draft-moiseenko-icnrg-flowclass

Dave Oran daveoran@orandom.net

Ilya Moiseenko ilmoisee@cisco.com

Note: Cisco IPR on this draft

Differentiated Services for ICN

- Problem statement: ***How do we support multiple classes of traffic in ICN?***
  - Specifically NDN or CCN – they are essentially identical in this regard
- Two aspects to this, relatively independent of each other
  1. How do we tell the forwarders what our desired treatment of packets is
  2. How do we group packets into equivalence classes (aka "*Flows*") for similar treatment and distinguish from other equivalence classes that ask for the same treatment?
- There is some high-level discussion of #1 in draft-xia-icn-multiservtag-00
- This draft addresses **#2**.

General Challenges in providing Differentiated Services

- If too finely granular (spatially or temporally) there are serious scalability problems in queuing/policing/shaping state
- If too coarse you cannot separate traffic a fine enough level to have meaningful fairness
- If not securely encoded, trust across domain boundaries is problematic (c.f. IP Diffserv)
- If securely encoded, flow aggregation is tricky/impossible

Differences from IP

- No 5-tuple, specifically no source addresses
  - This is the general “flow“ descriptor in IP
- Symmetric routing allows flowstate to mirror FIB state
- Pull-based interaction allows clean separation of
  - producer desire to be the specifier of traffic equivalence classes for its data
  - consumer desire to control the actual traffic treatment by the network.

Constraints

- Equivalence classes have to be bound tightly with the names in the corresponding Interest and Data packets
  - be stable over multiple exchanges
  - Be stable across a set of names sharing some common "handle"
- Simply using FIB does not provide a useful set of equivalence classes
  - Routing prefixes are too coarse; many equivalence classes of packets are generally covered by a single routing prefix
  - practical, scalable routing needs to do route aggregation, which further blurs the discrimination of the equivalence classes.
- Therefore, need to have something that both relates to the name structure but provides finer granularity for flow classification purposes.

Goals

- Devise a mechanism allowing ICN forwarders, consumers, producers to encode, decode, and process equivalence class identifiers (flows) at:
  - At least at granularity of a routable name prefix
  - More fine grained without scalability becoming intractable
- Lightweight encoding
- Reasonable security tradeoffs
  - Not clear we can achieve this with a single mechanism

Thoughts on Scaling

- What state must be kept on a per-flow basis when the flow count is very high?
- For consumers and producers, this state scales naturally with the number of applications and application interactions are going on simultaneously.
  - Therefore the scaling limit is not likely to be in the producers or consumers.
- For ICN forwarders that are operating at high speed and/or handling the traffic of many producers and consumers however, this state can scale quadratically or worse.
- If the ICN forwarder cannot keep all the state due to memory or processing limitations, it faces the common problem of which flows to remember and which to forget.
- We don't not solve this problem, which is fundamental, however…
  - The encoding schemes we define here provide a method for identifying equivalence classes using protocol machinery that already has to scale (e.g. name parsing and lookup) and hence does not introduce a new class of problems not inherently present.

Two possible mechanisms

- Include a *Name Component Count* in Interest and Data Packets
- Define an *Equivalence Class* name component type and put the equivalence class identifier directly into the content object names
- Details on following slides

Equivalence class component count (EC3)

- Set by a producer
- Counts the number of name components in the corresponding name that are considered one equivalence class instance.
  - This allows either finer (or coarser) granularity than a FIB prefix
  - producer can "regroup" equivalence classes dynamically by including more or fewer levels of the name hierarchy when they respond to Interests for the corresponding Data packets.
- EC3 could be inside or outside the security envelope
  - Outside permits ICN forwarders to modify it, allowing the aggregation/disaggregation of flows to be performed by the forwarders as well as the consumers.
  - Conversely, leaving the field outside the security envelope may enhance certain attack scenarios against flow classification for quality of service or firewall filtering

Equivalence class name component type (ECNCT)

- Producer encodes equivalence class information directly in the name, by adding a name component to the name of the content object(s)
  - Therefore immutable for the lifetime of the associated named data.
  - ECNCT present in Interest packets as well, and hence subject to both PIT and FIB matching.
- The Equivalence Class name component both names the equivalence class explicitly, and implicitly makes all Data packets named below it in the hierarchy part of that equivalence class.
  - Consequently, the name can have multiple equivalence classes markings (e.g. flow and sub-flows - see next side)
  - As with EC3 one can have either finer or coarser granularity than provided by FIB prefixes.
- In addition to the obvious uses by forwarders, ECNCT can be used by producers for:
  - QoS-driven demultiplexing of interests
  - load sharding

Consumer considerations

- Consumer can associate an arriving data packet with the correct equivalence class to manage subsequent Interest/ Data exchanges with the same name prefix and equivalence class identifier
- Associated measurements such as RTT or marginal delay can be leveraged to perform flow and congestion management for the equivalence class as a whole.

Forwarder considerations

- Forwarders need a flow instance granularity data structure (or its moral equivalent) in order support per-flow treatment of equivalence classes of Interests.
  - Typically, name prefixes in flow table are more granular than prefixes in the FIB, but less granular than names in the PIT.
  - As noted earlier: no magic pixie dust to sprinkle on the flow count scalability issues.

So, what can you do with all this?

Some examples:

1. Enforce rate control for the equivalence class as a whole (e.g., dropping packets, queuing packets, etc2.);
2. Estimate the number of simultaneous flows traversing a bottleneck link, which can improve the performance of many congestion control schemes; and
3. Make more intelligent selections of which packets to cache at the ICN forwarder, for example, to prefer to cache many packets of the same equivalence class.

Thanks – comments?

1. Interest in adopting this work to progress in ICNRG?
2. Any guidance as to which mechanism seems more promising?