The Role of IP in IoT/M2M

國立交通大學資訊工程系
Department of Computer Science
National Chiao Tung University
November 15, 2018

Outline

1. Overview - Relevant IETF WGs
2. IPv6
3. 6LoWPAN
4. CoAP
5. RPL

Overview - Relevant IETF WGs

- IP version 6 (IPv6) WG - the specification and standardization of the Internet Protocol version 6 (IPv6).
- 6LoWPAN (IPv6 for Low-Power Wireless Personal Area Network) WG – IPv6 adaptation for LoWPANs especially for 802.15.4.
- ROLL (Routing Over Low power and Lossy Networks or LLNs) WG – Routing Protocol for Low power and lossy networks (RPL) (IPv6 routing solutions for LLNs)
- CoRE (Constrained RESTful Environments) WG – Constrained Application Protocol (CoAP) (a framework for resource oriented applications intended to run on constrained IP networks such as LoWPANs)

IPv6

IPv6

- The Internet Engineering Task Force (IETF) created IPv6 (RFC 2460) as a replacement to IPv4 in 1998, when it became clear that the Internet would eventually run out of IPv4 addresses.
- IPv6 provides a much larger global address space than its predecessor IPv4.
- This enables global end-to-end communications and restores valuable properties of the IP architecture that have been lost in the IPv4 Internet.
- The core IPv6 standards are widely implemented and are starting to see global deployment.

Source: IETF IPv6 WG

Problems with IPv4

- IPv4 address exhaustion has been a concern since the 1980s, when Internet engineers came up with several techniques that would allow the Internet to continue growing while a replacement to IPv4 was developed.
- These techniques include
  - **Network Address Translation (NAT),** which allows organizations to hide private network addresses behind a single public IPv4 address, and
  - **Classless Inter-Domain Routing (CIDR),** a more scalable way of allocating and routing IP address blocks.

Address Space - IPv6 vs. IPv4

- 128 Bits vs. 32 Bits
- IPv4 address notation: expressed in d:d:d:d where d represents a byte value between 0 and 255, for example: 127.0.0.1.
- IPv6: expressed in x:x:x:x:x:x:x:x where x is a 16 bits hexadecimal field and represents four hexadecimal digits**,** for example: fe80::2a0:d2ff:fea5:e9f5

Why are there only 5 than 8 X's?

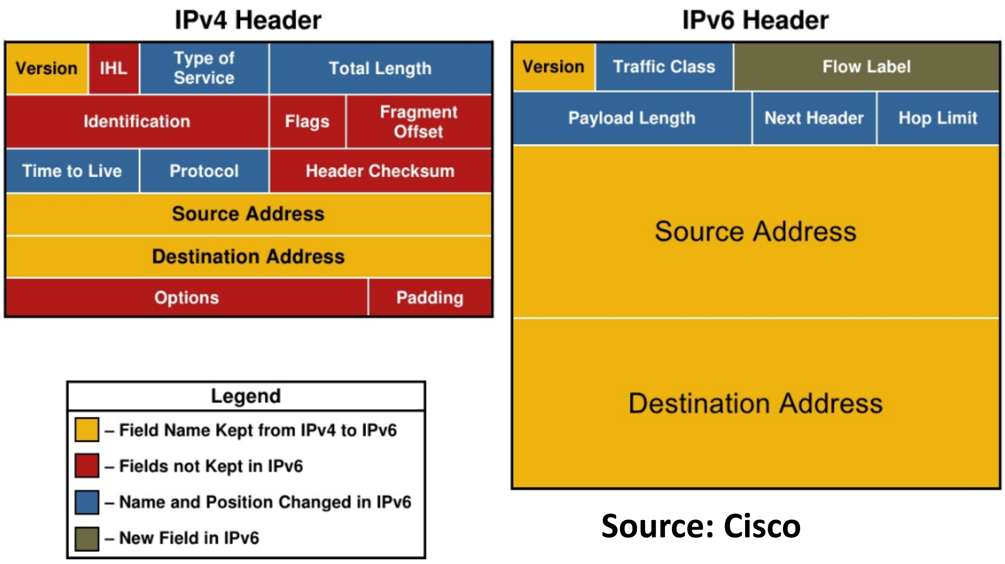
World's population is approximately 6.5 billion

7.7 Billion at 2018

$\frac{2^{128}}{6.5 \text{ Billion of 2005}}$ $10^{12}\ 10^{12}$ = 52 Trillion Trillion IPv6 addresses per person

Source: Cisco

IP Header – IPv6 vs. IPv4



Source: Cisco

IPv4 can support up to 4,294,967,296 unique addresses but in fact only about 3.7 billion addresses are assignable due to network classes and other IP address reservation for testing and multicasting etc.

Header Changes from IPv4 to IPv6

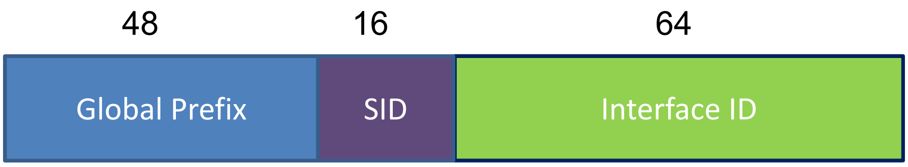
- Removed
  - Fragmentation fields moved out of base header
  - IP options moved out of base header
  - Header Checksum eliminated
  - Header Length field eliminated
  - Identification, Flags and Padding eliminated
- Revised
  - Time to Live =>Hop Limit
  - Protocol =>Next Header
  - Type of Service =>Traffic Class
  - Total Length => Payload Length (Length field excludes IPv6 header)
  - Addresses increased: 32 bits =>128 bits
- Extended
  - Flow Label field added

Abbreviation of IPv6 Addresses

- Rules
  - Leading zeros in a field are optional
  - Successive fields of 0 are represented as ::, but only once in an address
- Examples
  1. 2001:0DA8:E800:0000:0260:3EFF:FE47:0001 =>
     2001:DA8:E800:0:260:3EFF:FE47:1 =>
     2001:DA8:E800::260:3EFF:FE47:1
  2. 2001:0DA8:E800:0000:0000:0000:0000:0001 => 2001:DA8:E800::1
  3. FF02:0:0:0:0:0:0:1 => FF02::1
  4. 3FFE:0501:0008:0000:0260:97FF:FE40:EFAB =>
     3FFE:501:8:0:260:97FF:FE40:EFAB =>
     3FFE:501:8::260:97FF:FE40:EFAB
  5. 0:0:0:0:0:0:0:1 => ::1
  6. 0:0:0:0:0:0:0:0 => ::

IPv6 Address Allocation

- IPv6 Address Divided into Three Parts:



  - Global Prefix (GP)
  - Subnet Identifier (SID)
  - Interface ID (IID)

Derivation of IID

- From the MAC address (MAC-48 or EUI-64) of the interface
- Randomly Drawn
- Manually assigned with a small number
- From a certificate (CGA: Cryptographic Generated Address)

Common IPv6 Prefix

- Prefix for Link Local Address - FE80::/64 (packets inside a link only)
- Prefix for Unique Local Address – randomly generated for the site (Private Address)
- Multicast Prefixes – FF00::/8
- Special Group Prefixes –
  - FF02::1 (the group of devices connected to the sender's link)
  - FF02::2 (the group of routers connected to the sender's link)

IPv6 Address Type

- Unicast
  - One to One (Global, Link local, Site local)
  - An address destined for a single interface.
- Multicast
  - One to Many
  - An address for a set of interfaces
  - Delivered to a group of interfaces identified by that address.
  - Replaces IPv4 "broadcast"
- Anycast (new!)
  - One to Nearest (Allocated from Unicast)
  - Delivered to the closest interface as determined by the IGP

A single interface may be assigned multiple IPv6 addresses of any type (unicast, anycast, multicast).

Features – IPv6 vs. IPv4

- In addition to improvement on address space
  - Better mobility support (simplified Mobile IP)
  - Better security (IPSec is part of IPv6)
  - Support of auto configuration
  - Easier network renumbering
  - Header format helps speed processing/ forwarding
  - Header changes to facilitate QoS
- Allowing permanent assignment of an IP address thus improvement of IP performance via direct communication without intermediate routers, NATs, table look-ups or proxies.

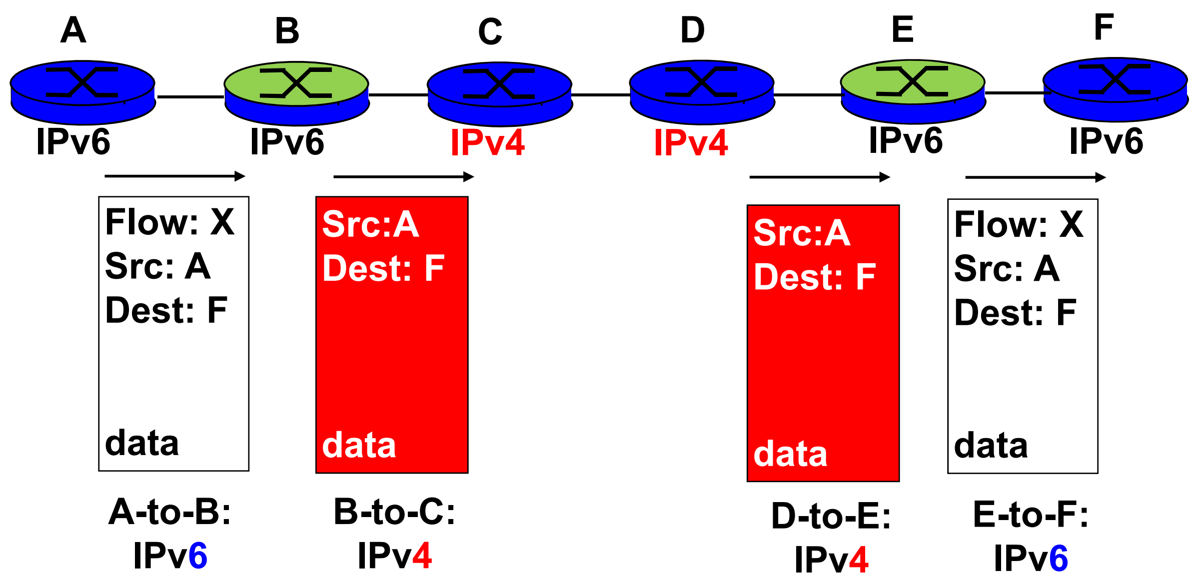
Support of Auto Configuration

- IPv6 promotes StateLess Auto Address Configuration (SLAAC) mechanisms.
- Neighbor Discovery Protocol (NDP) uses four specific ICMPv6 messages: RS – Router Solicitation; RA – Router Advertisement; NS – Neighbor Solicitation; NA – Neighbor Advertisement.
- First, RSs are sent by the nodes to request RAs for configuring the interfaces. The nodes configure their addresses with the prefixes returned in RAs from the router.
- Then, the nodes send NSs to neighbors to test the uniqueness of their addresses. This phase is known as DAD (Duplicate Address Detection).

