Internet of Things (IoT) Connectivity Technologies

Kayhan Zrar Ghafoor (Post doctorate Fellow)

Department of Software Eng., College of Engineering,

Salahaddin Univeristy-Erbil, Kurdistan, Iraq

Functionality based IoT Protocol Organization

- Connectivity (6LowPAN)
- Communication/ Transport (WiFi, Bluetooth)
- Application Protocols (MQTT, CoAP)

IoT Ecosystem

Applications	Smart health, smart home, smart grid, smart transportation, smart workspace	Security	Management
Session	MQTT		IEEE 1905
Routing	6LowPAN,RPL		IEEE 1451
Data link	WiFi, Bluetooth Smart, ZigBee Smart, Z-Wave, DECT/ULE, 3G/LTE, NFC, Weightless, HomePlug GP, 802.11ah, 802.15.4, G.9959, WirelessHART, DASH7, ANT+ , LoRaWAN		
Operating Systems	Linux, Android, Contiki-OS, TinyOS		
Hardware	Arduino, Raspberry Pi, ARC-EM4, Mote, Smart Dust, Tmote Sky		

MQ Telemetry Transport (MQTT)

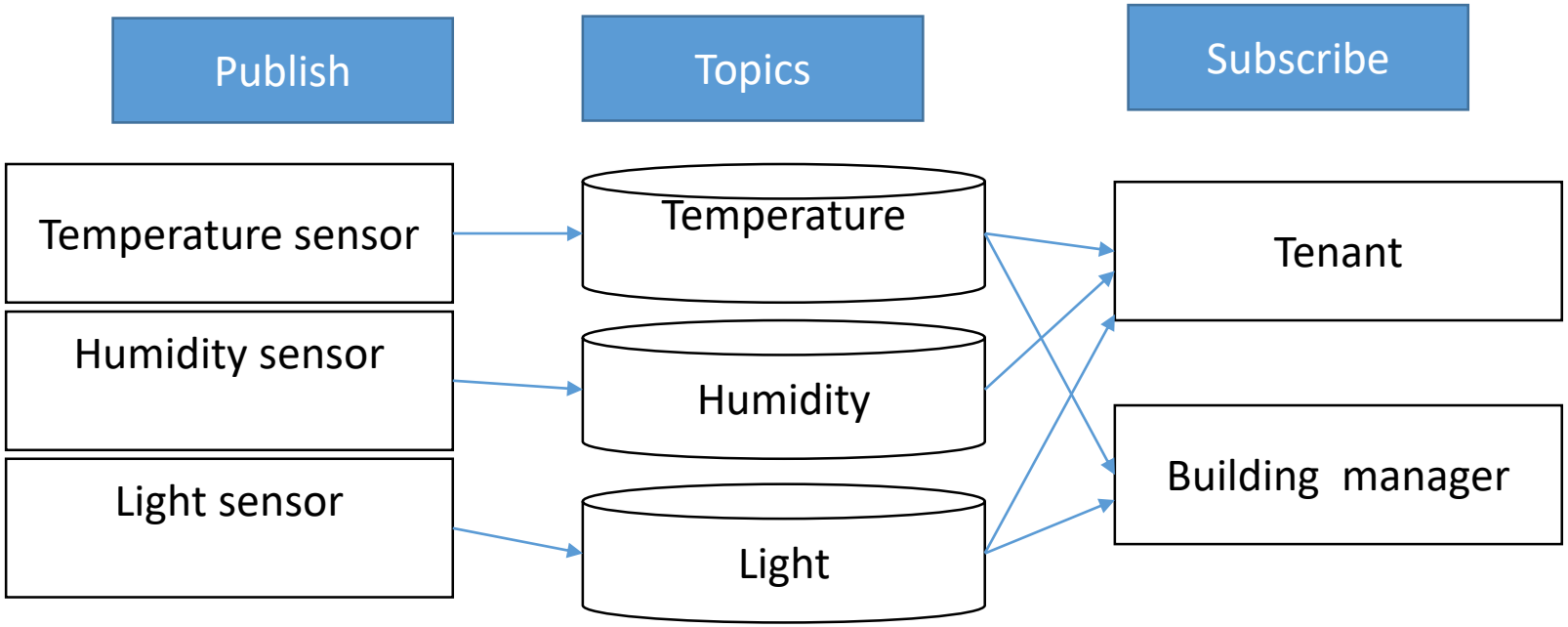
- Lightweight messaging protocol for M2M communication
- Telemetry = Tele-Metering = Remote measurements
- Invented and sponsored by IBM. Now Open source. Open Source libraries available.
- MQ originated from “message queueing (MQ)” architecture used by IBM for service oriented networks. There is no queueing in MQTT.
- Telemetry data goes from devices to a server or broker. Uses a publish/subscribe mechanism.
- Lightweight = Low network bandwidth and small code footprint.

