I2RS requirement discussion

WG LC has Closed

- draft-ietf-i2rs-pub-sub-requirements
  - Discussion at the interim discussed the high reliability of notification streams versus perfect reliability
- http://datatracker.ietf.org/doc/draft-ietf-i2rs-traceability/ (WG LC complete, consensus)
  - Traceability decided to add priority to its stream

WG feedback on draft-haas-i2rs-ephemeral-state-reqs

- Overlay model
  - Juergen suggested overlay was best
  - WG chairs need a draft providing details on this overlay model any further progression
- Storing priority associated with NACM must be with client
  - Suggestion was made to store with
  - Group will not work because group picks ID with highest level of role

Andy Bierman on Notifications Control Loop

- IMO NETCONF notifications are probably not the best choice for the high-performance low-latency signalling that I2RS wants between agent and client. There is a lot of overhead in the filtering, replay, YANG schema processing, XML encoding, etc.
-
- Binary notifications that contain hard-wired standard messages specially written to implement the I2RS protocol would be much faster and less expensive to implement. If this is to be as fast as a signalling protocol then it would be wise to consider protocol performance issues.

Additional Comments on Control Loop

- I have some concerns about the tight notification control loop that is proposed. IMO, this is going to be too slow and too complicated. It seems to me that the only company that has implemented something close to I2RS is using a design that does not rely on a near real-time reliable notification loop.

Error stop option – Addition

- For perform all or none, when one operations in the message causes all things go to original state
  - Rollback is implementation detail for failure
- For the stop-on-error, when one operation in the message causes an error, that operation is not done (can't b/c of error) AND no operations after that in the message are done.
- For recording errors, all operations in the message are attempted in order and any errors are recorded to send back to the client.
- If an operation caused an error, then the operation isn't completed.

10 Requirements (1)

1. The I2RS protocol SHOULD support highly reliable notifications (but not perfectly reliable notifications) from an I2RS agent to an I2RS client.

2. The I2RS protocol SHOULD support a high bandwidth, asynchronous interface, with real-time guarantees on getting data from an I2RS agent by an I2RS client.

3. The I2RS protocol will operate on data models which may be protocol independent or protocol dependent.

4. I2RS Agent needs to record the client identity when a node is created or modified. The I2RS Agent needs to be able to read the client identity of a node and use the client identity's associated priority to resolve conflicts. The secondary identity is useful for traceability and may also be recorded.

10 Requirements (2)

5. Client identity will have only one priority for the client identity. A collision on writes is considered an error, but priority is utilized to compare requests from two different clients in order to modify an existing node entry. Only an entry from a client which is higher priority can modify an existing entry (First entry wins). Priority only has meaning at the time of use.

6. The Agent identity and the Client identity should be passed outside of the I2RS protocol in a authentication and authorization protocol (AAA). Client priority may be passed in the AAA protocol. The values of identities are originally set by operators, and not standardized.

7. An I2RS Client and I2RS Agent mutually authenticate each other based on pre-established authenticated identities.

8. Secondary identity data is read-only meta-data that is recorded by the I2RS agent associated with a data model's node is written, updated or deleted. Just like the primary identity, the secondary identity is only recorded when the data node is written or updated or deleted.

10 Requirements (30

9. I2RS agent can have a lower priority I2RS client attempting to modify a higher priority client's entry in a data model. The filtering out of lower priority clients attempting to write or modify a higher priority client's entry in a data model SHOULD be effectively handled and not put an
undue strain on the I2RS agent.

Note: Jeff's suggests that priority is kept at the NACM at the client level (rather than the path level or the group level) will allow these lower priority clients to be filtered out using an extended NACM approach. This is only a suggestion of a method to provide the requirement 9.

10. The I2RS protocol MUST support the use of a secure transport. However, certain functions such as notifications MAY use a non-secure transport. Each model or service (notification, logging) must define within the model or service the valid uses of a non-secure transport.

Did I miss others?

- Please comment on others I missed