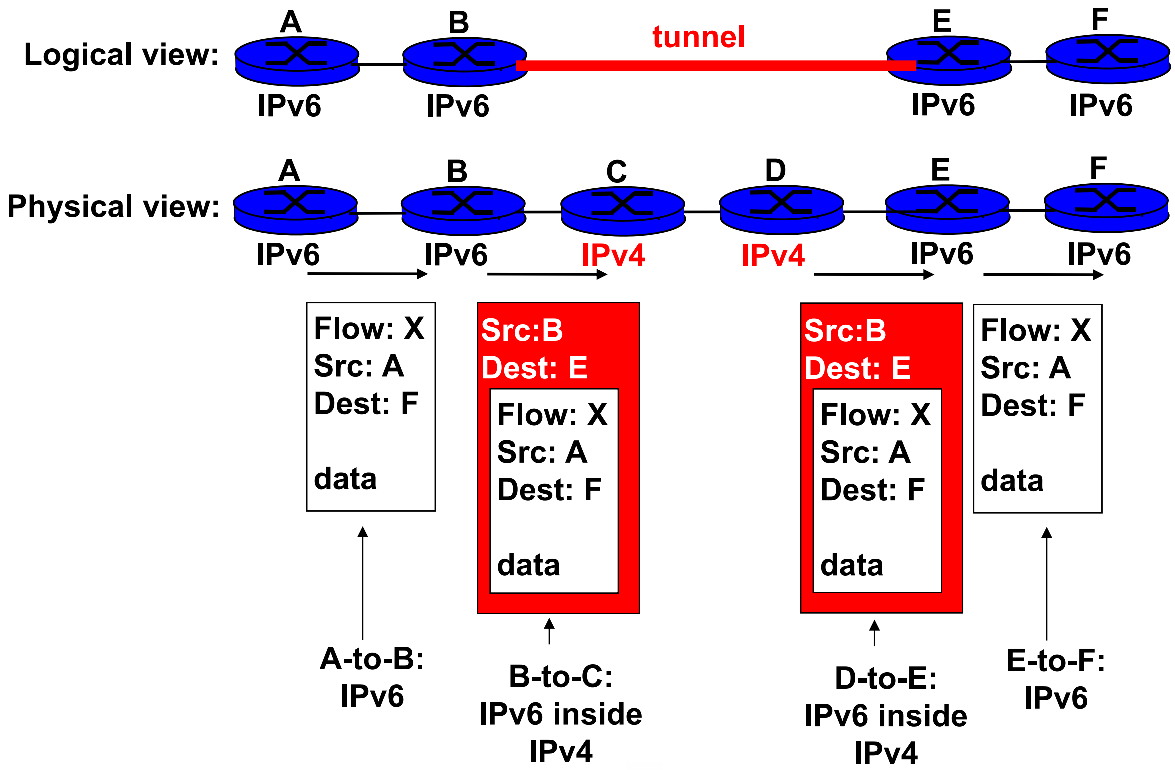
Transition From IPv4 To IPv6

- Not all routers can be upgraded simultaneously
  - no "flag days"
  - How will the network operate with mixed IPv4 and IPv6 routers?
- Two proposed approaches:
  - *Dual Stack*: some routers with dual stack (v6, v4) can "translate" between formats
  - *Tunneling:* IPv6 carried as payload in IPv4 datagram among IPv4 routers

Dual Stack Approach



Tunneling



IPv6 for IoT/M2M

- IPv6 cannot be applied directly to IoT/M2M.
  - IPv6 datagrams at the link layer must be of 1280 byes at minimum. This is too long for IoT/M2M.
  - IPv6 headers are too heavy. A compressed header is required that allows broadcast and routing.
  - IPv6 NDP involves too many messages and must be simplified.

6LoWPAN

Outline

1. Constrained IP Networks
2. What is 6LoWPAN?
3. Motivation and Goal
4. Topology
5. 6LoWPAN Adaptation Layer
6. Neighbor Discovery Protocols

Constrained IP Networks

- A constrained IP network has limited packet sizes, may exhibit a high degree of packet loss, and may have a substantial number of devices that may be powered off at any point in time but periodically "wake up" for brief periods of time.
- These networks and the nodes within them are characterized by severe limits on throughput, available power, and particularly on the complexity that can be supported with limited code size and limited RAM per node.
- Low-Power Wireless Personal Area Networks (LoWPANs) are an example of this type of network. Constrained networks can occur as part of home and building automation, energy management, and the Internet of Things.

Source: IETF CoRE WG

Devices on Constrained Networks

- The general architecture consists of nodes on the constrained network, called Devices, that are responsible for one or more Resources that may represent sensors, actuators, combinations of values or other information.
- Devices send messages to change and query resources on other Devices.
- Devices can send notifications about changed resource values to Devices that have subscribed to receive notification about changes.
- A Device can also publish or be queried about its resources.

Source: IETF CoRE WG

What is 6LoWPAN?

- 6LoWPAN is an acronym of IPv6 over Low power Wireless Personal Area Networks.
- It is designed by the 6LoWPAN working group in IETF (Internet Engineering Task Force).
- RFC 4919 included a detailed review of requirements, which were released in 2007.

- 6LoWPAN is not restricted to radio links, and can be extended to run over other media, for instance it has been extended to run over low-power PLC or G3 OFDM PLC.

- IPv6 is also being adapted to other physical layers, independently of 6LoWPAN, for example, for HomePlug PLC.
- Many fieldbus vendors are now considering an IPv6 adaptation layer for their products.

6LoWPAN WG Documents

draft-ietf-6lowpan-btle-11	Transmission of IPv6 Packets over BLUETOOTH Low Energy	2012-10-12
RFC 4919 (draft-ietf-6lowpan-problem)	IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals	2007-08
RFC 4944* (draft-ietf-6lowpan-format)	Transmission of IPv6 Packets over IEEE 802.15.4 Networks	2007-09
RFC 6282 (draft-ietf-6lowpan-hc)	Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks	2011-09
RFC 6568 (draft-ietf-6lowpan-usecases)	Design and Application Spaces for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)	2012-04
RFC 6606 (draft-ietf-6lowpan-routing-requirements)	Problem Statement and Requirements for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing	2012-05
RFC 6775 (draft-ietf-6lowpan-nd)	Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)	2012-11 new

*RFC 4944 (Proposed Standard) Updated by RFC 6282, RFC 6775

Motivation

- Traditionally, battery-powered networks or low-bitrate networks, such as most fieldbus networks or 802.15.4 were considered **incapable of running IP**.
- In the home and industrial automation networks world, the situation compares to the situation of corporate LANs in the **1980**s:
  "should I run Token-Ring, ATM or IPX/SPX?"
  translates to "should I run ZigBee, LON or KNX?"

Motivation

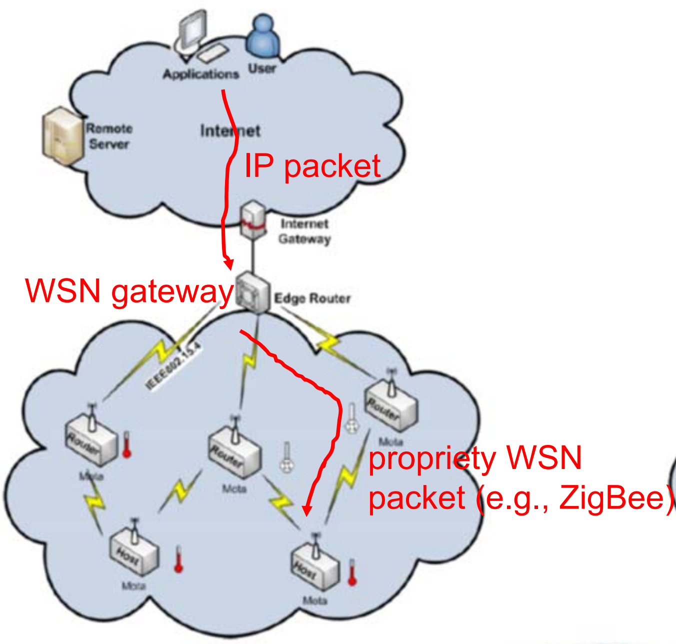
- IP, with its concept of layer 3 routing and internetwork technology, has made those debates about incompatible networks obsolete:
  - the vast majority of LANs and WANs today run IP, and many people can hardly remember which layer 2 technology their IP networks are running on.

Motivation

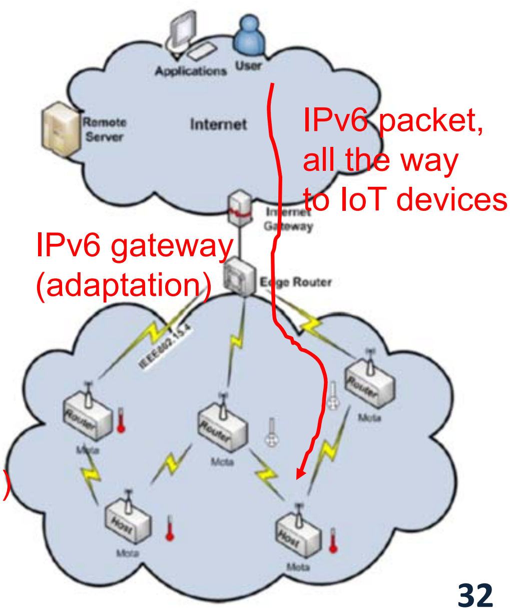
- Almost any layer 2 technology can be used and will simply extend the IP internetwork.
- The same transition to IP is now happening in the home and industrial automation worlds. 6LoWPAN and RPL have made this possible.