MQTT (Cont)

- Facebook messenger uses MQTT to minimize battery usage. Several other applications in medical, environmental applications
- Many open source implementations of clients and brokers are available
- Really small message broker (RSMB): C
- Mosquitto
- Micro broker: Java based for PDAs, notebooks

MQTT Concepts

- Topics/Subscriptions: Messages are published to topics. Clients can subscribe to a topic or a set of related topics
- Publish/Subscribe: Clients can subscribe to topics or publish to topics.

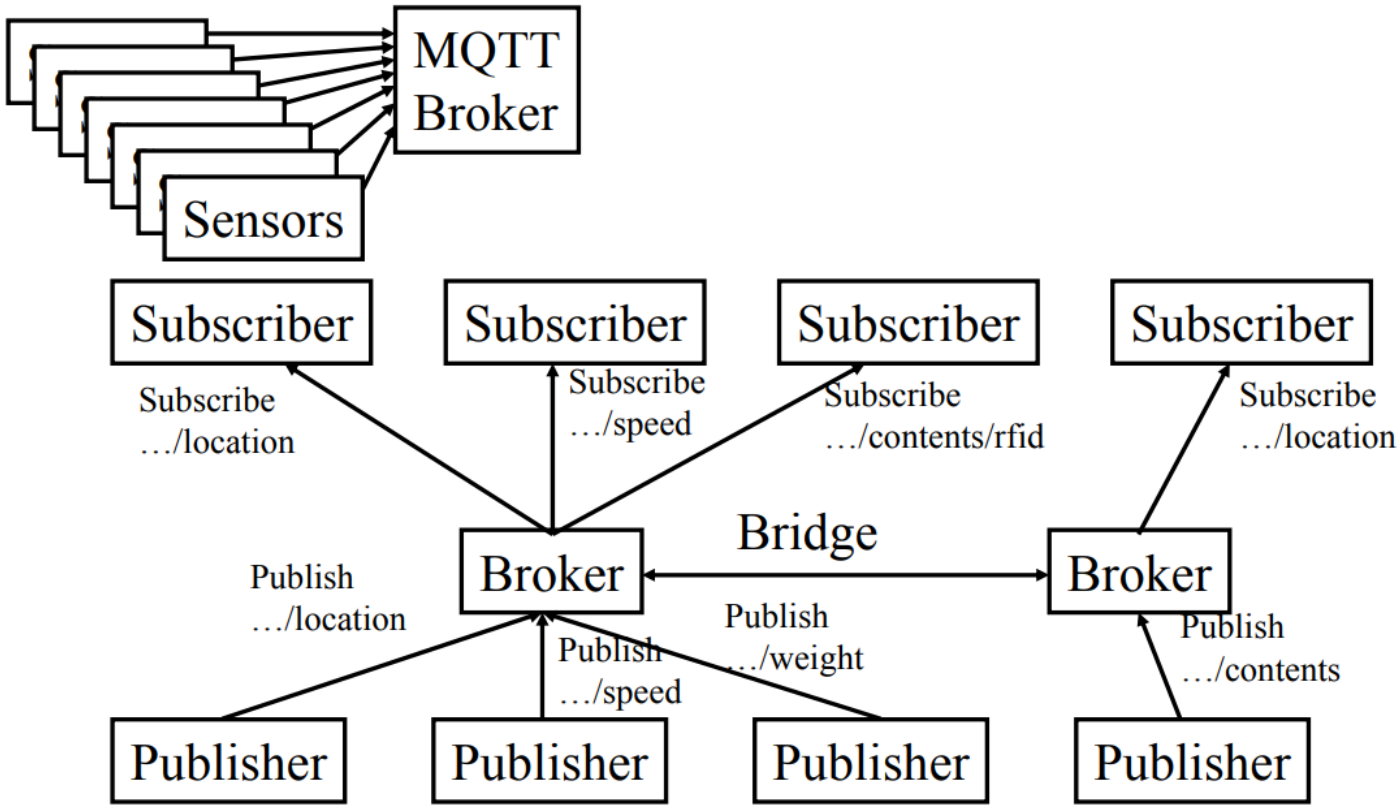


MQTT Concepts (Cont)

- Quality of Service Levels: Three levels:
- 0 = At most once (Best effort, No Ack),
- 1 = At least once (Acked, retransmitted if ack not received),
- 2 = Exactly once [Request to send (Publish), Clear-to-send (Pubrec), message (Pubrel), ack (Pubcomp)]
- Retained Messages: Server keeps messages even after sending it to all subscribers. New subscribers get the retained messages

