

**Lesson 04**  
**Messaging Protocols for Connected  
Devices:**  
**Message cache, Publish/Subscribe (Pub/Sub) and  
Message queues**

# Request/Response (Client/Server) based messaging Protocol

- An object (client) requests for a resource(s)
- Another end object (server) sends the response
- Both client and server using REST functions.
- Request adds the header words.

# Pub/Sub (Publish/Subscribe) based messaging Protocol

- An object (server) publishes the a resource(s) for the clients
- Another end object (client) receives the resource on subscription
- Numbers of clients can subscribe to a published resource, for example, weather report or location information

# Pub/Sub messaging protocol

- A separate subscription required for each resource-type or topic
- Provisions for publication of messages and their reception on subscription

# Pub/Sub messaging protocol

- PUT method used by Pub and GET Method by the registered or authenticated devices.
- Publisher registers or deregisters a device for a resource type

# Example

- Resource type: measured values of ambient light condition in smart streetlights example
- Another resource type: traffic presence or absence on the street.
- Another resource type, a lighting function report (proper or faulty) in the light

# Resource Discovery

- Resource discovery service may advertise (publish) at regular intervals, the availability of the resources or types of the resources available and their states.
- A client discovers the resource type and registers for the RD service.

# Registration

- *Registration* means a receiver registers with a service, for example, a resource directory service (RD).
- When one or more end-points or devices or nodes registers, then that gets the access to the resources and receives published messages.



# Registration

- Security considerations may require authentication of both ends (service provider and receiver) before registration.
- A separate registration required for each end-point (client or server)

# Registration Update

- Updating registrations for one or more end-points or devices or nodes
- Also includes unregistering for one or more end-points.

# Pull (Subscribe/Notify) Messages or Data

- *Pull* means pulling a resource value or message or data of a resource-type by registering and subscribing.
- Pull (Subscribe/Notify) Data Pull may be using GET or on initiating OBSERVE.

# Pull Messaging Method

- The server maintains state information for a resource and notifies on change of state
- Client pulls again the resource on the change.

# Polling or Observing

- Finding from where new messages available
- Finding whether new messages available
- Finding updates available
- Finding whether or not a need for refresh of information
- Finding the state information changed or same.

# Polling or Observing

- A Polling method client uses REST architecture GET method and server uses POST method.
- A state may mean connection, or disconnection, sleep, awake, created, alive (not deleted), old values persisting or updated with new values.. (OBSERVE method)

# Polling or Observing

- GET + OBSERVE method
- Observing means looking for change, if any, of a state at periodic intervals

# Push (Publish/Subscribe) Data

- Means a services (Server S/W) pushes using PUT ( ) the messages or information regularly
- Interested device or endpoint or potential receiver receives the pushes
- For example, a mobile service pushes the temperature and location information regularly for registered subscribers.



# Message Cache

- Cache means storing when available, for use later on when required
- Useful in environment of short or prolonged disconnections of a service
- A message accessed once or more times from a cache

# Network layer 2

- Transport and Network capabilities
- (For example, Connectivity layer in CISCO Reference Architecture)

# Services and Application Support layer 3

- Generic and Specific support capabilities
- [For example, Data abstraction, Accumulation, Elements Analysis and Transformation (CISCO Reference Architecture)]

# Message Queue

- Stores in sequences the messages (data) from devices or end-points
- When sought or when connection state changes then forwarding the messages
- Forwarding is in first-in first-out methods for a resource-type .

# Message Queue

- A message forwards once only from a queue.
- Separate queue forms for each resource-type
- The messages forward to the registered devices or end points and to the subscribed devices or end points.

# Message Queue

- A separate registered devices or end points list and a separate subscription list maintained and used for each resource type
- Forwarding takes place after matching the subscription from a list

# Information/Query

- A method is that object (client) requests information using query and another end object (server) responds by reply to the query
- A responding Application processes the query using query optimizer and the retrieval plan for the database or resources directory and resources

# CoAP-MQ

- A message queue protocol using a broker and resource directory (RD)
- CoAP end-points role as the client and server