Goal of 6LowPAN

- Traditional way: 2-stage



- End-to-end IP transmission



Challenges

- Huge packet header for IPv6:
  - IP address: 16 bytes for source + 16 bytes for destination
  - Many different options (called extension headers)
  - MTU (max trans. unit) = 1280 bytes
- IEEE 802.15.4 frame size: 127 bytes
  - max frame overhead: 25 bytes
  - 21 bytes for AES-128 (highly recommended for security)
  - leaving only 81 bytes for upper layers

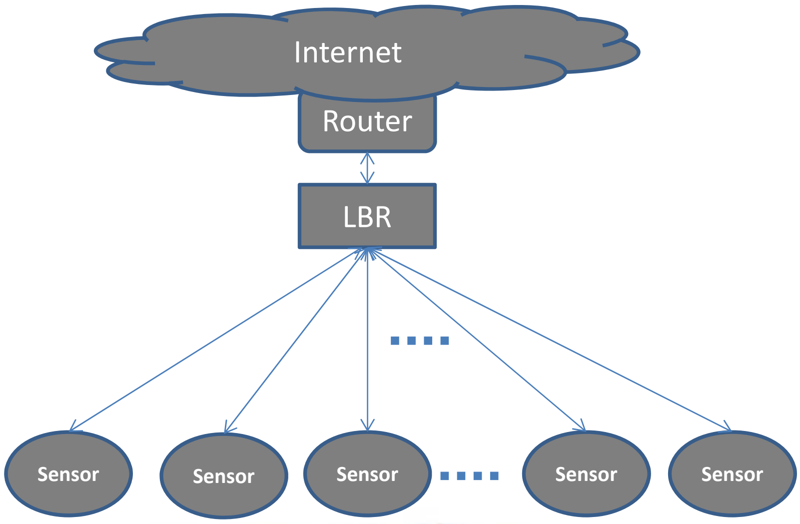
127-25 (Mac header & footer)-21(security)-40(IPv6)-20(TCP) = 21(Application Layer)

Topology

- 6LoWPAN network can be organized around three topologies:
  - Star topology
  - Meshed
  - Routed

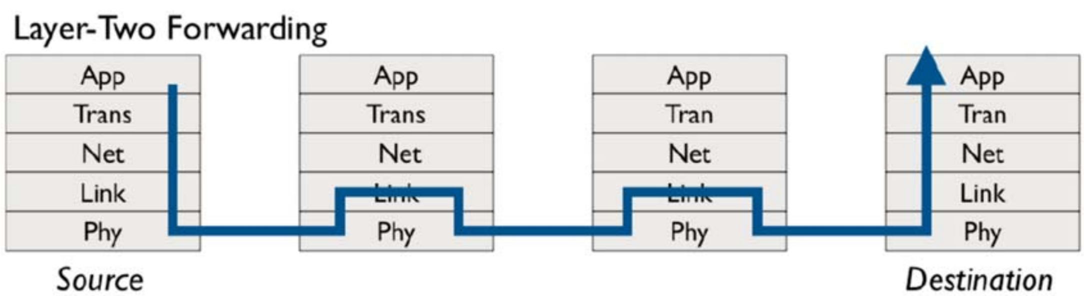
Star topology

- All sensor nodes can reach and are reachable from the LBR (LoWPAN Border Router)



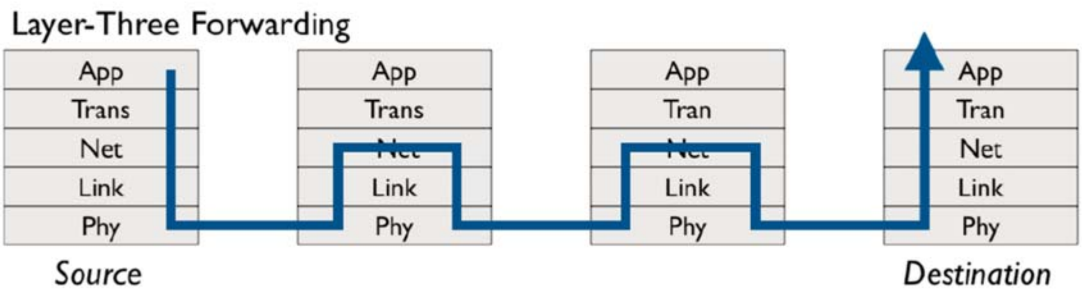
Meshed

- Nodes are organized at **Layer 2** in order to relay frames toward the destination.
- From point of view of the Internet , a meshed network is similar to an Ethernet network where every node shares the same prefix.
- 6LoWPAN refers to that technology as **mesh-under (MU)**.



Routed

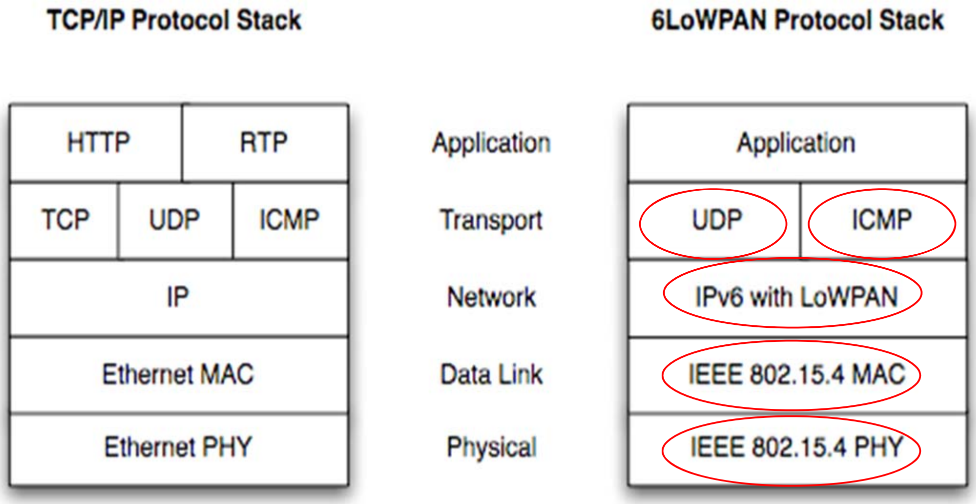
- Nodes act as **routers** and forward packets toward the destination.
- Nodes acting as a router inside the LoWPAN network and not directly connected to the Internet are called LoWPAN Routers (LRs).
- 6LoWPAN refers to that technology as **route-over(RO)**. The best example is RPL protocol.



6LoWPAN Adaptation Layer

- 6LoWPAN is designed to work on top of LLN (802.15.4 networks).
- The optional hop by hop acknowledgment feature of 802.15.4 is used, but the **macMaxFrameRetrie**s should be set to a relatively low value to make sure the 802.15.4 layer will not continue to retry when IP and application-level retransmission mechanisms trigger.

6LoWPAN Protocol Stack



6LoWPAN needs to solve 4 issues

1. **Header compression**
   - On battery-powered networks, long packet headers is synonymous with energy waste.
   - Native IPv6, with its **40-byte header**, was probably one of the worst possible candidates for such networks.
   - In the most favorable case, the LoWPAN and UDP compressed headers require just **6 bytes (based on RFC 6282, RFC 6775) or 7 bytes (based on RFC 4944)**.

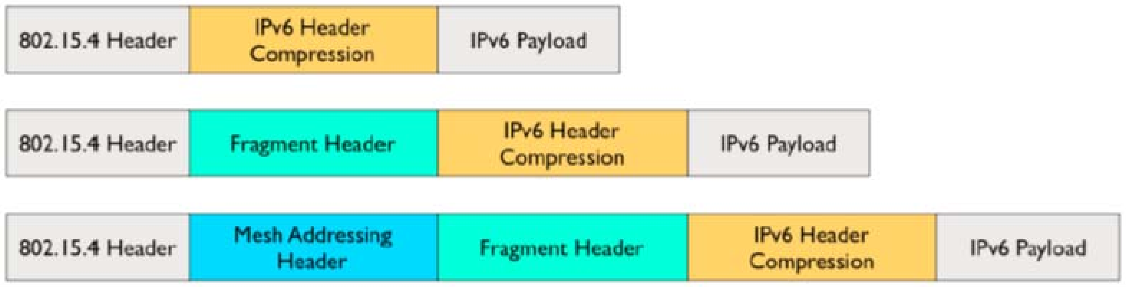
2. **Packet fragmentation and reassembly**
   - low-power networks usually provide small MTUs, because transmission uses energy, and transmission time is proportional to the packet size.
   - Also, small packets are less subject to packet loss that may occur over lossy networks such as 802.15.4.

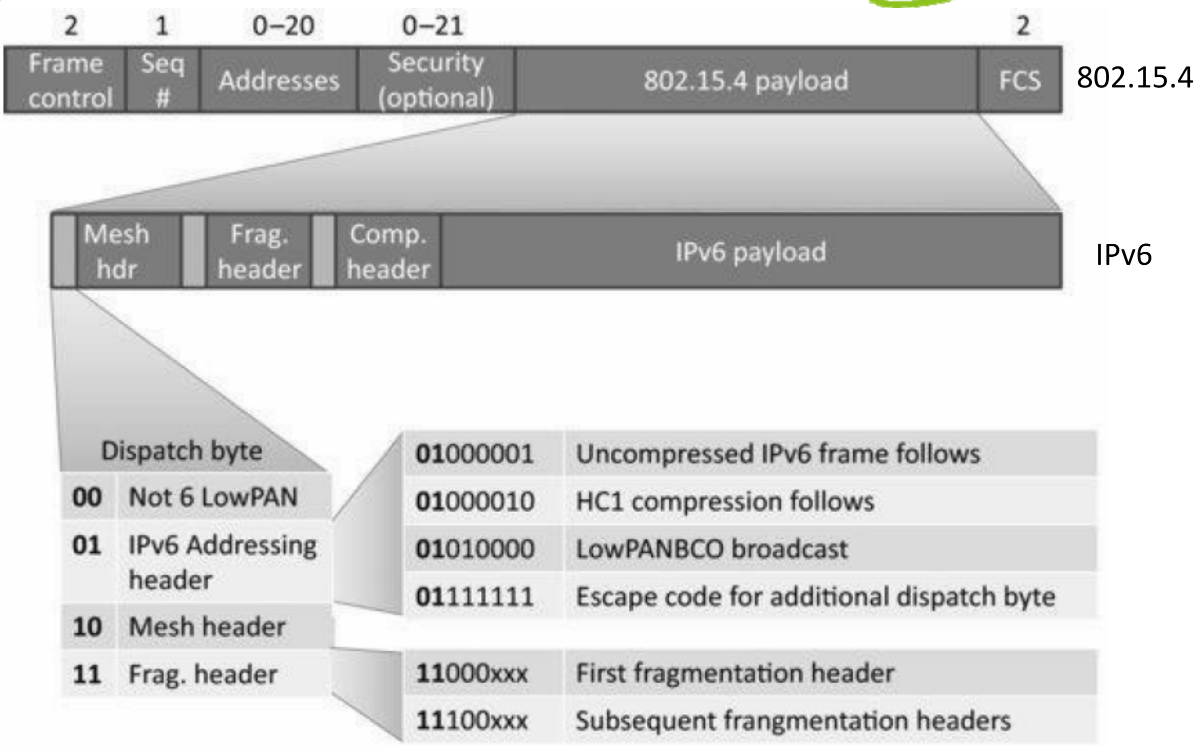
3. **Adaptation of IPv6 neighbor discovery** defined in RFC4861 and 4862.
4. **Support for "mesh under" layer 2 forwarding**.

We will address only Header Compression based on *RFC 4944*,
not the update by *RFC 6282, RFC 6775*.

