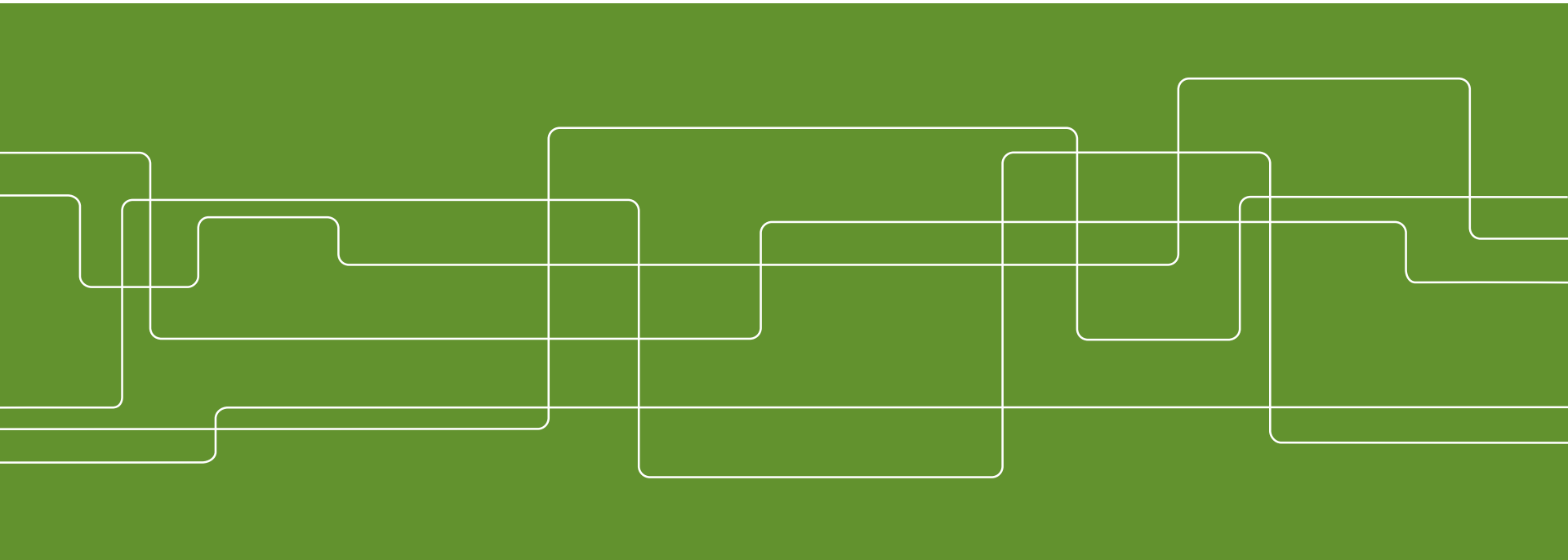
IK2215
Advanced Internetworking

IoT—Internet of Things

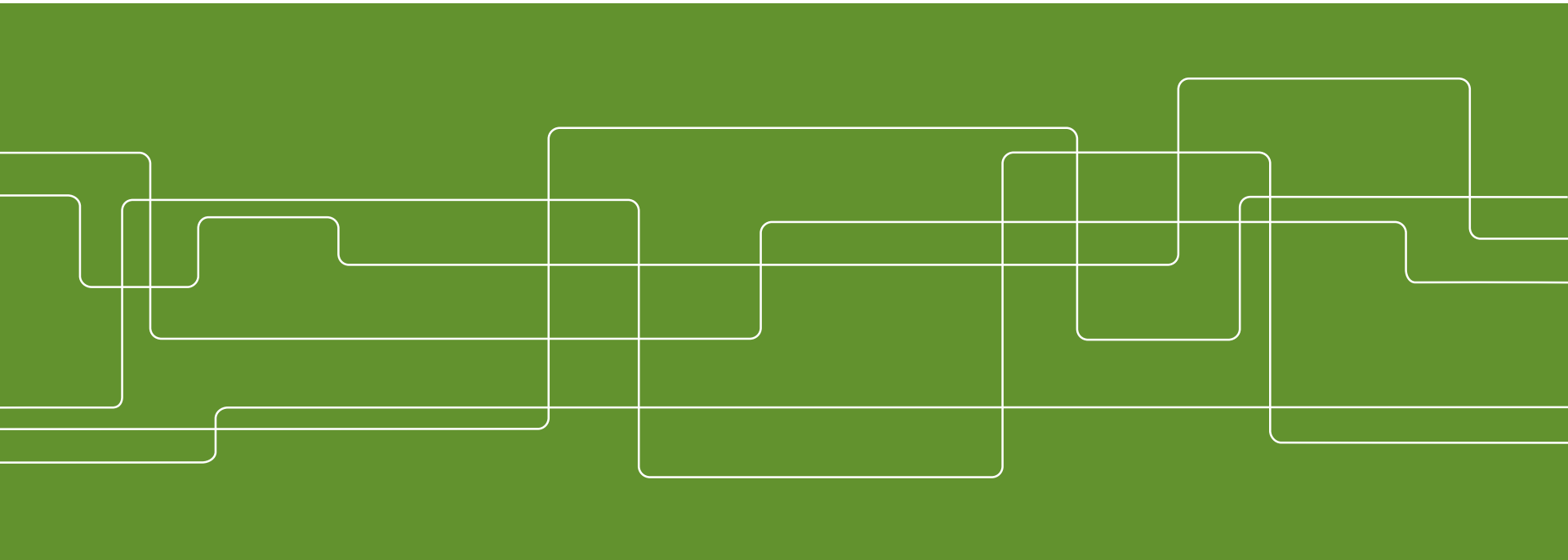
Markus Hidell and Robert Olsson



