

CCNx Selector Based Discovery draft-mosko-icnrg-selectors-01

Marc Mosko

ICNRG Interim July 21, 2019

EITF 105

Montreal, Canada

Outline

- What are selectors
- CCNx 1.0 Selector operation
- Alternatives and path forward

What are selectors

- Discovering content based on a name prefix
- Designed to operate similarly to original CCNx 0.x selectors
- Discovery operates on top of CCNx layer 3:
 - Only participating nodes need respond to a Selector query.
 - Intermediate nodes forward it as normal, oblivious to the Selector query.
 - Any node may verify that an encapsulated response satisfies the Selector query but looking deep into the response.
- Remainder of presentation will hit some of the highlights from the draft.

Encoding & Mechanics

- Appended to the end of an Interest Name
 - /NAME=example.com/NAME=widget/SELECTOR=<TLV-Encoded Query>
- The TLV-Encoded Query could be 0-length if using all default values
 - /NAME=example.com/NAME=widget/SELECTOR=<NULL>
- The response is an Encapsulated Content Object
 - The outer CO has name matching selector Interest.
 - The outer PayloadType is “Encapsulated”.
 - The outer cache control applies to the response.
 - The inner CO satisfies the Selector query.
 - If the Interest has a KeyIdRestriction, it matches the *outer* content object, as normal. This means the responding agent signs or MACs the response.

The Seven Selectors

- **(Min | Max)SuffixComponents:** the minimum/maximum number of additional name segments a matching Content Object must have, excluding the Interest's SELECTOR. The default value is 0/unlimited.
- **ChildSelector:** Answer with the left-most or right-most child.
- **NameExcludes:** A set of range and singleton exclusions to eliminate Content Objects. The exclusions match against the name segment that would immediately follow the Interest name prefix up to but not including the SELECTOR name segment.
- **InnerKeyId:** Matches the KeyId of the encapsulated object.
- **HashExcludes:** A list of ContentObjectHashRestrictions to exclude.
- **SelectorNonce:** A number to make the query unique.

Exclusion Encoding

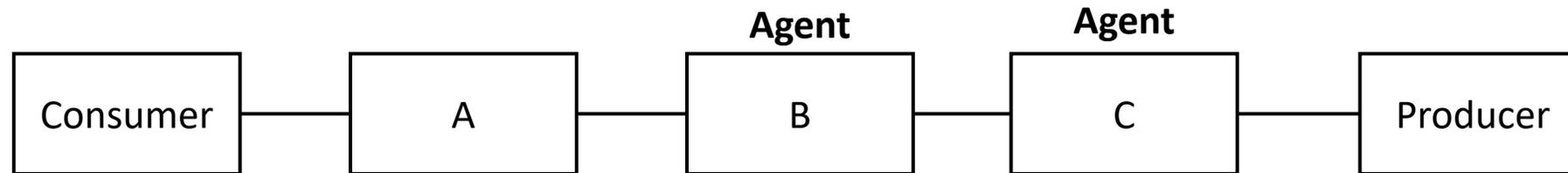
- Name component exclusions differ from the original CCNB format
 - Must be in sorted order (normal shortlex convention).
 - It is the whole Name Component TLV, not just the Value.
 - The exclusion includes the Name Component TLV Type.
 - T_EX_SINGLE: Excludes a single name component.
 - T_EX_RANGE: Excludes half open range beginning at this value and up to but not including the next T_EX_SINGLE or an implicit infinity.
 - CCNx 1.0 does not have an implicit hash component at the end of the name.

Examples of Exclusions

Pattern* (S=Single, R=Range)	Meaning
S[ace] R[bat]	NAME=ace, [NAME=bat, infty)
R[ace] S[bat] S[bat]	[NAME=ace, NAME=bat], <i>inclusive range</i>
R[CHUNK=0] S[CHUNK=20]	[CHUNK=0, CHUNK=20)
R[] S[ace]	(-infty, NAME=ace), <i>excludes any preceding TLV types</i>
R[NAME=] S[ace]	[NAME=, NAME=ace)
R[]	(-infty, +infty)
S[zoo] S[ape]	Invalid range, not sorted
R[NAME=ace] S[CHUNK=0]	[NAME=ace, CHUNK=0), <i>this will span TLV ranges type between T_NAMESEGMENT and T_CHUNK</i>
R[CHUNK=] S[CHUNK+1=]	[CHUNK=, CHUNK+1=), <i>excludes all CHUNK TLV</i>
S[apple] S[banana] R[can] S[dan] R[ed] S[fromage]	NAME=apple, NAME=banana, [NAME=can, NAME=dan), [NAME=ed, NAME=fromage)

* Actual patterns use numeric values, not symbolic name component types

Discussion on a Topology



- Node A does normal caching
 - Selector NONCE used to bypass such caches.
- If node B cannot satisfy a Selector query, it continues to C, etc.
- Node B may verify Selector responses from node C.
- Consumer must verify Selector response.

Comments on Selector Discovery

- Encapsulation:
 - You cannot encapsulate very large objects because you run out of packet size.
 - We recommend only caching Discovery responses briefly, if they cannot be verified, to avoid consumers needing to use a Nonce to skip cache entries.
- Nonce:
 - These are not like the old Interest nonce. A node does not need to track them. They are used only to make a request unique.
- One-by-one discovery:
 - Like the original selector discovery, it is one-by-one returning the whole object.
- What does multiple discovery mean?
 - `/NAME=example.com/NAME=widget/SELECTOR=<QueryA>/SELECTOR=<QueryB>`

Alternatives

- Discovery more like a search engine.
 - Submit a query and get a set of results, not the objects themselves.
 - You can cursor over the results.
 - You talk with a specific Agent for a consistent set of results.
 - For example, you send a plain query and get a response and in it is the name to use to keep talking about the response.
 - Better yet, you discover an agent for a namespace and then CCNxKE with it for privacy.
- Do not use Excludes.
 - If you need to Exclude, you asked the wrong question.
- More expressive queries than operating at a single name component node.
- Distinguish between authoritative/blessed agents and opportunistic?

Discussion