6LoWPAN Headers

- 6LoWPAN currently defines several headers, which appear in the following order when present:





6LoWPAN Dispatch Byte and Header Stacks

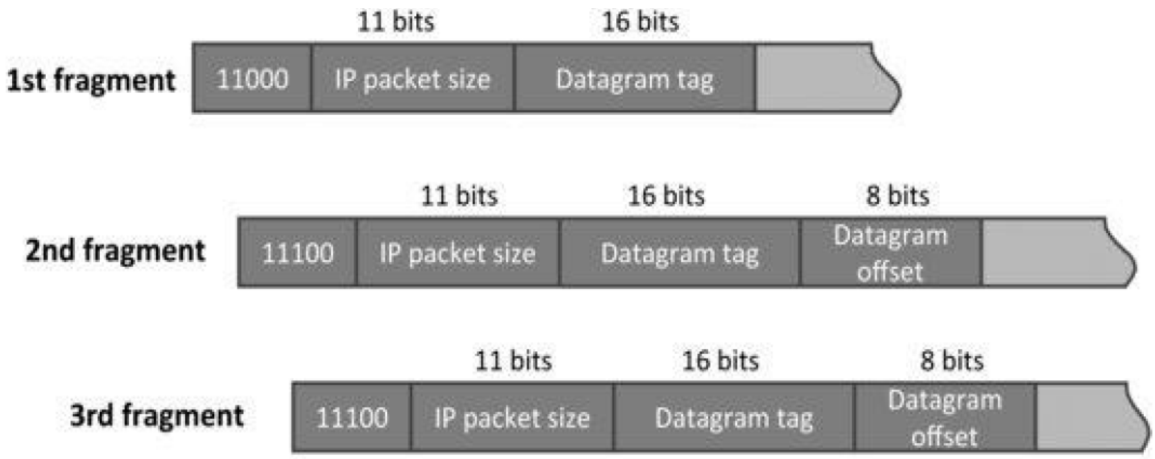
Mesh Addressing Header

- No “mesh-under” protocol is defined for 802.15.4
- Only a facility provided to make it possible in the future

Mesh Addressing Header

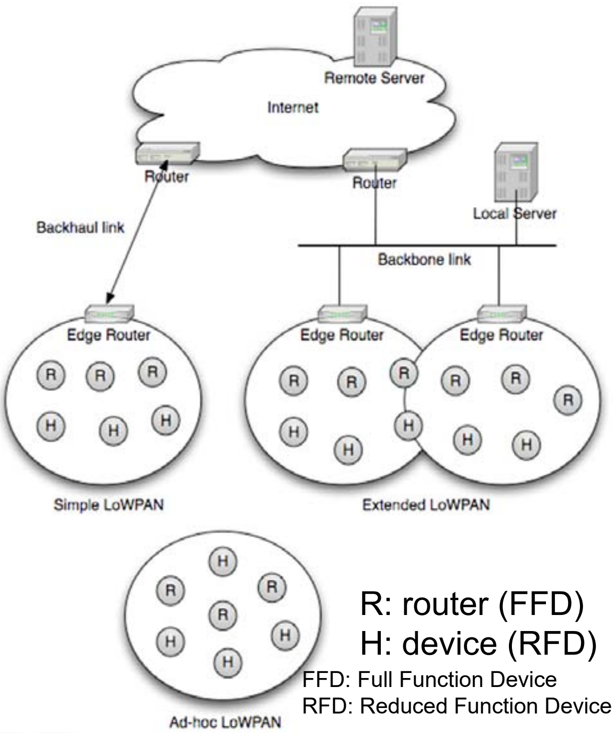
- When 802.15.4 mesh-under routing is enabled, the MAC frame contains the source and destination addresses for each hop, therefore a container is needed for the original and final MAC addresses.
- The mesh addressing header
  - provides a container
  - contains a “HopLeft” counter that should be decremented by each layer2 hop.

Fragment Header



Header Compression Concept

- Architecture
- The compression is stateless.
  - No need to memorize anything.
- Header compression (HC) is executed by edge routers.



- HC1: for IPv6 header
  - compression of IP addresses:
    - from **128 bits** to as low as **2 bits** (sometimes higher, but of low probability)
    - by an in-line technique (explained next)
  - compression of other fields
    - also by an in-line technique
- HC2: for next-level header (such as UDP)

- Compression can be done when the "context" is well-known.
  - **address part**:
    - prefix: removed when it is known by all nodes in network
    - Interface ID: removed when it is directly implied by the MAC address
  - **in-line technique**: a single-bit indicator
    - **1**: "compressed" and uses the default value
    - **0:** "uncompressed" and the actual content is attached at the end
      - In this case, 1 extra single-bit is wasted.
- Can compress for multihop dst/src addresses.

HC1 Encoding

- Starting with bit 0 and ending at bit 7
  - SA: IPv6 source address (bits 0 and 1)
  - DA: IPv6 destination address (bits 2 and 3)
  - TF: Traffic Class and Flow Label (bit 4)
  - NH: Next Header (bits 5 and 6)
  - HC2 encoding (bit 7)

- SA (source address) and DA (dest address):
  - First bit
    - 1: to be transmitted within a local network;
    - 0: other case, and the ***uncompressed fields*** is used
  - Second bit
    - 1: the IID can be derived from the MAC address;
    - 0: other case, and the ***uncompressed fields*** is used

- TF (traffic class and flow label): one bit
  - 0: not compressed (full 8 bits for Traffic Class and 20 bits for Flow Label are attached in the “uncompressed fields”)
  - 1: Traffic Class and Flow Label are zero
- NH (next header): two bits
  - 00: not compressed; full 8 bits are attached in the “uncompressed fields”
    - In this case, 2 bits are wasted.
  - 01: UDP
  - 10: ICMP
  - 11: TCP

- HC2 encoding:
  - 0: No more header compression following HC1 encoding
  - 1: HC1 encoding immediately followed by more header compression. (Bits 1 and 2 of NH determine which of the possible HC2 encodings apply, i.e., UDP, ICMP, or TCP encodings.)

HC2 for Encoding UDP Header Fields

0 — 7	7 — 31
HC_UDP encoding	Fields carried in-line follow...

- bit 0: UDP source port
  - 0: Not compressed, carried "in-line"
  - 1: Compressed to 4 bits.
- bit 1: UDP destination port
  - 0: Not compressed, carried "in-line"
  - 1: Compressed to 4 bits.
- bit 2: Length
  - 0: not compressed, carried "in-line"
  - 1: compressed, length computed from IPv6 header length information.
- bits 3~7: reserved

CoAP

CoAP

- The Constrained Application Protocol (CoAP) is defined by IETF CoRE WG for the manipulation of resources on a device that is on the constrained IP networks.

CoRE WG Documents

draft-ietf-core-block-10	Blockwise transfers in CoAP	2012-10-21
draft-ietf-core-coap-12	Constrained Application Protocol (CoAP)	2012-10-01
draft-ietf-core-groupcomm-03	Group Communication for CoAP	2012-10-19
draft-ietf-core-observe-07	Observing Resources in CoAP	2012-10-22
RFC 6690 (draft-ietf-core-link-format)	Constrained RESTful Environments (CoRE) Link Format	2012-08

Application Scope of CoAP

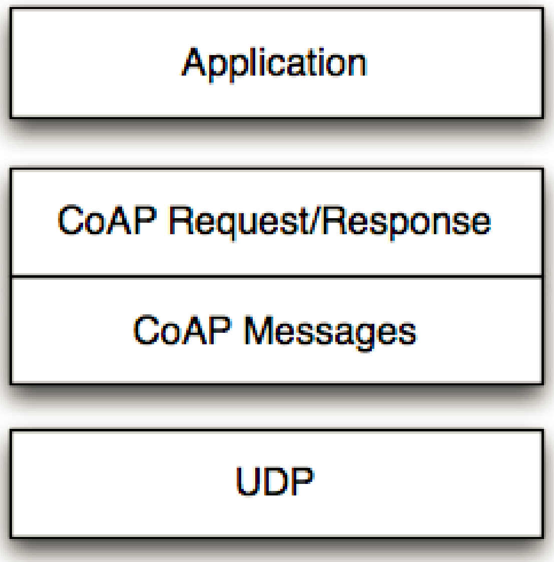
- CoAP targets the type of operating environments defined in the ROLL and 6LowPAN working groups which have additional constraints compared to normal IP networks, but the CoAP protocol will also operate over traditional IP networks.
- This includes applications to monitor simple sensors (e.g. temperature sensors, light switches, and power meters), to control actuators (e.g. light switches, heating controllers, and door locks), and to manage devices.

Source: IETF CoRE WG

CoAP vs. HTTP

- Like HTTP, the CoAP is a way of structuring REST communications but optimized for M2M applications.
- TCP and HTTP are considered too heavy for 6LowPAN devices such as sensors. CoAP is thus based on UDP and a compressed simplified message exchange.

CoAP RESTful Applications



Source: IETF IPv6 WG

CoAP Message Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Ver| T |  OC   |      Code     |          Message  ID          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Options (if any) ...!
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Payload (if any) ...!
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Ver - Version (1)
T - Transaction Type
- CON – Confirmable
- NON - Non-Confirmable
- ACK – Acknowledgement
- RST - Reset

OC - Option Count, number of options after this header
Code - Request Method (1-10) or Response Code (40-255)
Message ID - Identifier for matching responses

CoAP Code and Message ID

- Code: compressed from HTTP text representation (3 numbers) into one byte
  - HTTP requests =>first 3 bits 000; next five bits 0~32 (1: GET; 2: POST; 3:PUT; 4:DELETE etc.)
  - HTTP responses=>first 3 bits 001-101 (1~5) representing the first number of 1xx: informational, 2xx: success, 3xx: redirection, 4xx: client error, 5xx: server error; xx represented by next five bits 00001~01111 (1~15 used only; e.g. with HTTP response 201 is represented as 010-00001; HTTP response 400 is represented as 100-00000 etc.)
- Message ID: used in the acknowledgment process to tie a request with a response.

CoAP Option Count (OC)

- OC is a 4-bit field specifying the number of options.
- Options are stored as TLV (Type Length Value).
- Options are always sent in the same numerical order based on the type and the type is encoded as an *option delta*.
- For examples, if a message contains options types 1, 5, 6, 7 and 11, the option types sent will be 1, 4, 1, 1 and 4.