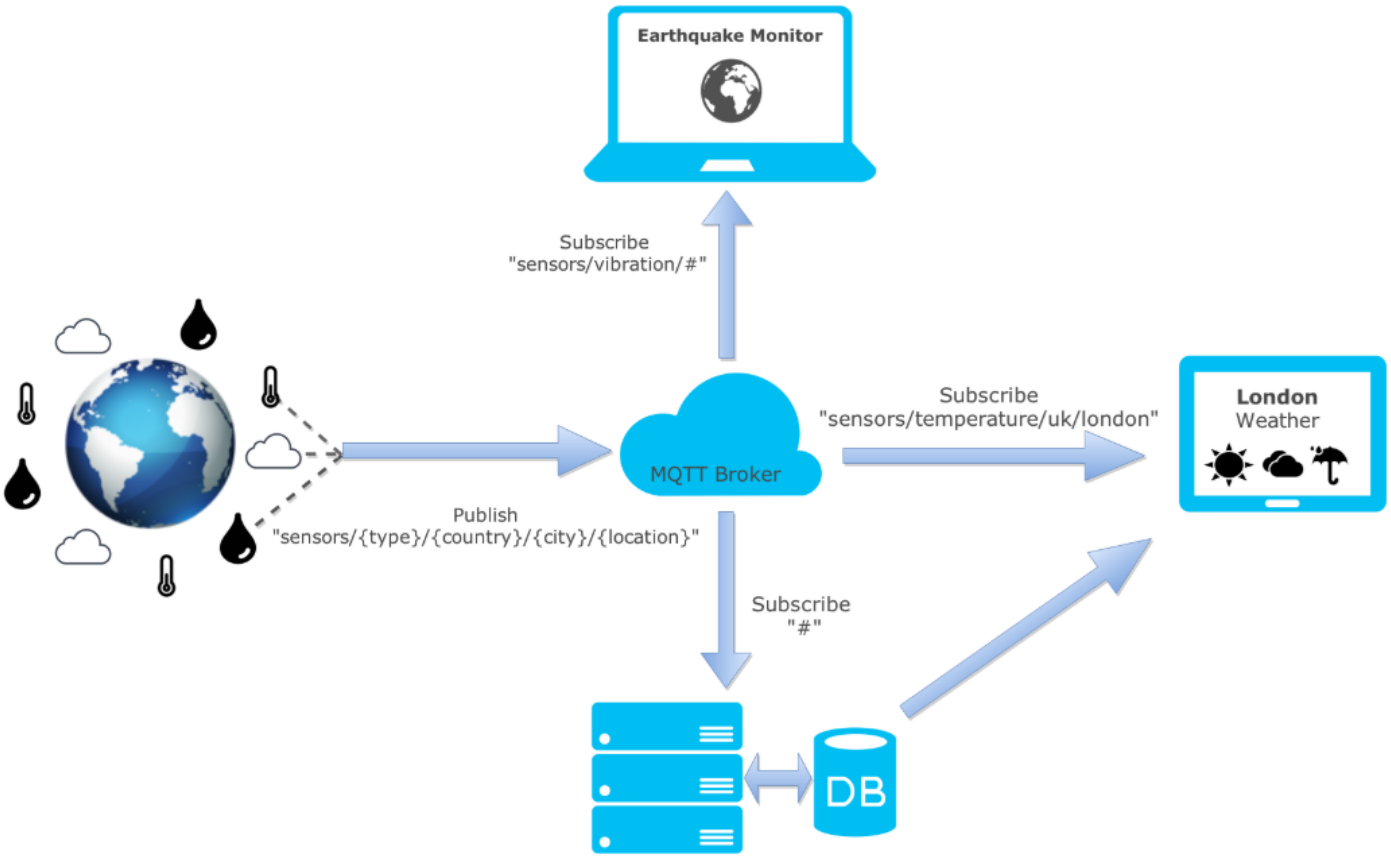
MQTT Concepts (Cont)

- Clean Sessions and Durable Connections: At connection set up: Clean session flag
- With a **non persistent connection (**Clean Sessions**)** the broker **doesn't store** any subscription information or undelivered messages for the client.
- In the mode of Durable Connections, the broker **will store** subscription information, and undelivered messages for the client.
- Wills: At connection a client can inform that it has a will or a message that should be published if unexpected disconnection, Alarm if the client looses connection
- Periodic keep alive messages, If a client is still alive
- Topic Trees: Topics are organized as trees using / character /# matches all sublevels /+ matches only one sublevel



MQTT example

MQTT example



Ref: https://zoetrope.io/tech-blog/brief-practical-introduction-mqtt-protocol-and-its-application-iot/

MQTT example

- weather service which has a network of internet connected temperature sensors all over the world.
- They publish to a particular topic based on their location in the following format:

sensors/temperature/{country}/{city}/{street name}

- So a sensor on Baker Street in London would publish to ‘sensors/temperature/uk/london/baker_street’ with a message containing the current temperature.

Ref: https://zoetrope.io/tech-blog/brief-practical-introduction-mqtt-protocol-and-its-application-iot/

MQTT example

- So the global temperature database might subscribe to *sensors/temperature/#* and it would receive temperature readings from every sensor in the world.
- If however the UK government wanted to use the data for their own weather service, they could just subscribe to *sensors/temperature/uk/#*, thereby limiting the sensor readings to those within the UK.
- And if a service wanted to get data from all sensor types in a particular location, it could use something like this: *sensors/+/uk/london/baker_street*.

