RESTCONF and CoMI

Thing to Thing PRG
IETF #92 Dallas, TX, USA

Andy Bierman <andy@yumaworks.com>

v0.4

Agenda

- RESTCONF Protocol
- Constrained Management Interface (CoMI)

Starting Point: NETCONF

- Network Configuration Protocol (RFC 6241)
  - allows all-or-none transaction and rollback-on-error support, using YANG data models
  - Problems for constrained environments
    - SSH session support
    - XML message encoding
    - multi-message operations

Next step: RESTCONF

- RESTCONF Protocol (work-in-progress)
  - Provides REST-like API to developers to access YANG data in NETCONF datastores
    - GET for retrieval
    - POST, PUT, DELETE, PATCH for resource editing
    - SSE for notifications
  - Problems for constrained environments
    - XML encoding mandatory (JSON is optional)
    - TCP transport
    - SSE requires long-lived HTTP/TCP connection

Final step: CoMI

- Constrained Management Interface (work-in-progress)
  - Large subset of the resource management features of RESTCONF
    - Uses CoAP/DTLS/UDP instead of HTTP/SSL/TCP
    - YANG data nodes identified with 30-bit hash instead of long XPath path expression
    - CBOR binary encoding instead of XML or JSON
    - CoAP Observe-based notifications instead of SSE

Hash-based IDs vs. Path Exprs

- Want a short permanent identifier for objects
  - Hash done on canonical XPath expression of object
  - Instance identified with object ID [+ key-leafs]
  - Collisions within a server are possible, not across servers
    - 38581 objects before 50% probability of collision
- Module set ID used to trigger hash check
  - YANG library identifies all modules used (could be from a CFG server)
  - Client can pre-compute hashes for modules
  - Module set not expected to change often (if at all)
- Client will discover any rehashed objects from each server
  - Adds complexity, even if though is unlikely a client will need to use it
- Possibly useful outside of network management?

YANG to CBOR

- YANG uses hierarchical data and presents some encoding issues
  - Sometimes verbose (e.g., string representation for 'enumeration' and 'bits' data types)
  - QNames in content (identityref, XPath, etc.)
- Servers want to stream data to a client
  - Document representation not stored in the server
  - Distributed content and access control make “next node” unknown to server
  - CBOR supports streaming of data in this manner, using indefinite length arrays

Observe Based Notifications

- System events vs. Resource monitoring
  - Client needs to be informed of interesting events, not attempt to track every state change in every resource instance
  - RMON alarms/events might work better for thresholds
  - Observe designed for sensors
    - System state changes may be handled better with data-model specific event notifications
  - May need aggregation mechanism to prevent flooding
    - What if lots of server resources change state at once?

Importance of YANG

- YANG Data Modeling Language (RFC 6020)
  - Syntax/semantics should be defined out-of-band
    - Minimize meta-data on the wire
    - Maximize interoperability with detailed schema
  - Automation tools work better with YANG
    - Many machine-readable constraints
    - Extensions allow tool-specific directives
    - Common code in the stack increases consistency and reduces the size and complexity of data-model instrumentation code