CoAP Options

```
 0   1   2   3   4   5   6   7!
+---+---+---+---+---+---+---+---+
| option delta  |    length     | for 0..14
+---+---+---+---+---+---+---+---+

                                                 for 15..270:
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| option delta  | 1   1   1   1 |          length - 15          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

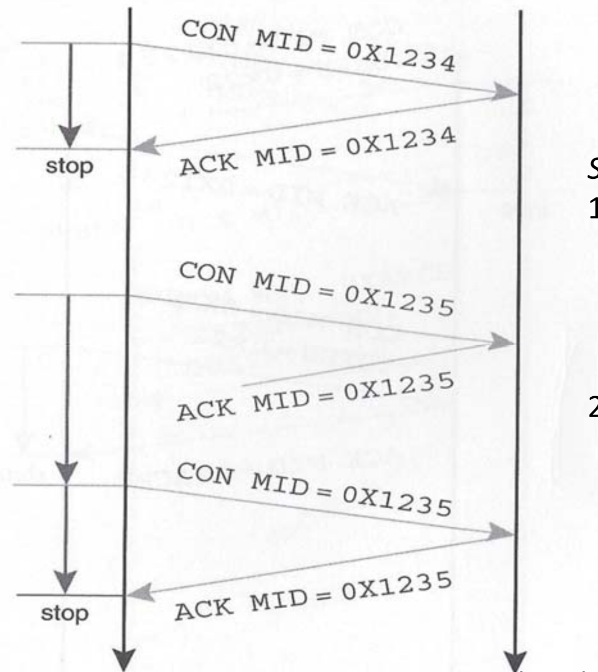
Option Delta - Difference between this option type and the previous
Length - Length of the option value (0-270)
Value - The value of Length bytes immediately follows Length

Examples of Option types

No.	C/E	Name	Format	Length	Default
1	Critical	Content-Type	uint	1-2 B	0
2	Elective	Max-Age	uint	0-4 B	60
3	Critical	Proxy-Uri	string	1-270 B	(none)
4	Elective	ETag	opaque	1-8 B	(none)
5	Critical	Uri-Host	string	1-270 B	(see below)
6	Elective	Location-Path	string	1-270 B	(none)
7	Critical	Uri-Port	uint	0-2 B	(see below)
8	Elective	Location-Query	string	1-270 B	(none)
9	Critical	Uri-Path	string	1-270 B	(none)
11	Critical	Token	opaque	1-8 B	(empty)
15	Critical	Uri-Query	string	1-270 B	(none)

1 (Content-type) is the value referring to a mime value describing the syntax of the payload, for example, 0: text/plain, 44: application/soap+xml.
2 (Max-Age) gives the maximum duration in seconds for which the answer may be cached.
4 (Etag) is used for caching
10 (Observe) is used to receive regularly updated values from the server.
11 (Token) is used to match a response with a request.
16 (Block) is used to transfer blocks of responses

Example 1 of CoAP Requests



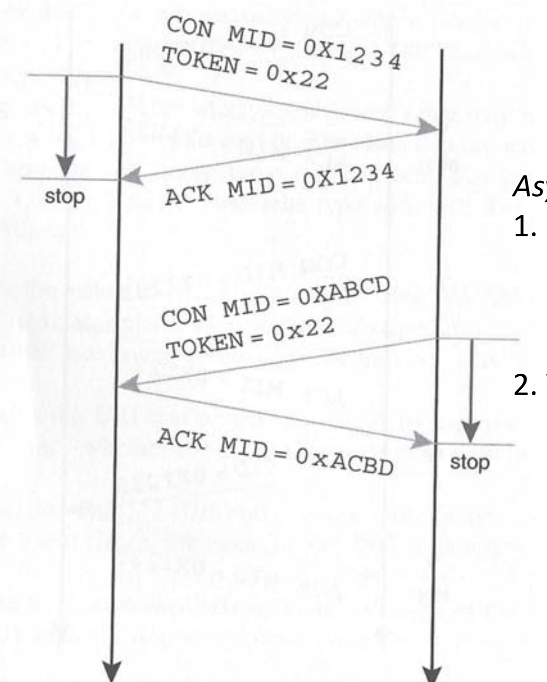
Synchronous Message Exchange

1. A CONfirmable message followed by ACKowledgement piggybacked with the response in the same Message ID (MID).

2. When ACKnowledgment was lost, Client's timer expires and it resends the message.

Source: M2M Communications: A Systems Approach, Wiley, 2012

Example 2 of CoAP Requests



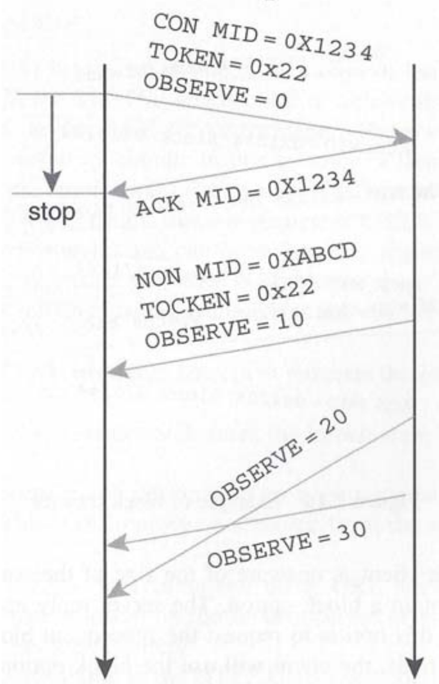
Asynchronous Message Exchange

1. A CONfirmable message with TOKEN option can be acknowledged immediately without a response.

2. When the response is available, it can be returned in a new CON message with the same TOKEN ID.

Source: M2M Communications: A Systems Approach, Wiley, 2012

Example 3 of CoAP Requests



Periodic response from a server

1. A CONfirmable message from the client contains OBSERVE option asking periodic responses from the server.

2. The server send NON responses with the same TOKEN ID.

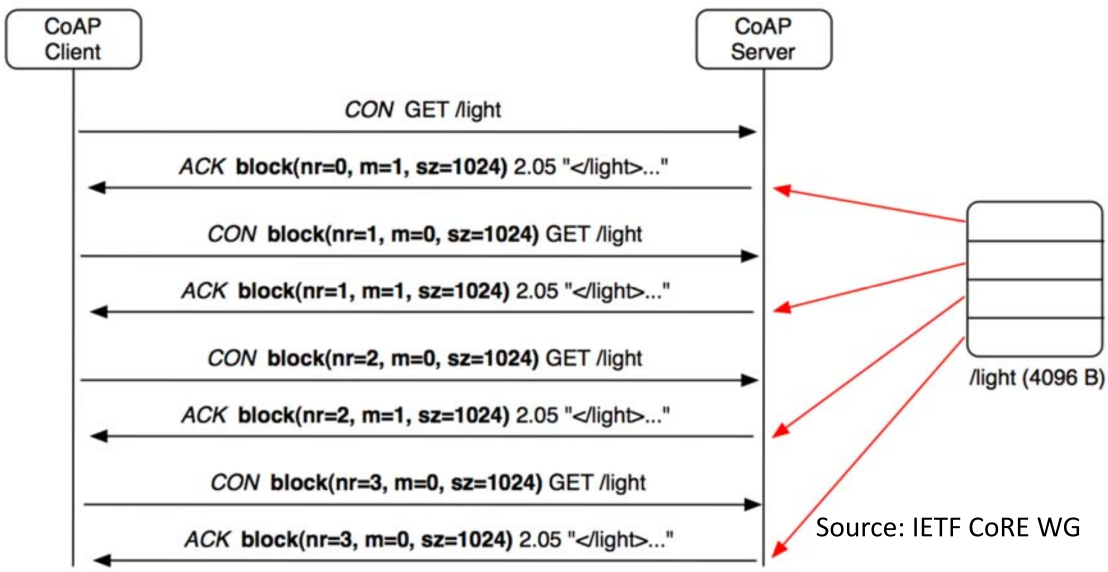
3. OBSERVE will be increased to indicate the order of the response.

4. The client will ignore OBSERVE=20 since it arrives later than OBSERVE=30.

5. Either client or server can terminate the process.

Source: M2M Communications: A Systems Approach, Wiley, 2012

Example 4 of CoAP Requests

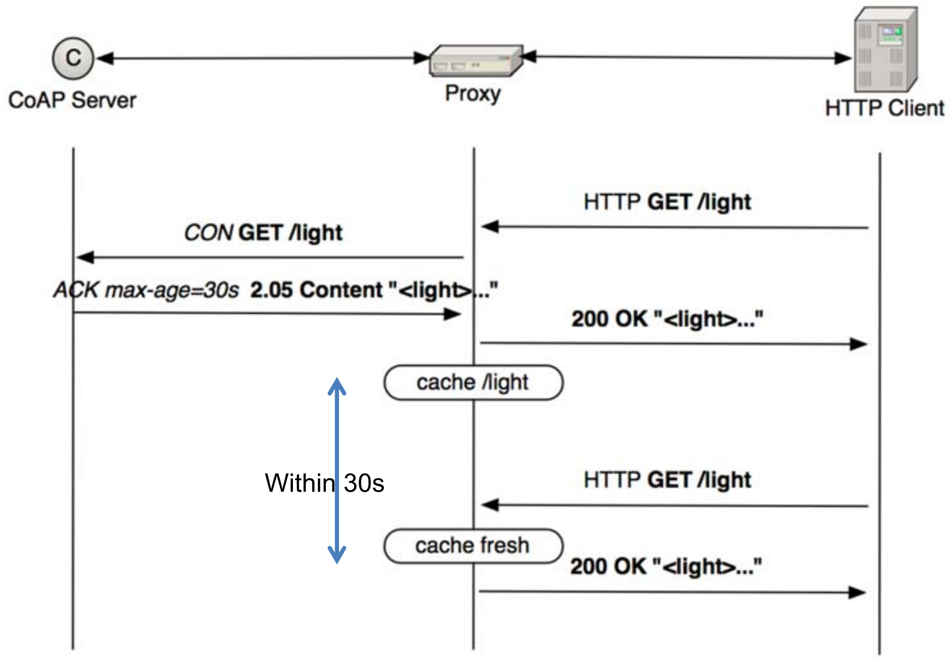


Source: IETF CoRE WG

Block Transfer from Server to Client

1. A CONfirmable message from Client to get information.
2. Server indicates it has block of information to send.
3. Client then asks for more blocks of information.

Proxying and Caching



Source: IETF IPv6 WG

CoAP Caching Model

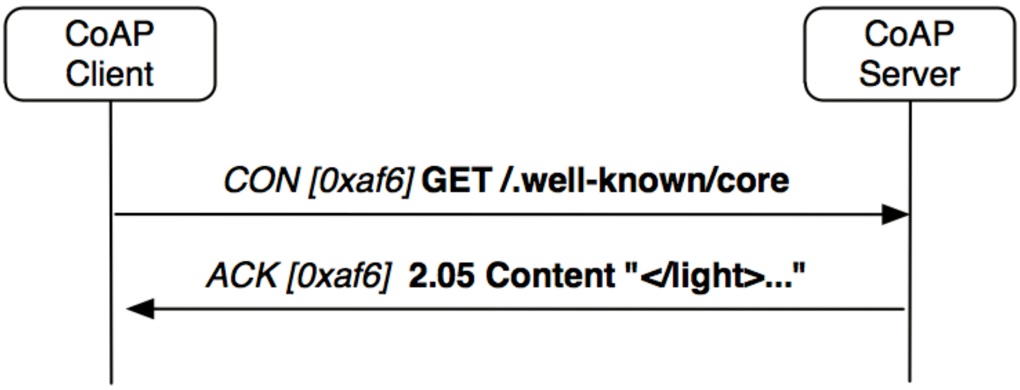
Cacheability determined by response code

- Freshness model
  - Max-Age option indicates cache lifetime
- Validation model
  - Validity checked using the Etag Option (http://en.wikipedia.org/wiki/HTTP_ETag)

CoAP Resource Discovery

- Resource Discovery with CoRE Link Format
  - Discovering the links hosted by CoAP servers
  - GET /.well-known/core
  - Returns a link-header style format based on RFC5988 including URL, relation, type, interface, content-type etc.
  - See draft-ietf-core-link-format

Example of Resource Discovery



</light>;rt="Illuminance";ct=0,
</s/maastr.xml>;title="Maastricht weather";ct=1,
