

• Idea:
  • the downstream chooses one upstream node as the “designated forwarder”, which can forward the interest further and can return NACK if it fails.
  • Other upstream nodes can return data from its cache or local producer, but don’t forward the interest, and don’t return NACK.
  • The “designated forwarder” can be identified during the routing stage.
  • If an interest doesn’t carry “designated forwarder” (in the link layer), any upstream node can forward the interest, but don’t send NACK.
Repo

- in wide area: if repos for a namespace are not co-located with producer, how to get Interest into repo
  - Each repo makes routing announcement of the same repo prefix
    - break aggregation, like IP anycast.
  - Encapsulate data under the hosting network’s prefix or use link object.
  - do we need to look use cases one by one?

- Once an interest reaches NFD of the final node, how to direct it to the repo process?
  - the repo may host data with different name prefixes, does the repo have to register with NFD of all the prefixes?
  - feasible to use bloom filter here??
Interest Multicast

• Need to multicast interests to a group of participating nodes
  • E.g., ChronoSync

• Approach 1: every participant announces the same sync prefix, /ndn/sync/..., and routers adopts multicast strategy so that any interest will be forwarded to all participants.

• Approach 2: build an overlay to connect the participants, and broadcast the interests on the overlay.
  • Doesn’t need special router support.
  • But do need a separate app-level protocol to build and maintain the overlay.
• How liberal should we make use of LINKs in Interest forwarding?
  • Currently we try to put app-name after node-name, so that routing can steer each interest to the right producer.
  • Some people propose to move app name up to make it easy for forwarding based on app names. But this would require interest to carry a LINK to steer it to the producer node.
Undetected “loop” (by J. J.)

- I1 and I2 carry the same name but different nonces.