IoT Protocols

IoT Lecture 4

Rahul Shandilya

Transport Layer

The transport layer protocols provide end-to-end message transfer capability independent of the underlying network. In the application layer, the application has produced an initial message within its own protocol. There are four things that must take place after that which is carried out by transport layer.

Transport Layer

The transport layer protocols provide end-to-end message transfer capability independent of the underlying network. In the application layer, the application has produced an initial message within its own protocol. There are four things that must take place after that which is carried out by transport layer.

- A connection must be established between the source and destination machines using handshake.

Transport Layer

The transport layer protocols provide end-to-end message transfer capability independent of the underlying network. In the application layer, the application has produced an initial message within its own protocol. There are four things that must take place after that which is carried out by transport layer.

- A connection must be established between the source and destination machines using handshake.
- The second role of the transport layer is to create chunks of data, known as packets (or datagrams). There are two types TCP and UDP.

Transport Layer

The transport layer protocols provide end-to-end message transfer capability independent of the underlying network. In the application layer, the application has produced an initial message within its own protocol. There are four things that must take place after that which is carried out by transport layer.

- A connection must be established between the source and destination machines using handshake.
- The second role of the transport layer is to create chunks of data, known as packets (or datagrams). There are two types TCP and UDP.
- The transport layer is also responsible for flow control and multiplexing. In flow control, it is possible that the transport layer can postpone the transmission of packets when network congestion is high. It can also request that the destination computer respond at a slower rate if processing is slow.

Transport Layer

The transport layer protocols provide end-to-end message transfer capability independent of the underlying network. In the application layer, the application has produced an initial message within its own protocol. There are four things that must take place after that which is carried out by transport layer.

- A connection must be established between the source and destination machines using handshake.
- The second role of the transport layer is to create chunks of data, known as packets (or datagrams). There are two types TCP and UDP.
- The transport layer is also responsible for flow control and multiplexing. In flow control, it is possible that the transport layer can postpone the transmission of packets when network congestion is high. It can also request that the destination computer respond at a slower rate if processing is slow.
- The final responsibility of the transport layer is to maintain the connection already established. A connection is typically left open while packets and acknowledgments are traded back and forth. Closing the connection usually occurs by an explicit command from the source resource, indicating that the session should close, or because of a timeout.

TCP Segment Header Format

Bit #	0 – 7	8 – 15	16 – 23	24 – 31
0	Source Port		Destination Port	
32	Sequence Number			
64	Acknowledgement Number			
96	Data Offset	Res	Flags	Window Size
128	Header and Data Checksum		Urgent Pointer	
160...	Options			

UDP Datagram Header Format

Bit #	0 – 7	8 – 15	16 – 23	24 – 31
0	Source Port		Destination Port	
32	Length		Header and Data Checksum	

Transmission Control Protocol (TCP)

TCP creates virtual circuits between hosts and guarantees features such as reliability, flow control, congestion avoidance, and multiplexing Transmission Control Protocol (TCP) is the most widely used transport layer protocol, that is used by web browsers (along with HTTP, HTTPS application layer protocols), email programs (SMTP application layer protocol) and file transfer (FTP).

Transmission Control Protocol (TCP)

TCP creates virtual circuits between hosts and guarantees features such as reliability, flow control, congestion avoidance, and multiplexing Transmission Control Protocol (TCP) is the most widely used transport layer protocol, that is used by web browsers (along with HTTP, HTTPS application layer protocols), email programs (SMTP application layer protocol) and file transfer (FTP).

- TCP is a connection oriented and stateful protocol.

Transmission Control Protocol (TCP)

TCP creates virtual circuits between hosts and guarantees features such as reliability, flow control, congestion avoidance, and multiplexing Transmission Control Protocol (TCP) is the most widely used transport layer protocol, that is used by web browsers (along with HTTP, HTTPS application layer protocols), email programs (SMTP application layer protocol) and file transfer (FTP).

- TCP is a connection oriented and stateful protocol.
- While IP protocol deals with sending packets, TCP ensures reliable transmission of packets in-order.