</s/maastr/temp>;title="Temperature in Maastrich";ct=1,
</s/oulu.xml>;title="Oulu weather";ct=1,
</s/oulu/temp>;title="Temperature in Oulu";ct=1,
</s/temp>;rt="Temperature";ct=0

Source: IETF CoRE WG

RPL

What is RPL?

- The IETF Routing Over Low-power and Lossy networks (ROLL) Working Group was formed in 2008
  - to create an IP level routing protocol adapted to the requirements of mesh networking for IoT/M2M
- The first version of RPL (Routing Protocol for Low-power and lossy networks) was finalized in April 2011

Working Items of ROLL WG

- Protocol work
  - http://datatracker.ietf.org/doc/draft-ietf-roll-rpl/
  - RPL is designed to support different LLN application requirements
    - RFC 5548 - Routing requirements for Urban LLNs
    - RFC 5673 - Routing requirements for Industrial LLNs
    - RFC 5826 - Routing requirements for Home Automation LLNs
    - RFC 5867 - Routing requirements for Building Automation LLNs
- Routing metrics
  - http://tools.ietf.org/id/draft-ietf-roll-routing-metrics/
- Security Framework
  - http://tools.ietf.org/id/draft-ietf-roll-security-framework/
- The Trickle Algorithm (RFC 6206): adjustable transmission window scheme
- Terminology
  - http://tools.ietf.org/id/draft-ietf-roll-terminology/
- Applicability statement
  - http://tools.ietf.org/id/draft-ietf-roll-applicability-ami/

Main Functionality of RPL

- RPL specifies a routing protocol specially adapted for the needs of IPv6 communication over “low-power and lossy networks” (LLNs), supporting
  - peer to peer traffic (P2P)
  - point to multipoint (P2MP) communication
  - multipoint to point (MP2P) communication

Five Criteria

- Table scalability: how does the routing table size scale?
- Loss response: how expensive is it when links come and go?
- Control cost: how does the control overhead scale?
- Link cost: can the protocol consider link properties?
- Node cost: can the protocol consider node properties?

Slide from IETF-72

Routing Protocols

- Static: can’t handle dynamic networks well
  - Open Shortest Path First (OSPF)
  - Intermediate System to Intermediate System (IS-IS)
- Dynamic: not scalable enough
  - Ad-hoc On-demand Distance Vector (AODV)
  - Optimized Link State Routing (OLSR)

Summary

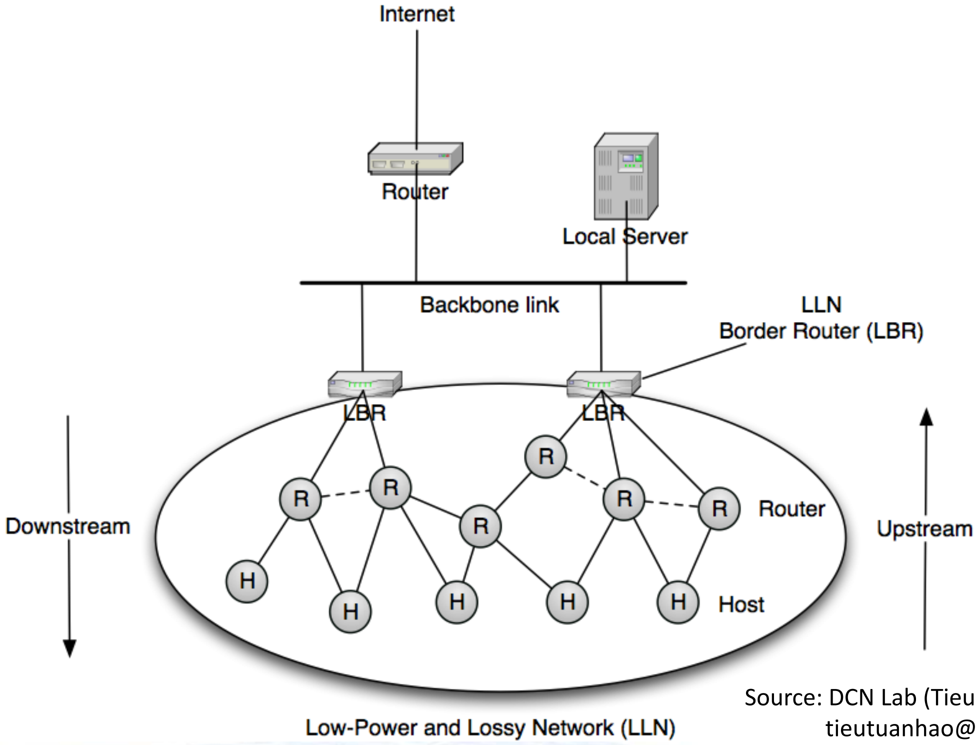
Name	Table Size	Loss Response	Control Cost	Link Cost	Node Cost
OSPF	fail	fail	fail	pass	fail
OLSRv2	fail	fail	fail	pass	pass
TBRPF	fail	pass	fail	pass	?
RIP	fail	fail	fail	?	fail
AODV	pass	?	pass	fail	fail
DSDV	fail	fail	fail	?	fail
DYMO[-low]	pass	fail	pass	fail	fail
DSR	fail	?	pass	fail	?

Slide from IETF-72

Key Design of RPL

- Based on *distance vector algorithms*, such as Routing Information Protocol (RIP)
- Correct performance issues due to bad loop detection
- Remain flexible in the definition of routing strategies
- A Trickle algorithm is used to limit the number of periodic messages
- The base RPL specification is optimized only for MP2P traffic or P2MP
  - P2P is optimized only through use of additional mechanisms
- Utilize only bidirectional links

ROLL Architecture



Source: DCN Lab (Tieu Tuan Hao; tieutuanhao@dcn.ssu.ac.kr)

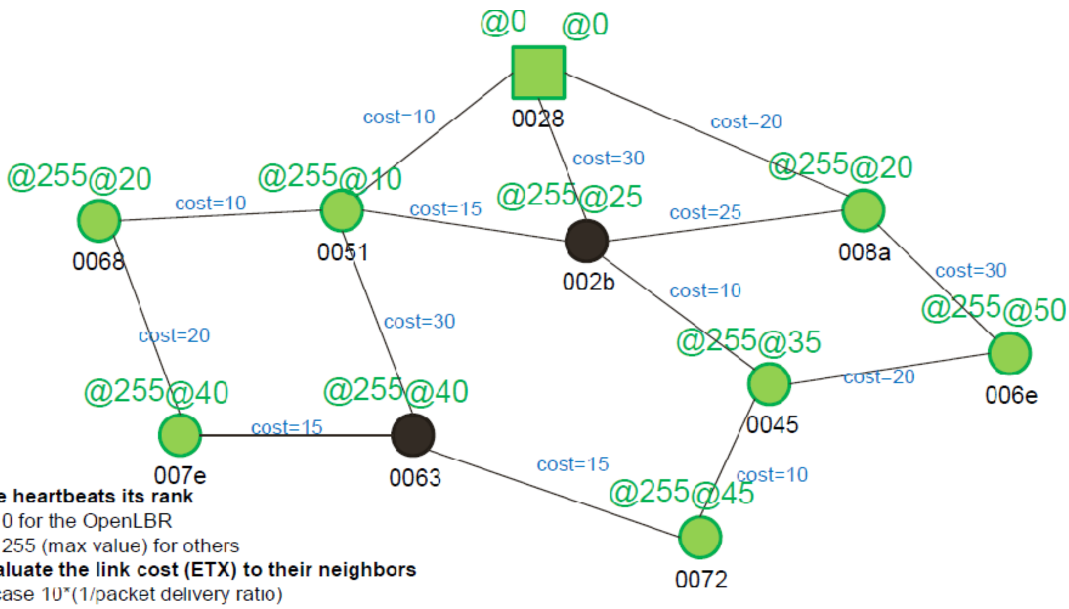
Destination Oriented Direct Acyclic Graphs (DODAGs)

- RPL builds one or more DODAGs.
- Each DODAG is a directed graph with no cycles and with a single root node, derived from optimization objectives specified by an **Objective Function (OF)**
  - the OF is designated by the *OCP (Objective Code Point)* field of **DIO (DODAG Information Object)**
    - the OF computes the "rank" measuring the "distance" between the node and the DODAG root, and
    - also defines the parent node selection policy
  - bidirectional connectivity must be verified before accepting a router as a parent
- OFs are defined in other companion documents
  - Refer to draft-ietf-roll-routing-metrics.

Metrics and Rank

