UNIT III

Constrained Application Protocol (CoAP)

Background

Messaging Model

Request/Response Model

Intermediaries and Caching

Constrained Application Protocol (CoAP) Background

CoAP is a simple application layer protocol targeted to simple electronic devices (e.g., IoT/M2M things) to allow them to communicate interactively over the Internet. CoAP is designed for low power sensors (wireless sensor network [WSN] nodes and actuators. CoAP can be seen as a specialized web transfer protocol for use with constrained networks and nodes for M2M applications. CoAP operates with HTTP (hypertext transfer protocol) for basic support with the web

CoAP protocol are as follows:(i) minimal complexity for the mapping with HTTP;(ii) low header overhead and low parsing complexity;(iii) support for the discovery of resources;(iv) simple resource subscription process;(v) simple caching based on max-age.

CoAP makes use of two message types, requests and responses, using a simple binary base header format. Any bytes after the headers in the packet are considered the message body if any. The length of the message body is implied by the datagram length.

The constrained nodes for which CoAP is targeted often have 8-bit microcontrollers with small amounts of ROM and RAM, while networks such as 6LoWPAN (IPv6 OVER LOWPOWER WPAN)CoAP provides a method/response interaction model between application end-points, supports built-in resource discovery, and includes key web concepts such as URIs (uniform resource identifiers) and content-types. CoAP easily translates to HTTP for integration with the web.

The use of Web Services (WS) on the Internet has become ubiquitous in most applications; it depends on the fundamental representational state transfer (REST) architecture of the web.

CoAP has the following main features: Constrained web protocol fulfilling M2M requirements; UDP (User datagram protocol) binding with optional reliability supporting unicast and multicast requests; Asynchronous message exchanges; Low header overhead and parsing complexity; URI and content-type support; Simple proxy and caching capabilities; A stateless HTTP mapping, allowing proxies to be built providing access to

CoAP resources via HTTP in a uniform way or for HTTP simple interfaces to be realized alternatively over CoAP Security binding to datagram transport layer security (DTLS).

M2M interactions typically result in a CoAP implementation acting in both client and server roles (called an end-point).A CoAP request is equivalent to that of HTTP and is sent by a client to request an action (using a method code) on a resource (identified by a URI) on a server. The server then sends a response with a response code; this response may include a resource representation. Unlike HTTP, CoAP deals with these interchanges asynchronously over a datagram- oriented transport such as UDP. This is done logically using a layer of messages that supports optional reliability (with exponential back-off).CoAP defines four types of messages: confirmable (CON), non-confirmable (NON), acknowledgement, reset; Method codes and response codes included in some of these messages make them carry requests or responses. The basic exchanges of the four types of messages are transparent to the request/response interactions.

CoAP logically as using a two-layer approach, a CoAP messaging layer used to deal with UDP (User Datagram Protocol)the asynchronous nature of the interactions, the request/response interactions using method and response codes

CoAP is, however, a single protocol, with messaging and request/response just features of the CoAP header. Figure depicts the overall protocol stack that is being considered in the CoAP context.

Constrained Application Protocol (CoAP) Messaging Model
The CoAP messaging model is based on the exchange of messages over UDP between end- points. It uses a short fixed-length binary header (4 bytes) that may be followed by compact binary options and a payload. This message format is shared by requests and responses. Each CoAP message contains a message ID used to detect duplicates and for optional reliability.

Reliability is provided by marking a message as CON.A CON message is retransmitted using a default timeout and exponential back-off between retransmissions, until the recipient sends an acknowledgement message (ACK) with the same message ID from the corresponding end-point. When a recipient is not able to process a CON message, it replies with a reset message (RST) instead of an ACK.A message that does not require reliable delivery, for example, each single measurement out of a stream of sensor data, can be sent as a NON message. These are not acknowledged, but still have a message ID for duplicate detection. When a recipient is not able to process a NON message, it may reply with an RST. Since CoAP is based on UDP, it also supports the use of multicast IP destination addresses, enabling multicast CoAP requests.

Constrained Application Protocol (CoAP) Request/Response Model

CoAP messages, which include either a method code or response code, respectively. Optional (or default) request and response information, such as the URI (uniform resource identifier) and payload content-type, are carried as CoAP options. A token option is used to match responses to requests independent of the underlying messages.

A request is carried in a CON (confirmable) or NON (non- confirmable) message, and if immediately available, the response to a request carried in a CON message is carried in the resulting ACK message. This is called a piggy-backed response. If the server is not able to respond immediately to a request carried in a CON message, it simply responds with an empty ACK message so that the client can stop retransmitting the request. When the response is ready, the server sends it in a new CON message (which then in turn needs to be acknowledged by the client). This is called a separate response. Likewise, if a request is sent in a NON message, then the response is usually sent using a new NON message, although the server may send a CON message. CoAP makes use of GET, PUT, POST, and DELETE methods in a similar manner to HTTP.

Constrained Application Protocol (CoAP) Intermediaries and Caching

The protocol supports the caching of responses in order to efficiently fulfill requests. Simple caching is enabled using freshness and validity information carried with CoAP responses. A cache could be located in an end-point or an intermediary.

Proxying is useful in constrained networks for several reasons, including (i) network traffic limiting,(ii) to improve performance,(iii) to access resources of sleeping devices,(iv) for security reasons. The proxying of requests on behalf of another CoAP end-point is supported in the protocol. The URI of the resource to request is included in the request, while the destination IP address is set to the proxy.