Transmission Control Protocol (TCP)

TCP creates virtual circuits between hosts and guarantees features such as reliability, flow control, congestion avoidance, and multiplexing Transmission Control Protocol (TCP) is the most widely used transport layer protocol, that is used by web browsers (along with HTTP, HTTPS application layer protocols), email programs (SMTP application layer protocol) and file transfer (FTP).

- TCP is a connection oriented and stateful protocol.
- While IP protocol deals with sending packets, TCP ensures reliable transmission of packets in-order.
- TCP also provides error detection capability so that duplicate packets can be discarded and lost packets are retransmitted.

Transmission Control Protocol (TCP)

TCP creates virtual circuits between hosts and guarantees features such as reliability, flow control, congestion avoidance, and multiplexing Transmission Control Protocol (TCP) is the most widely used transport layer protocol, that is used by web browsers (along with HTTP, HTTPS application layer protocols), email programs (SMTP application layer protocol) and file transfer (FTP).

- TCP is a connection oriented and stateful protocol.
- While IP protocol deals with sending packets, TCP ensures reliable transmission of packets in-order.
- TCP also provides error detection capability so that duplicate packets can be discarded and lost packets are retransmitted.
- The flow control capability of TCP ensures that rate at which the sender sends the data is not too high for the receiver to process.

Transmission Control Protocol (TCP)

TCP creates virtual circuits between hosts and guarantees features such as reliability, flow control, congestion avoidance, and multiplexing Transmission Control Protocol (TCP) is the most widely used transport layer protocol, that is used by web browsers (along with HTTP, HTTPS application layer protocols), email programs (SMTP application layer protocol) and file transfer (FTP).

- TCP is a connection oriented and stateful protocol.
- While IP protocol deals with sending packets, TCP ensures reliable transmission of packets in-order.
- TCP also provides error detection capability so that duplicate packets can be discarded and lost packets are retransmitted.
- The flow control capability of TCP ensures that rate at which the sender sends the data is not too high for the receiver to process.
- The congestion control capability of TCP helps in avoiding network congestion and congestion collapse which can lead to degradation of network performance.

User Datagram Protocol (UDP)

UDP is a best-effort protocol that allows packet transmissions with minimum reliability. UDP is useful for time-sensitive applications that have very small data units to exchange and do not want the overhead of connection setup. Owing to its simplicity of implementation, it is a very attractive option for IoT scenarios.

User Datagram Protocol (UDP)

UDP is a best-effort protocol that allows packet transmissions with minimum reliability. UDP is useful for time-sensitive applications that have very small data units to exchange and do not want the overhead of connection setup. Owing to its simplicity of implementation, it is a very attractive option for IoT scenarios.

- UDP is a connectionless protocol

User Datagram Protocol (UDP)

UDP is a best-effort protocol that allows packet transmissions with minimum reliability. UDP is useful for time-sensitive applications that have very small data units to exchange and do not want the overhead of connection setup. Owing to its simplicity of implementation, it is a very attractive option for IoT scenarios.

- UDP is a connectionless protocol
- UDP is a transaction oriented and stateless protocol.

User Datagram Protocol (UDP)

UDP is a best-effort protocol that allows packet transmissions with minimum reliability. UDP is useful for time-sensitive applications that have very small data units to exchange and do not want the overhead of connection setup. Owing to its simplicity of implementation, it is a very attractive option for IoT scenarios.

- UDP is a connectionless protocol
- UDP is a transaction oriented and stateless protocol.
- UDP is a best-effort protocol that allows packet transmissions with minimum reliability.

User Datagram Protocol (UDP)

UDP is a best-effort protocol that allows packet transmissions with minimum reliability. UDP is useful for time-sensitive applications that have very small data units to exchange and do not want the overhead of connection setup. Owing to its simplicity of implementation, it is a very attractive option for IoT scenarios.

- UDP is a connectionless protocol
- UDP is a transaction oriented and stateless protocol.
- UDP is a best-effort protocol that allows packet transmissions with minimum reliability.
- UDP is “lightweight" - there is no ordering of messages, no tracking connections, and so on. In other words, UDP does not have an option for flow control.