Ref: https://zoetrope.io/tech-blog/brief-practical-introduction-mqtt-protocol-and-its-application-iot/

MQTT & HTTP

	MQTT	HTTP
Design	Data centric	Document centric
Pattern	Publish/Subscribe	Request /Response
Complexity	Simple	More Complex
Message Size	Small. Binary with 2B header	Large. ASCII
Service Levels	Three	One
Libraries	30kB C and 100 kB Java	Large
Data Distribution	1 to zero, one, or n	1 to 1 only

Summary

- MQTT is a protocol used to publish and subscribe sensor information
- Lightweight, low code size, open source

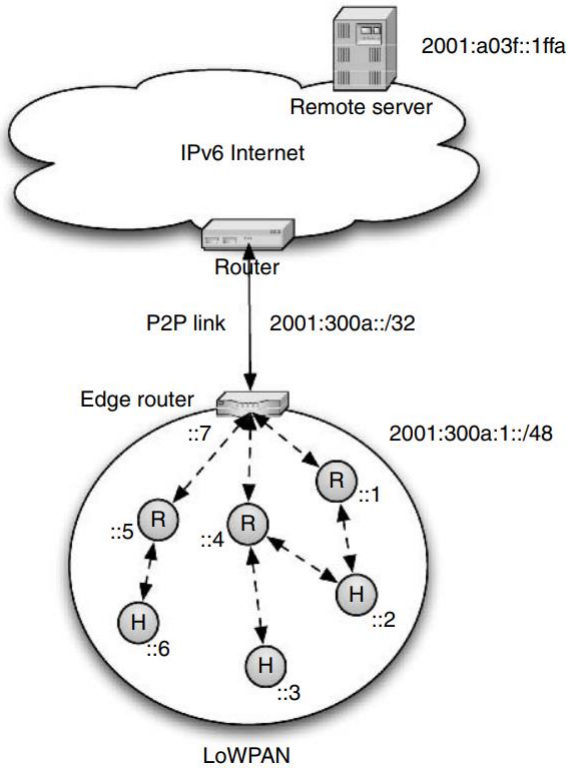
6LowPAN: An Open IoT Networking Protocol

- It stands for Low-power Wireless Personal Area Network
- Allows smallest devices with limited processing capabilities to transmit information wirelessly using an internet protocol.
- Allows low power device to connect to the internet.
- It is developed by IETF

- Open:Open Source, IoT: making objects interent-aware, Networking: stopping at layer 3

6LowPAN: An Open IoT Networking Protocol

- Edge routers help a 6LowPAN Domains to get to the internet. Edge router performs three main Tasks:
- Exchange info between 6LowPAN Devices and the internet.
- Data exchange between devices Present in the 6LowPAN domain.
- Generation and Maintenance of radio subnet.



6LowPAN: An Open IoT Networking Protocol

- IETF developed a group (6LowPAN) to enable IPv6 over IEEE 802.15.4
- The 6LowPAN protocol is primarily focused on applying IPv6 on MAC and PHY layers of IEEE 802.15.4. **Then, they faced two crucial challenges:**
- Maximum frame size of IEEE 802.15.4 is 127 bytes.
- Minimum MTU for IPv6 is 1280 bytes
- To address this issue **adaptation layer** was developed.

Adaptation Layer of 6LowPAN

- It is just above data link layer.
- One function of adaptation layer is ip packet fragmentation
- the first byte of the payload is used as a dispatch byte