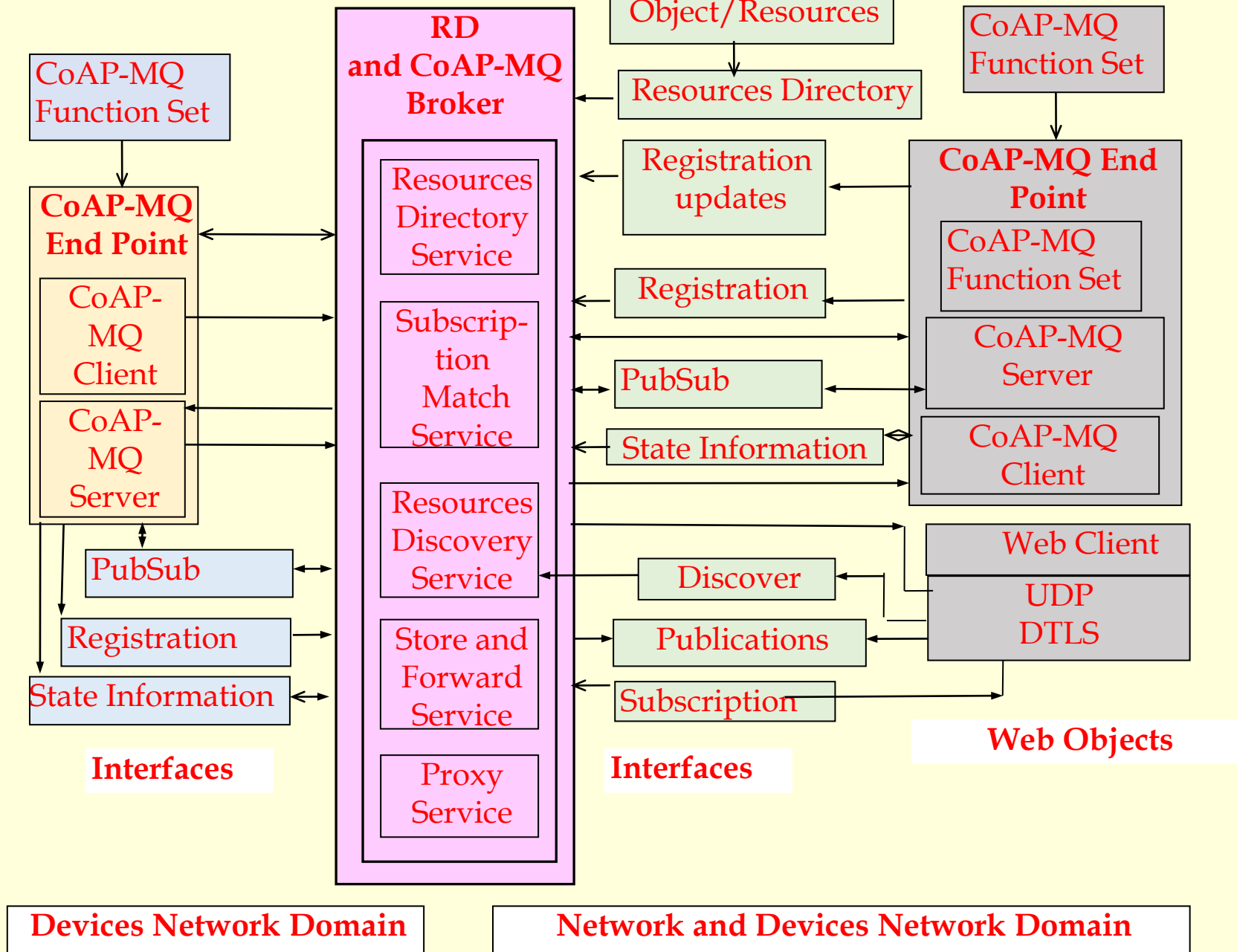


Fig. 3.5 Data interchanges between CoAP-MQ End Points, CoAP-MQ Clients, CoAP-MQ Servers through CoAP-MQ Broker and its services. [PubSub means publication to RD and subscription to MQ.]

# CoAP-MQ-Service

- Sending CoAP messages of one endpoint to another,
- queuing of messages (store) by intermediate node (s), and
- forwarding only when it suits for example, when the message recipient endpoint is awake (not sleeping) or connected and alive

# CoAP-MQ End point Functions

- Implements the functions at the CoAP-MQ function set,
- serves as CoAP-MQ client and CoAP-MQ server,
- Register with RD server for using Broker, (RD server advertises a service)

# CoAP-MQ End point Functions

- Receives advertisements from Broker which may advertise service, and
- May permit implementation of sleeping end points and message queuing for receiving on awaking of end-point

# CoAP-MQ Broker Functions

- Functions as a server node capable of storing messages to and from other nodes
- Performs a store-and-forward function between web-clients and the CoAP-MQ capable endpoints,

# CoAP-MQ Broker Functions

- Matches subscriptions and publications in order to route messages to right end-points,
- Sends state of an end point when end-point or a web client subscribes to the state of the endpoint,

# CoAP-MQ Broker Functions

- Enables the web client publishing of updates to the endpoint state through the CoAP-MQ broker,
- Returns the last published value to web clients or other endpoints on behalf of endpoints that are sleeping, and
- Acts as a proxy



# Summary

We learnt

- Request/Response Messaging
- Pub/Sub Messaging
- Polling/Observe Messaging
- Pull Messaging
- Push Messaging
- Message Cache

# Summary

We learnt

- Message Queue using a separate registered devices or end points list and a separate subscription list maintained and used for each resource type
- Information/Query based Messaging
- CoAP-MQ Service, End Point and Broker Functions

**End of Lesson 4 on  
Messaging Protocols for Connected Devices:  
Message cache, Publish/Subscribe (Pub/Sub) and  
Message queues**