Application Layer

Application layer protocols define how the applications interface with the lower layer protocols to send the data over the network. The application data, typically in files, is encoded by the application layer protocol and encapsulated in the transport layer protocol which provides connection or transaction oriented communication over the network. Port numbers are used for application addressing (for example port 80 for HTTP, port 22 for SSH, etc.). Application layer protocols enable process-to-process connections using ports.

Hypertext Transfer Protocol (HTTP)

Hypertext Transfer Protocol (HTTP) is the application layer protocol that forms the foundation of the World Wide Web (WWW). HTTP includes commands such as GET, PUT, POST, DELETE, HEAD, TRACE, OPTIONS, etc.

Hypertext Transfer Protocol (HTTP)

Hypertext Transfer Protocol (HTTP) is the application layer protocol that forms the foundation of the World Wide Web (WWW). HTTP includes commands such as GET, PUT, POST, DELETE, HEAD, TRACE, OPTIONS, etc.

- The protocol follows a request-response model where a client sends requests to a server using the HTTP commands.

Hypertext Transfer Protocol (HTTP)

Hypertext Transfer Protocol (HTTP) is the application layer protocol that forms the foundation of the World Wide Web (WWW). HTTP includes commands such as GET, PUT, POST, DELETE, HEAD, TRACE, OPTIONS, etc.

- The protocol follows a request-response model where a client sends requests to a server using the HTTP commands.
- HTTP is a stateless protocol and each HTTP request is independent of the other requests.

Hypertext Transfer Protocol (HTTP)

Hypertext Transfer Protocol (HTTP) is the application layer protocol that forms the foundation of the World Wide Web (WWW). HTTP includes commands such as GET, PUT, POST, DELETE, HEAD, TRACE, OPTIONS, etc.

- The protocol follows a request-response model where a client sends requests to a server using the HTTP commands.
- HTTP is a stateless protocol and each HTTP request is independent of the other requests.
- An HTTP client can be a browser or an application running on the client (e.g., an application running on an IoT device, a mobile application or other software).

Hypertext Transfer Protocol (HTTP)

Hypertext Transfer Protocol (HTTP) is the application layer protocol that forms the foundation of the World Wide Web (WWW). HTTP includes commands such as GET, PUT, POST, DELETE, HEAD, TRACE, OPTIONS, etc.

- The protocol follows a request-response model where a client sends requests to a server using the HTTP commands.
- HTTP is a stateless protocol and each HTTP request is independent of the other requests.
- An HTTP client can be a browser or an application running on the client (e.g., an application running on an IoT device, a mobile application or other software).
- HTTP protocol uses Universal Resource Identifiers (URIs) to identify HTTP resources.

Constrained Application Protocol (CoAP)

Constrained Application Protocol (CoAP) is an application layer protocol for machine-to-machine (M2M) applications, meant for constrained environments with constrained devices and constrained networks.

Constrained Application Protocol (CoAP)

Constrained Application Protocol (CoAP) is an application layer protocol for machine-to-machine (M2M) applications, meant for constrained environments with constrained devices and constrained networks.

- CoAP is a web transfer protocol and uses a request-response model, however it runs on top of UDP instead of TCP.

Constrained Application Protocol (CoAP)

Constrained Application Protocol (CoAP) is an application layer protocol for machine-to-machine (M2M) applications, meant for constrained environments with constrained devices and constrained networks.

- CoAP is a web transfer protocol and uses a request-response model, however it runs on top of UDP instead of TCP.
- CoAP uses a client-server architecture where clients communicate with servers using connectionless datagrams.