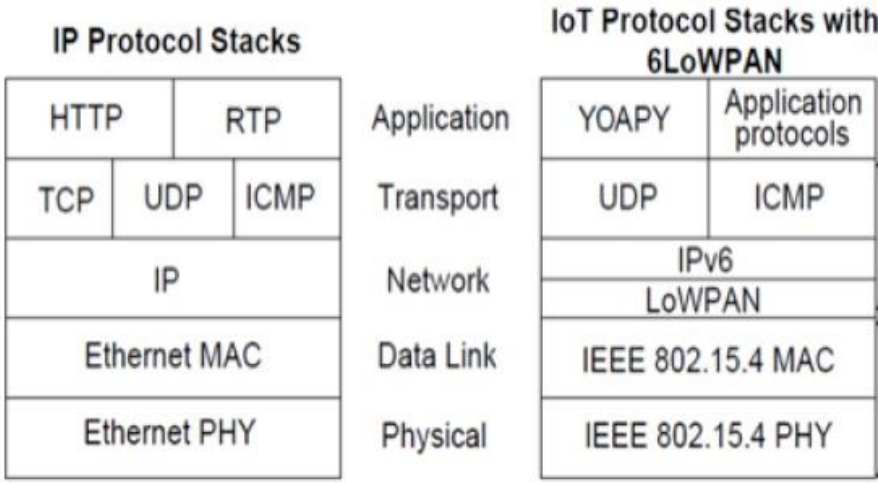


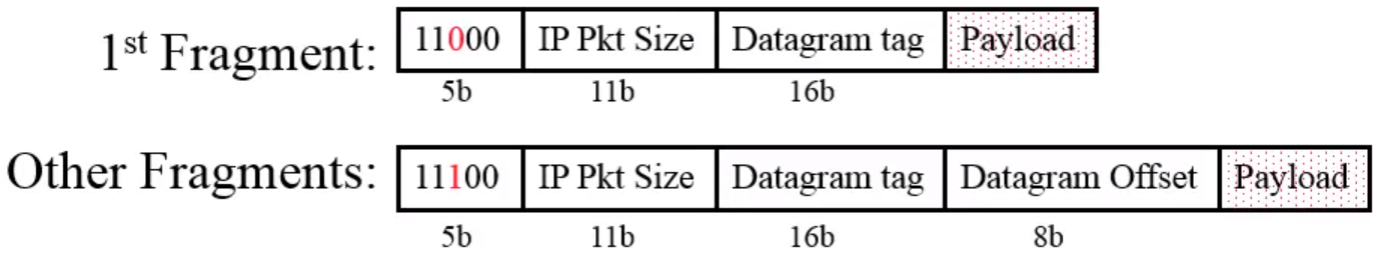
Table 2.1 The two most-significant bits in the dispatch byte.

00	Not a LoWPAN packet (NALP)
01	Normal dispatch
10	Mesh header, see Section 2.5 below
11	Fragmentation header, see Section 2.7 below

*http://keshi.ubiwna.org/2015IotComm/6LoWPAN_The%20Wireless%20Embedded%20Internet.pdf

Adaptation Layer of 6LowPAN

- If first two bit of dispatch byte is 11, it means fragment header
- Full packet size exist in fragment's fragment header
- Datagram tag with sender and destination MAC addresses: fragments of the same packet.

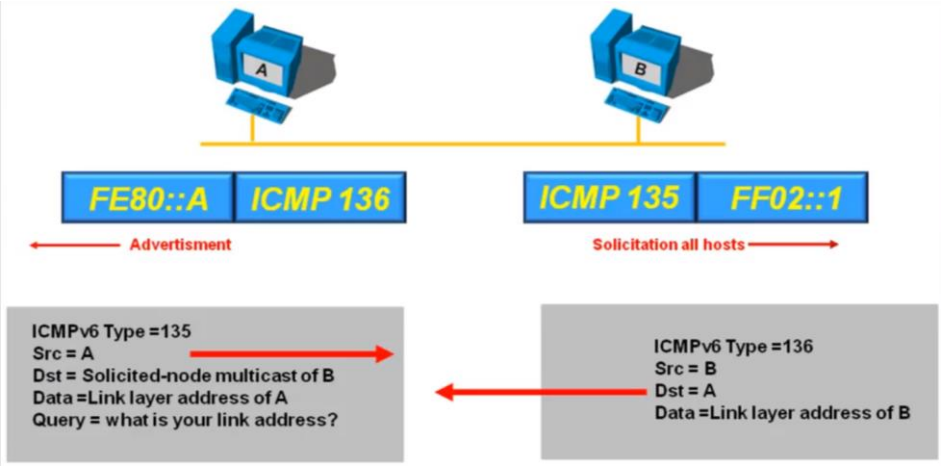


Neighbour Discovery

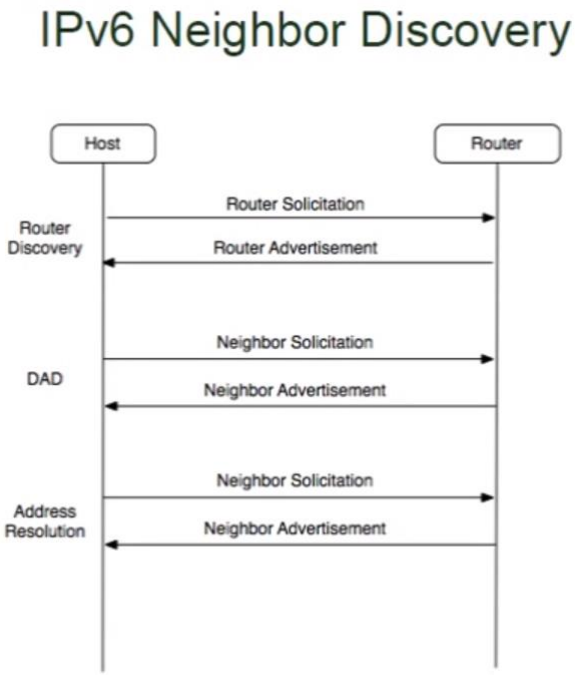
- A LoWPAN consists of nodes, which may play the role of host or router, along with one or more edge routers. The network interfaces of the nodes in a LoWPAN share the same IPv6 prefix which is distributed by the edge router and routers throughout the LoWPAN.
- In order to facilitate efficient network operation, nodes register with an edge router. These operations are part of Neighbor Discovery (ND), which is an important basic mechanism of IPv6.