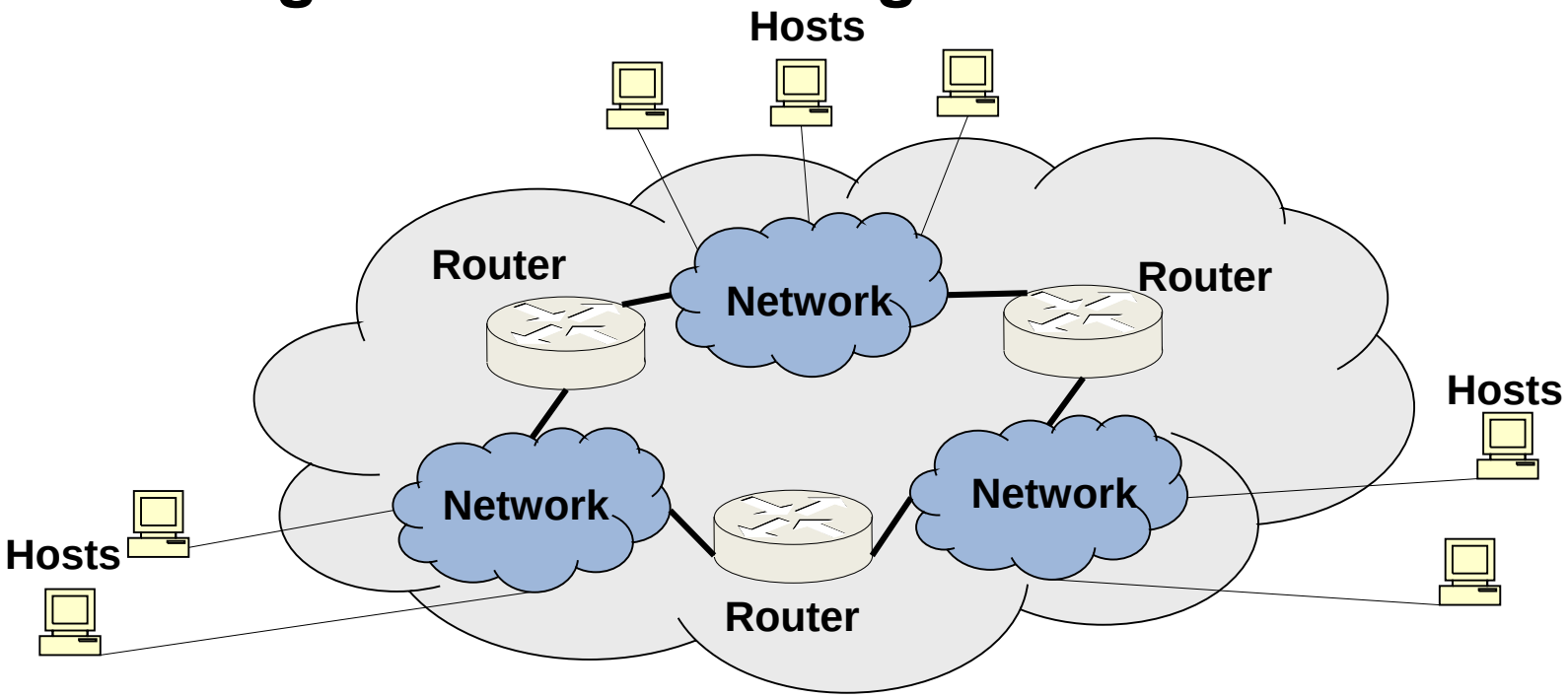
Outline

- Overview of IoT
- Application level protocols
- Network level protocols
- Lower-level protocols