Constrained Application Protocol (CoAP)

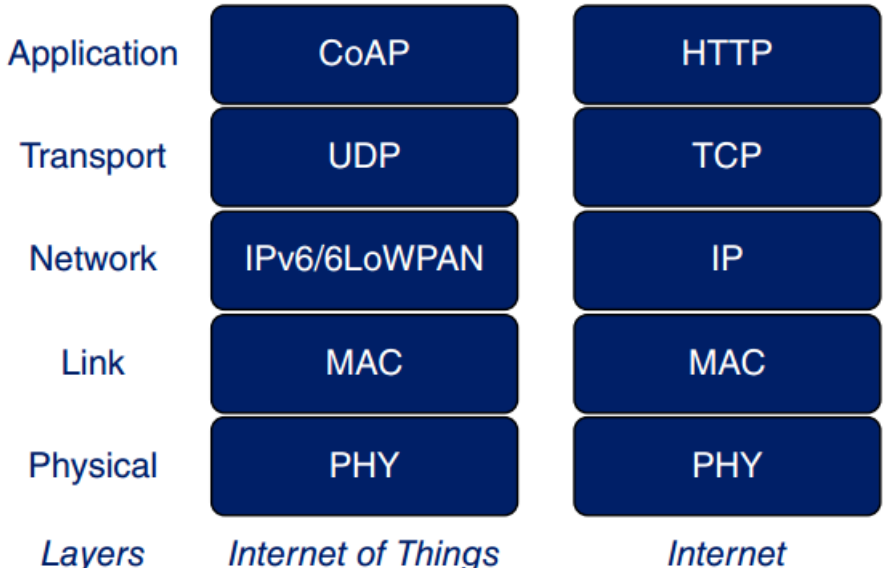
Constrained Application Protocol (CoAP) is an application layer protocol for machine-to-machine (M2M) applications, meant for constrained environments with constrained devices and constrained networks.

- CoAP is a web transfer protocol and uses a request-response model, however it runs on top of UDP instead of TCP.
- CoAP uses a client-server architecture where clients communicate with servers using connectionless datagrams.
- CoAP is designed to easily interface with HTTP. Like HTTP, CoAP supports methods such as GET, PUT, POST, and DELETE.

Constrained Application Protocol (CoAP)

Constrained Application Protocol (CoAP) is an application layer protocol for machine-to-machine (M2M) applications, meant for constrained environments with constrained devices and constrained networks.

- CoAP is a web transfer protocol and uses a request-response model, however it runs on top of UDP instead of TCP.
- CoAP uses a client-server architecture where clients communicate with servers using connectionless datagrams.
- CoAP is designed to easily interface with HTTP. Like HTTP, CoAP supports methods such as GET, PUT, POST, and DELETE.
- Asynchronous message exchanges, Support for URI and content-types


Application
Transport
Network
Link
Physical
Layers
CoAP
UDP
IPv6/6LoWPAN
MAC
PHY
Internet of Things
HTTP
TCP
IP
MAC
PHY
Internet

Message Queue Telemetry Transport (MQTT)

MQTT stands for MQ Telemetry Transport. It is a publish/subscribe, extremely simple and lightweight messaging protocol, designed for constrained devices and low-bandwidth, high-latency or unreliable networks. The design principles are to minimise network bandwidth and device resource requirements whilst also attempting to ensure reliability and some degree of assurance of delivery.

Message Queue Telemetry Transport (MQTT)

MQTT stands for MQ Telemetry Transport. It is a publish/subscribe, extremely simple and lightweight messaging protocol, designed for constrained devices and low-bandwidth, high-latency or unreliable networks. The design principles are to minimise network bandwidth and device resource requirements whilst also attempting to ensure reliability and some degree of assurance of delivery.

- Message Queue Telemetry Transport (MQTT) is a light-weight messaging protocol based on the publish-subscribe model.

Message Queue Telemetry Transport (MQTT)

MQTT stands for MQ Telemetry Transport. It is a publish/subscribe, extremely simple and lightweight messaging protocol, designed for constrained devices and low-bandwidth, high-latency or unreliable networks. The design principles are to minimise network bandwidth and device resource requirements whilst also attempting to ensure reliability and some degree of assurance of delivery.

- Message Queue Telemetry Transport (MQTT) is a light-weight messaging protocol based on the publish-subscribe model.
- MQTT uses a client-server architecture where the client (such as an IoT device) connects to the server (also called MQTT Broker) and publishes messages to topics on the server.