Neighbour Discovery – IP6

- However, in IP6 neighbour discovery every node is shouting in the network. This scheme does not work in IoT enabled network as we have sleepy devices.



Neighbour discovery IPv6



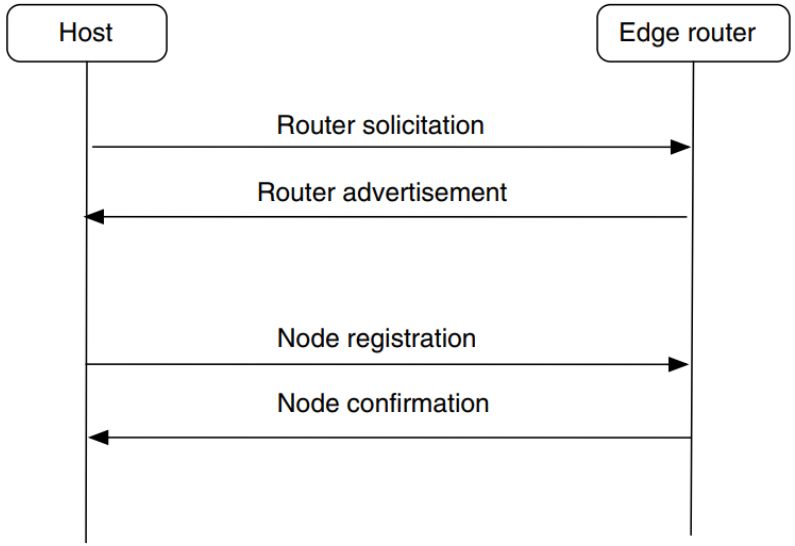
Neighbour Discovery – 6LowPAN

- 6LoWPAN-ND uses the edge routers as the focal point of Duplicate Address Detection.
- Every edge router maintains a whiteboard on which nodes can scribble their address and which other nodes can later read.
- This is done using two new ICMPv6 messages: Node Registration (NR) and Node Confirmation (NC);

Neighbour Discovery – 6LowPAN

- After obtaining a prefix as well as the address of the edge router, the host attempts to register one or more of its own addresses with the edge router by sending a Node Registration message. The edge router replies with a Node Confirmation message listing the addresses that were acceptable and includes those in its whiteboard

Neighbour Discovery – 6LowPAN



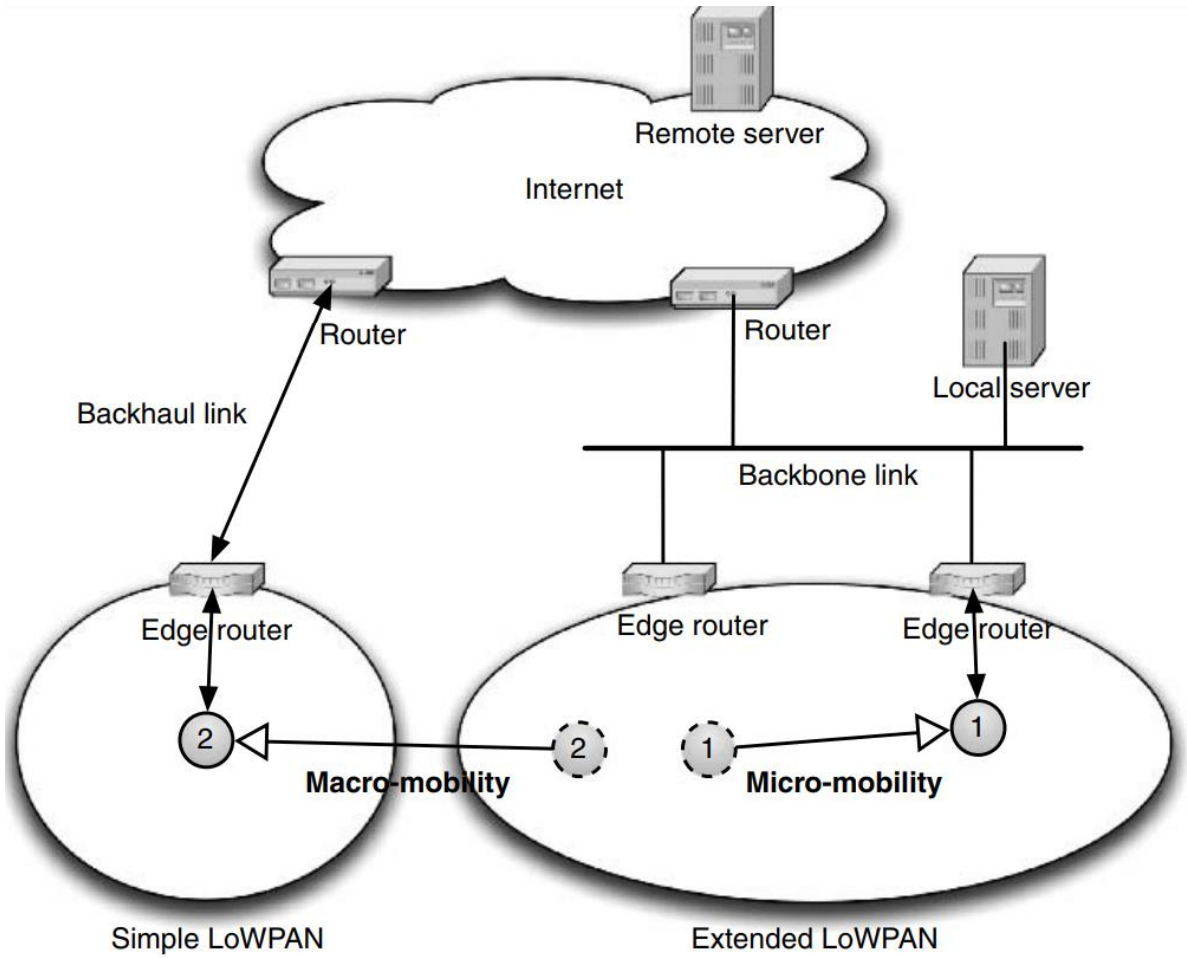
Basic router discovery and registration process with an edge router