- *Routing Metrics* are used by routing protocols to compute shortest paths to achieve an optimization goal.
- *Rank*: path calculation according to objective Metrics
  - Scalar that represents relative position within a DAG
  - Strictly increasing from the root
  - Topological constraint to avoid and detect loops
  - Coarse granularity allows siblings (in addition to parents, children)
- *Objective Functions (OF)* transform the **metrics** into a **rank**. Two OFs are defined
  - OF0: based on hop counts
  - MRHOF (Minimum Rank with Hysteresis Objective Function)

Rank and ETX (Expected Transmission Count)



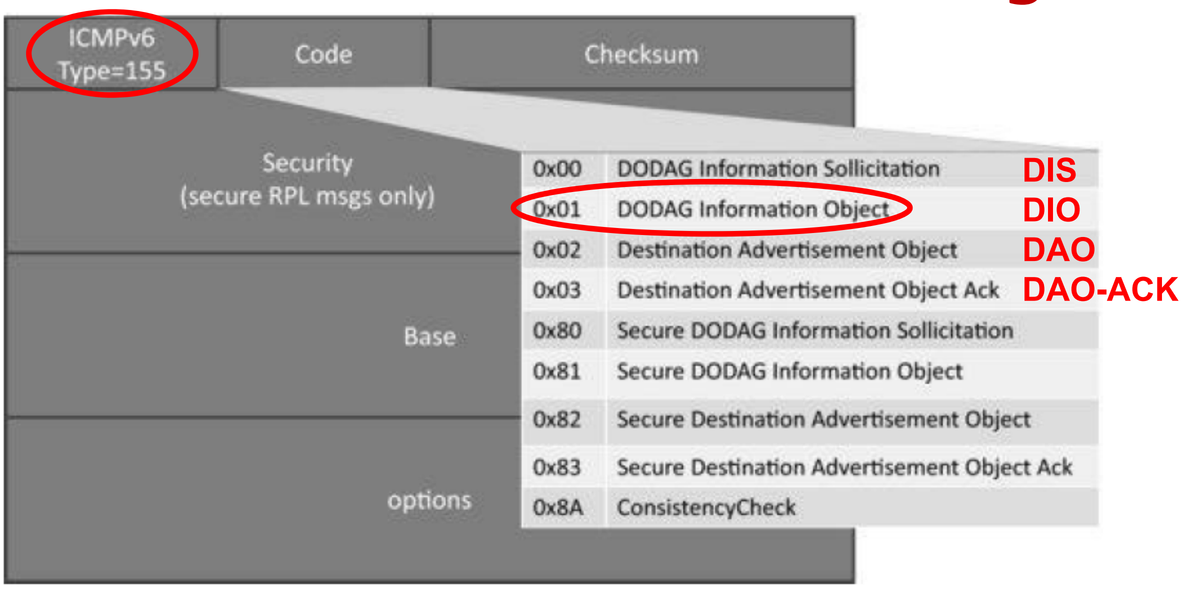
1. **Each node heartbeats its rank**
   - Initially 0 for the OpenLBR
   - Initially 255 (max value) for others
2. **Nodes evaluate the link cost (ETX) to their neighbors**
   - In our case 10*(1/packet delivery ratio)
   - Perfect link: cost=10
   - Link with 50% loss: cost=20
3. **Nodes update their rank as min(rank neighbor+link cost) over all neighbors**
   - The chosen neighbor is prefered routing parent
4. **Continuous updating process**

Source: DCN Lab (Tieu Tuan Hao; tieutuanhao@dcn.ssu.ac.kr)

RPLinstanceID

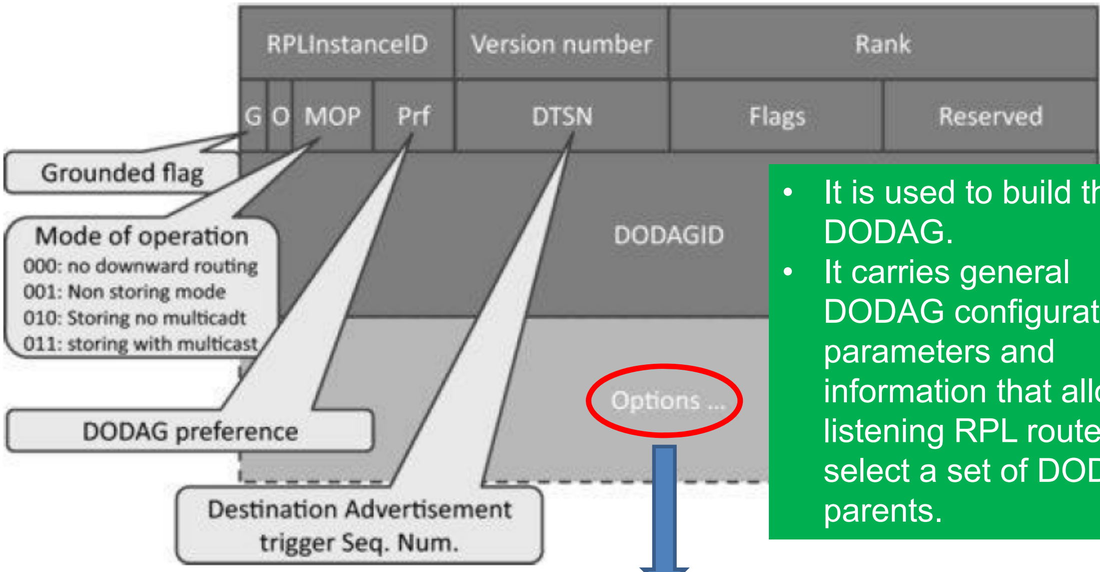
- Multiple concurrent instances of RPL may operate in a given network, each of them is characterized by a unique RPLinstanceID.
  - Below, the behavior of an individual RPL instance is described.

ICMPv6 RPL Control Message



Reference: Figure 12.7: Structure of ICMPv6 RPL control message

DODAG Information Object (DIO)



- It is used to build the DODAG.
- It carries general DODAG configuration parameters and information that allows listening RPL routers to select a set of DODAG parents.

Next Slide

Reference: Figure 12.8: RPL DIO base object (followed by options)

Options of RPL DIO

Option:

Type	Length	Data ...

- Option types
  - 0x02: metric container option
    - Estimate the cost to reach destinations
  - 0x03: routing information option
    - Contains the same fields as the IPv6 neighbor discovery route information option
  - 0x04: DODAG information option
    - Contains the rank a node can advertise when reattaching to a DODAG, or
    - The default lifetime of all RPL routes
  - 0x08: prefix information option
    - Contains the same fields as the IPv6 neighbor discovery prefix option

Control Message Exchange

- Each DODAG, uniquely identified by RPLInstanceID and DODAGID, is incrementally built from the root to the leaf nodes.
- RPL nodes send DIOs periodically via link-local multicasts.
- Joining nodes may request DIOs from their neighbors by multicasting DIS.
- DTSN (Destination Advertisement Trigger Seq. Num.) is an 8-bit unsigned integer set by the issuer of the message. In the storing mode, increasing DTSN is to request updated DAOs from child nodes.

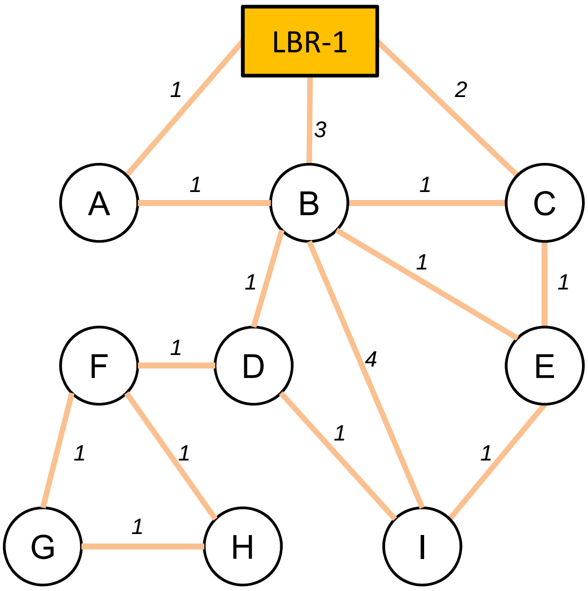
Example Routing Metrics in LLNs

Node Metrics	Link Metrics
Node State and Attributes Object Purpose is to reflects node workload (CPU, Memory…) "O" flag signals overload of resource "A" flag signal node can act as traffic aggregator	**Throughput Object** Currently available throughput (Bytes per second) Throughput range supported
Node Energy Object "T" flag: Node type: 0 = Mains, 1 = Battery, 2 = Scavenger "I" bit: Use node type as a constraint (include/exclude) "E" flag: Estimated energy remaining	**Latency** Constraint - max latency allowable on path Metric - additive metric updated along path
Hop Count Object Constraint - max number of hops that can be traversed Metric - total number of hops traversed	**Link Reliability** Link Quality Level Reliability (LQL) 0=Unknown, 1=High, 2=Medium, 3=Low Expected Transmission Count (ETX) (Average number of TX to deliver a packet)
Object can be used as metric and/or constraint - metric can be additive/max/..	**Link Colour** Metric or constraint, arbitrary admin value