Message Queue Telemetry Transport (MQTT)

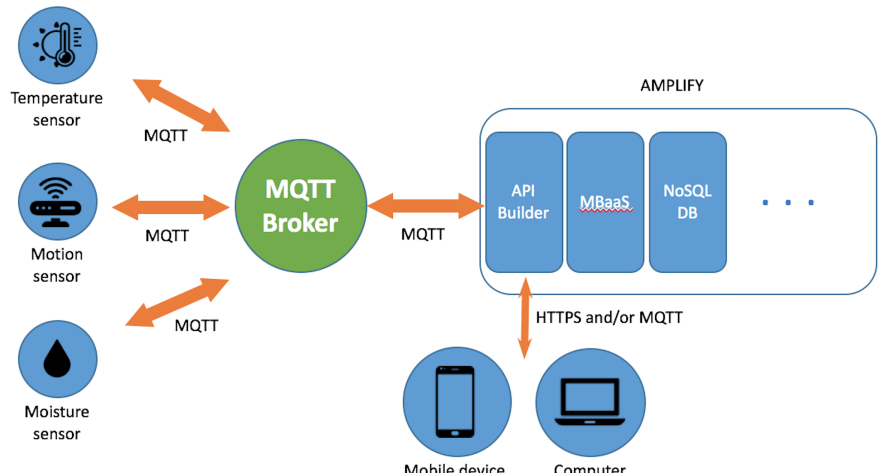
MQTT stands for MQ Telemetry Transport. It is a publish/subscribe, extremely simple and lightweight messaging protocol, designed for constrained devices and low-bandwidth, high-latency or unreliable networks. The design principles are to minimise network bandwidth and device resource requirements whilst also attempting to ensure reliability and some degree of assurance of delivery.

- Message Queue Telemetry Transport (MQTT) is a light-weight messaging protocol based on the publish-subscribe model.
- MQTT uses a client-server architecture where the client (such as an IoT device) connects to the server (also called MQTT Broker) and publishes messages to topics on the server.
- The broker forwards the messages to the clients subscribed to topics.

Message Queue Telemetry Transport (MQTT)

MQTT stands for MQ Telemetry Transport. It is a publish/subscribe, extremely simple and lightweight messaging protocol, designed for constrained devices and low-bandwidth, high-latency or unreliable networks. The design principles are to minimise network bandwidth and device resource requirements whilst also attempting to ensure reliability and some degree of assurance of delivery.

- Message Queue Telemetry Transport (MQTT) is a light-weight messaging protocol based on the publish-subscribe model.
- MQTT uses a client-server architecture where the client (such as an IoT device) connects to the server (also called MQTT Broker) and publishes messages to topics on the server.
- The broker forwards the messages to the clients subscribed to topics.
- MQTT is well suited for constrained environments where the devices have limited processing and memory resources and the network bandwidth is low.


Temperature sensor
MQTT
Motion sensor
MQTT
Moisture sensor
MQTT
MQTT Broker
MQTT
AMPLIFY
API Builder
MBaaS
NoSQL DB
HTTPS and/or MQTT
Mobile device
Computer

WebSocket

WebSocket protocol allows full-duplex communication over a single socket connection for sending messages between client and server. WebSocket is based on TCP and allows streams of messages to be sent back and forth between the client and server while keeping the TCP connection open. The client can be a browser, a mobile application or an IoT device.

Extensible Messaging and Presence Protocol (XMPP)

- Extensible Messaging and Presence Protocol (XMPP) is a real-time communication and streaming XML data between network entities.

Extensible Messaging and Presence Protocol (XMPP)

- Extensible Messaging and Presence Protocol (XMPP) is a real-time communication and streaming XML data between network entities.
- XMPP powers wide range of applications including messaging, presence, data syndication, gaming, multi-party chat and voice/video calls.

Extensible Messaging and Presence Protocol (XMPP)

- Extensible Messaging and Presence Protocol (XMPP) is a real-time communication and streaming XML data between network entities.
- XMPP powers wide range of applications including messaging, presence, data syndication, gaming, multi-party chat and voice/video calls.
- XMPP allows sending small chunks of XML data from one network entity to another in near real-time.