Mobility in 6LowPAN

- Roaming: A process in which a mobile node moves from one network to another, typically with no existing packet streams.

- Handover: A process in which a mobile node disconnects from its existing point of attachment and attaches itself to a new point of attachment.

Mobility in 6LowPAN



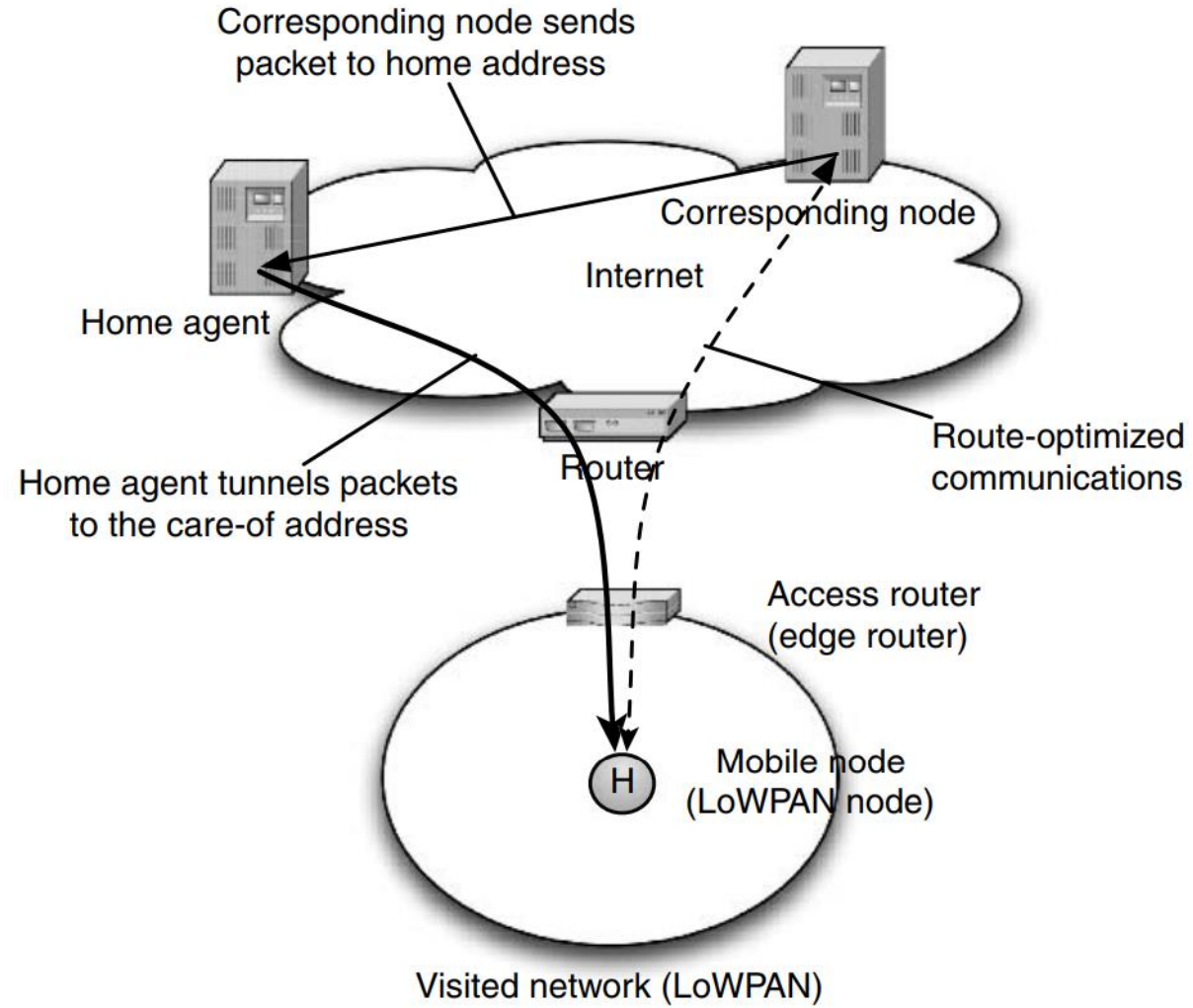
Mobility in 6LowPAN

- **Mobile IP6**
- After detecting that the subnet has changed, and that the node is no longer in its home network, it sends a MIPv6 binding update message to its HA.
- A bidirectional IPv6-in-IPv6 tunnel is then set up between the HA and mobile node for exchanging data packets. The packet is then encapsulated in another IPv6 header with the destination address set to the mobile node’s care-of address.

Mobility in 6LowPAN

- **Mobile IP6**
- After receiving and decapsulating the packet, the mobile node can then respond through the IPv6-in-IPv6 tunnel through its HA back to the corresponding node. Alternatively, the mobile node can respond directly to the corresponding node using its care-of address.