Specified in draft-ietf-roll-routing-metrics

Reference: IoT Workshop RPL Tutorial, Cisco Systems

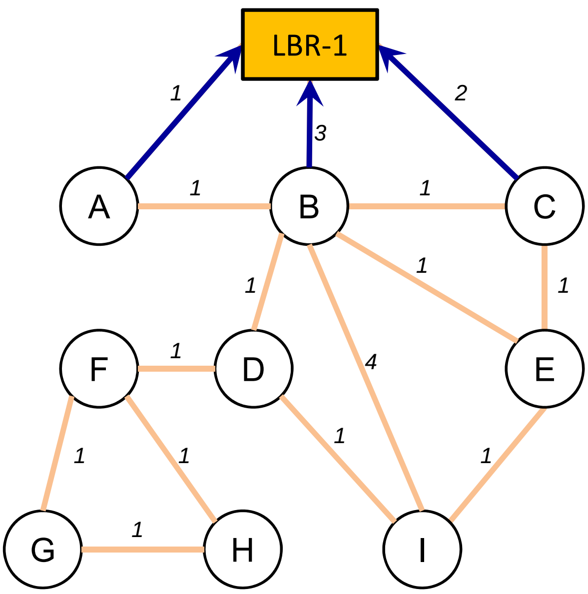
DAG Construction



- LLN links are described
- Links are annotated by ETX (Expected Transmission Count) value
- Objective Code Point (OCP) for example
  - Metric: ETX
  - Objective: Minimize ETX
  - Depth computation:
    - Depth ~ ETX

Source: DCN Lab (Tieu Tuan Hao; tieutuanhao@dcn.ssu.ac.kr)

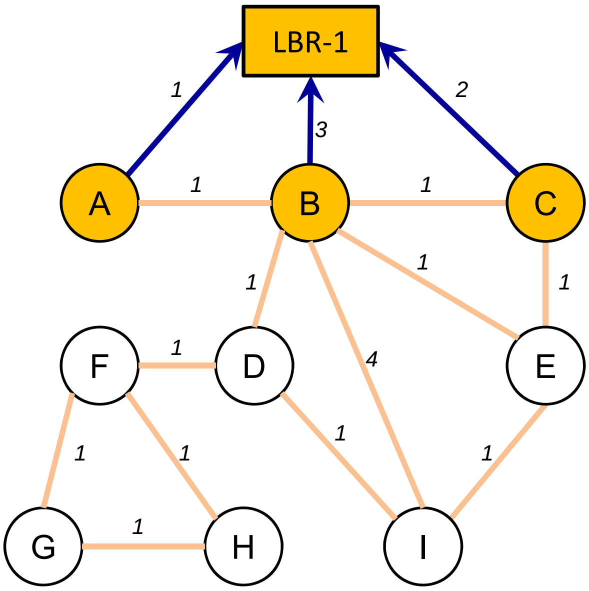
DAG Construction



- LBR-1 multicasts RA-DIO
- Nodes A, B, C receive and process RA-DIO
- Nodes A, B, C consider link metrics to LBR-1 and the optimization objective
- The optimization objective can be satisfied by joining the DAG rooted at LBR-1
- Nodes A, B, C add LBR-1 as a DAG parent and join the DAG

Source: DCN Lab (Tieu Tuan Hao; tieutuanhao@dcn.ssu.ac.kr)

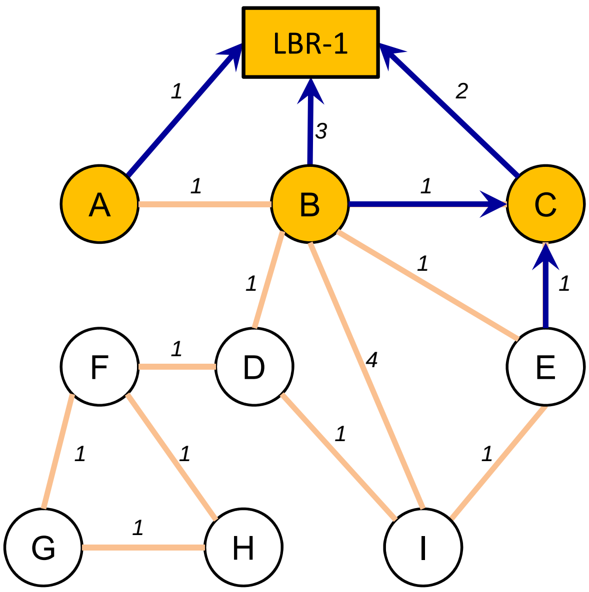
DAG Construction



- Node A is at Depth 1 in the DAG, as calculated by the routine indicated by the example OCP (Depth ~ ETX)
- Node B is at Depth 3, Node C is at Depth 2
- Nodes A, B, C have installed default routes (::/0) with LBR-1 as successor

Source: DCN Lab (Tieu Tuan Hao; tieutuanhao@dcn.ssu.ac.kr)

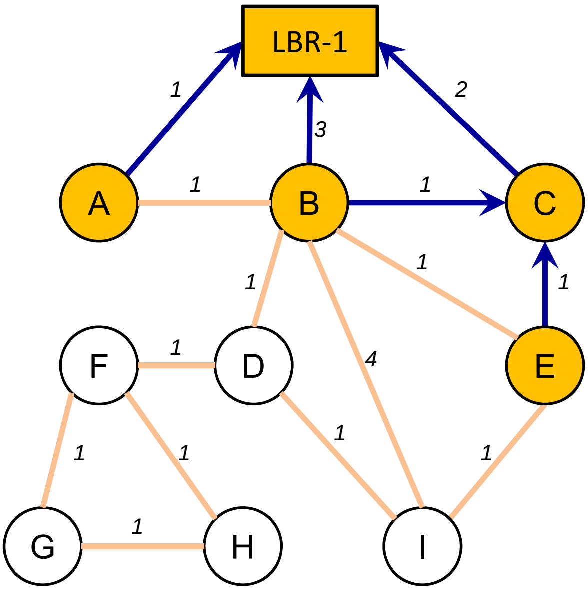
DAG Construction



- The RA-DIO timer on Node C expires
- Node C multicasts RA-DIO
- LBR-1 ignores RA-DIO from deeper node.
- Node B can add Node C as ***alternate*** DAG Parent, remaining at Depth 3
- Node E joins the DAG at Depth 3 by adding Node C as DAG Parent

Source: DCN Lab (Tieu Tuan Hao; tieutuanhao@dcn.ssu.ac.kr)

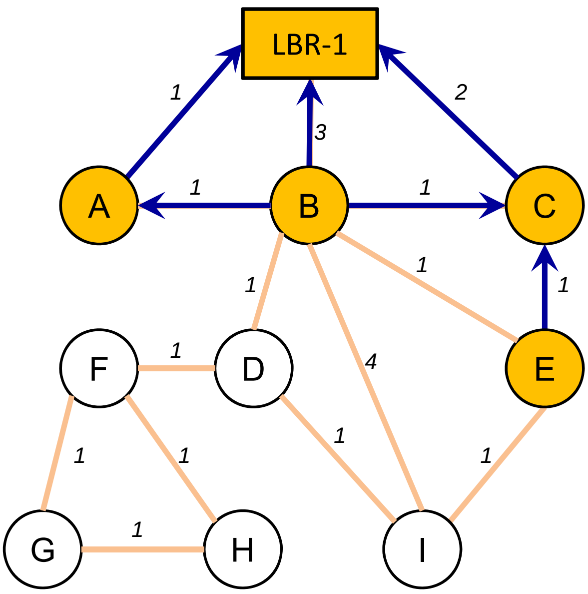
DAG Construction



- Node A is at Depth 1, and can reach ::/0 via LBR-1 with ETX 1
- Node B is at Depth 3, with DAG Parent LBR-1, and can reach ::/0 via LBR-1 or C with ETX 3
- Node C is at Depth 2, reach ::/0 via LBR-1 with ETX 2
- Node E is at Depth 3, reach ::/0 via C with ETX 3

Source: DCN Lab (Tieu Tuan Hao; tieutuanhao@dcn.ssu.ac.kr)

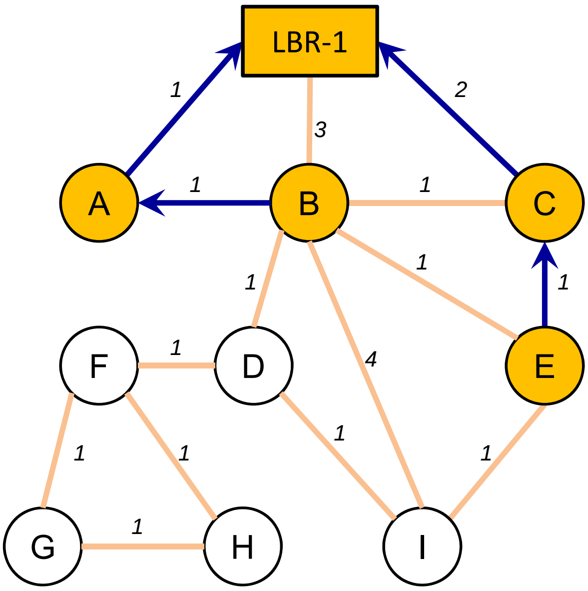
DAG Construction



- The RA-DIO timer on Node A expires
- Node A multicasts RA-DIO
- LBR-1 ignores DIO from deeper node
- Node B adds Node A
- Node B can improve to a more optimum position in the DAG via A with ETX 2
- Node B *removes* LBR-1, Node C as DAG Parents

Source: DCN Lab (Tieu Tuan Hao; tieutuanhao@dcn.ssu.ac.kr)

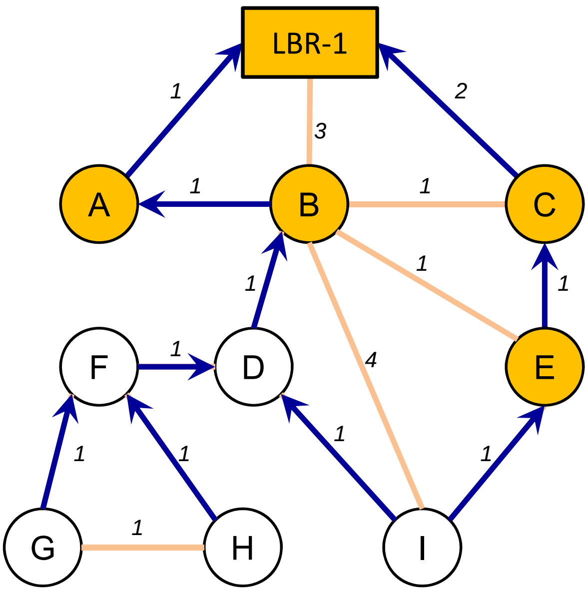
DAG Construction



- Node A is at Depth 1, ::/0 via LBR-1 with ETX 1
- Node B is at Depth 2, ::/0 via A with ETX 2
- Node C is at Depth 2, ::/0 via LBR-1 with ETX 2
- Node E is at Depth 3, ::/0 via C with ETX 3

Source: DCN Lab (Tieu Tuan Hao; tieutuanhao@dcn.ssu.ac.kr)

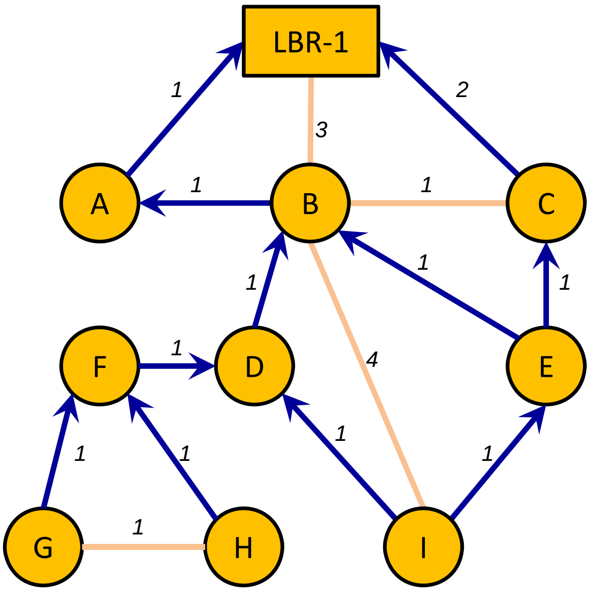
DAG Construction



- DAG Construction continues…
- And is continuously maintained

Source: DCN Lab (Tieu Tuan Hao; tieutuanhao@dcn.ssu.ac.kr)

MP2P Traffic



- *MP2P traffic* is inward traffic toward DAG Root
- DAG Root may also extend connectivity to other prefixes beyond the DAG root, as specified in the DIO
- Nodes may join multiple DAGs as necessary to satisfy application constraints

Source: DCN Lab (Tieu Tuan Hao; tieutuanhao@dcn.ssu.ac.kr)

DODAG Maintenance

- The DIO information is also used to maintain an existing DODAG affiliation.
- DODAG repair
  - Each DIO announcement is attached to a specific DODAG version.
  - DODAG repair: The root can trigger a complete recalculation of the DODAG topology by change the DODAG version.
- Poison routes: A node may poison previously announced routes by advertising a special rank value of INFITIE_RANK (= 0xFFFF) to break all sub-DAGs topology.

Summary

- We cover four key IP protocols for IoT/M2M applications
  - IPv6
  - CoAP
  - 6LoWPAN
  - RPL
- 6LoWPAN, RPL and CoAP specifically targeted at 802.15.4 types of LLN; CoAP, however, applicable also to any IP networks.
- Open source software available for these protocols.