IoT Overview



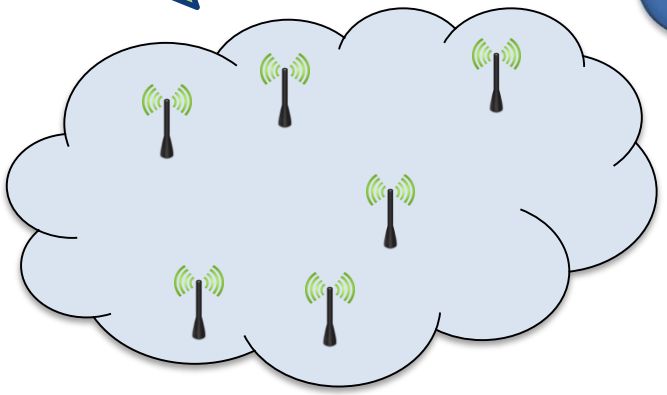
”Regular IP Networking”



- The Internet—hosts, routers, networks, and the TCP/IP protocols
- Hosts send data packets that are forwarded by routers

Internet-of-Things—”Non-computers” Connected

Constrained devices
Tiny operating system
Low-power communication



Sensor network

Small MTU, not regular TCP/IP, few ”connections” per sensor, lossy communication

Regular IP network.....

Sensor gateway

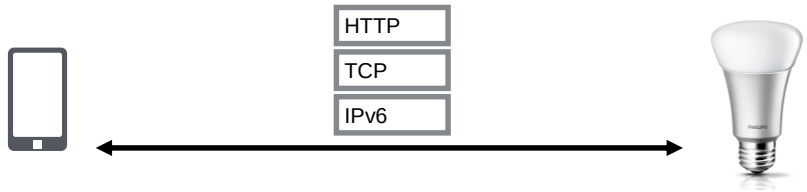
IoT Protocols

Current ”standard model”

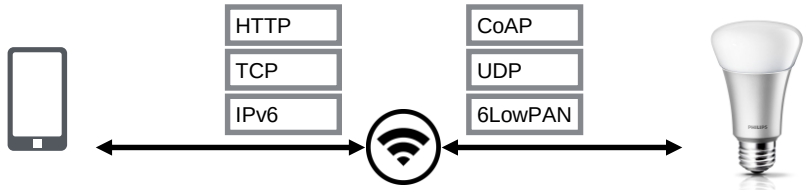
Layer	Protocol
Application	HTTP, CoAP, MQTT
Transport	TCP, UDP
Network, routing	IPv6, RPL
Adaptation	6LoWPAN
MAC	CSMA/CA
Radio Duty Cycling	ContikiMAC, TSCH
Radio	IEEE 802.15.4