Extensible Messaging and Presence Protocol (XMPP)

- Extensible Messaging and Presence Protocol (XMPP) is a real-time communication and streaming XML data between network entities.
- XMPP powers wide range of applications including messaging, presence, data syndication, gaming, multi-party chat and voice/video calls.
- XMPP allows sending small chunks of XML data from one network entity to another in near real-time.
- XMPP is a decentralized protocol and uses a client-server architecture. XMPP supports both client-to-server and server-to-server communication paths.

Extensible Messaging and Presence Protocol (XMPP)

- Extensible Messaging and Presence Protocol (XMPP) is a real-time communication and streaming XML data between network entities.
- XMPP powers wide range of applications including messaging, presence, data syndication, gaming, multi-party chat and voice/video calls.
- XMPP allows sending small chunks of XML data from one network entity to another in near real-time.
- XMPP is a decentralized protocol and uses a client-server architecture. XMPP supports both client-to-server and server-to-server communication paths.
- In the context of IoT, XMPP allows real-time communication between IoT devices.

Data Distribution Service (DDS)

- Data Distribution Service (DDS) is a data-centric middleware device-to-device or machine-to-machine communication.

Data Distribution Service (DDS)

- Data Distribution Service (DDS) is a data-centric middleware device-to-device or machine-to-machine communication.
- DDS uses a publish-subscribe model where publishers (e.g. devices that generate data) create topics to which subscribers (e.g., devices that want to consume data) can subscribe.

Data Distribution Service (DDS)

- Data Distribution Service (DDS) is a data-centric middleware device-to-device or machine-to-machine communication.
- DDS uses a publish-subscribe model where publishers (e.g. devices that generate data) create topics to which subscribers (e.g., devices that want to consume data) can subscribe.
- Publisher is an object responsible for data distribution and the subscriber is responsible for receiving published data.

Data Distribution Service (DDS)

- Data Distribution Service (DDS) is a data-centric middleware device-to-device or machine-to-machine communication.
- DDS uses a publish-subscribe model where publishers (e.g. devices that generate data) create topics to which subscribers (e.g., devices that want to consume data) can subscribe.
- Publisher is an object responsible for data distribution and the subscriber is responsible for receiving published data.
- DDS provides quality-of-service (QoS) control and configurable reliability.

Advanced Message Queuing Protocol (AMQP)

- Advanced Message Queuing Protocol (AMQP) is an open application layer protocol for business messaging.

Advanced Message Queuing Protocol (AMQP)

- Advanced Message Queuing Protocol (AMQP) is an open application layer protocol for business messaging.
- AMQP supports both point-to-point and publisher/subscriber models, routing and queuing.

Advanced Message Queuing Protocol (AMQP)

- Advanced Message Queuing Protocol (AMQP) is an open application layer protocol for business messaging.
- AMQP supports both point-to-point and publisher/subscriber models, routing and queuing.
- AMQP brokers receive messages from publishers (e.g., devices or applications that generate data) and route them over connections to consumers (applications that process data).

Advanced Message Queuing Protocol (AMQP)

- Advanced Message Queuing Protocol (AMQP) is an open application layer protocol for business messaging.
- AMQP supports both point-to-point and publisher/subscriber models, routing and queuing.
- AMQP brokers receive messages from publishers (e.g., devices or applications that generate data) and route them over connections to consumers (applications that process data).
- Publishers publish the messages to exchanges which then distribute message copies to queues.

Advanced Message Queuing Protocol (AMQP)

- Advanced Message Queuing Protocol (AMQP) is an open application layer protocol for business messaging.
- AMQP supports both point-to-point and publisher/subscriber models, routing and queuing.
- AMQP brokers receive messages from publishers (e.g., devices or applications that generate data) and route them over connections to consumers (applications that process data).
- Publishers publish the messages to exchanges which then distribute message copies to queues.
- Messages are either delivered by the broker to the consumers which have subscribed to the queues or the consumers can pull the messages from the queues.