- Now communication can continue directly between the correspondent node and the mobile node.

Mobility in 6LowPAN



Mobile IPv6

The Problem of Mobile IP6 when applied in 6LowPAN

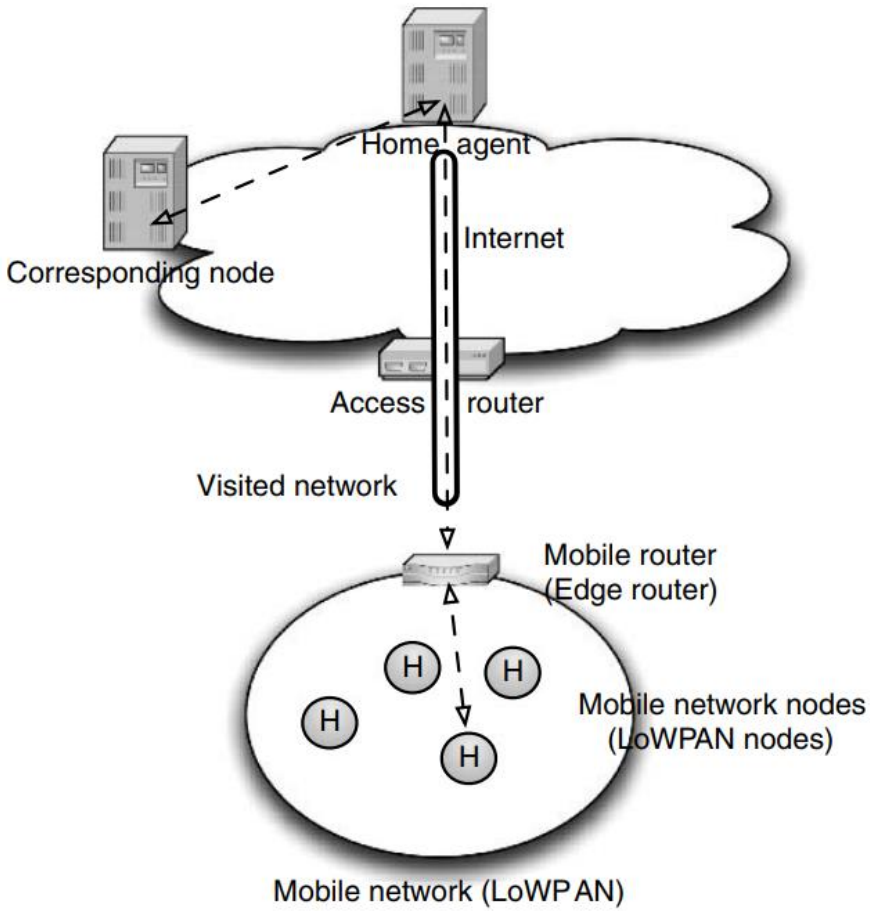
- IPv6-in-IPv6 tunneling between the HA and LoWPAN Node would incur large header overheads as the encapsulated IPv6 and transport headers cannot be compressed by the existing compression methods.
- The requirement of IPsec security associations between MIPv6 entities may be unreasonable for LoWPAN Nodes.

Network Mobility applicable to 6LowPAN

- A mobile router functions like a normal MIPv6 host setting up a bidirectional tunnel with its Home Agent, but in addition it negotiates prefixes to be forwarded to it by the Home Agent.

- The Home Agent then forwards all packets matching the bound prefix (therefore packets for the MNNs) to the mobile router.

Network Mobility applicable to 6LowPAN



Example of the basic NEMO protocol working with 6LoWPAN