IoT Protocols and ”Regular” TCP/IP

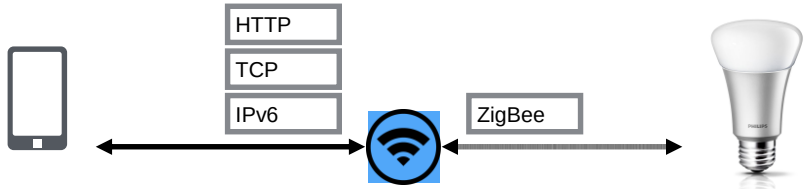
- End-to-end TCP/IP



- Gateway translates

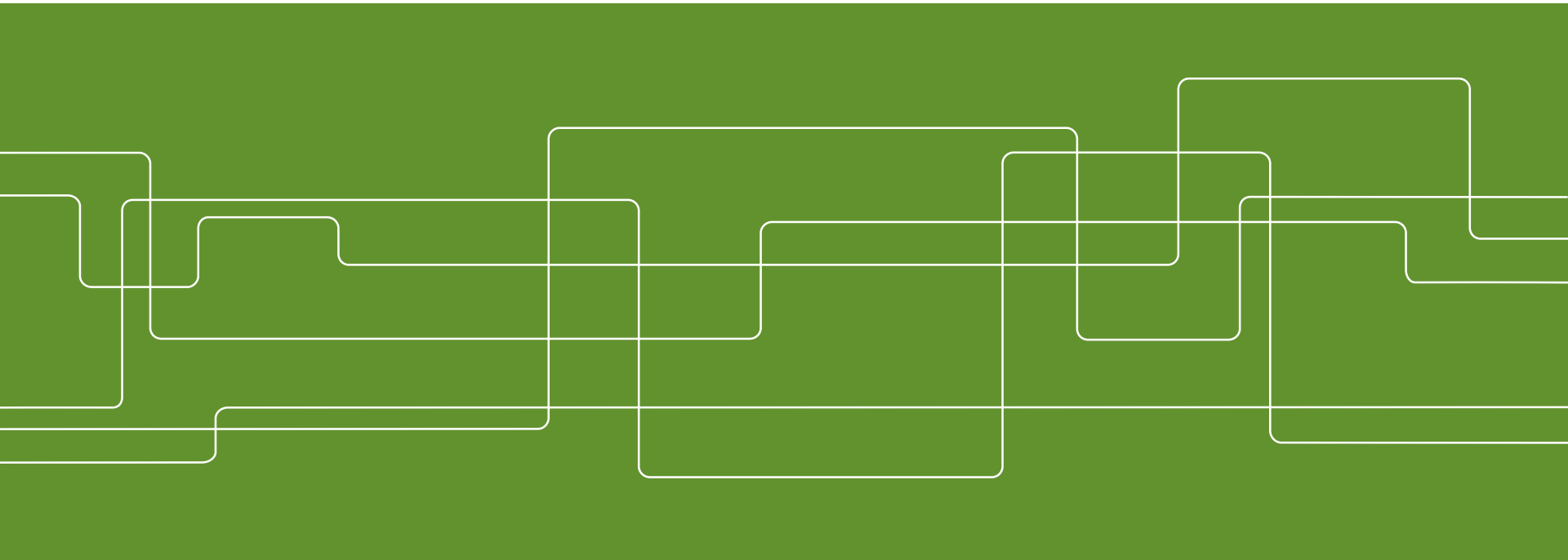


- Gateway translates



Application level protocols

CoAP and MQTT vs HTTP



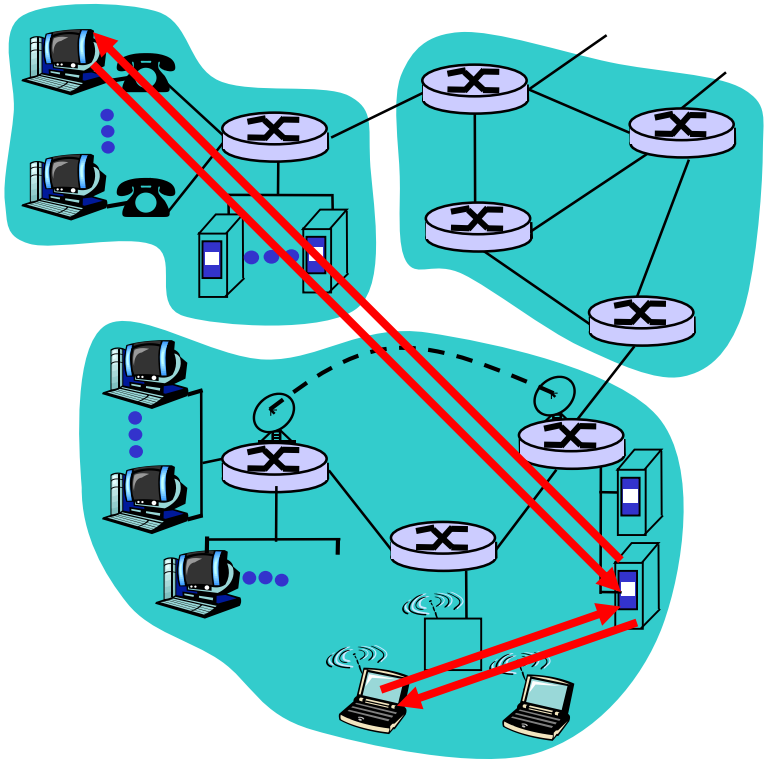
Traditional Client-Server in TCP/IP

Server:

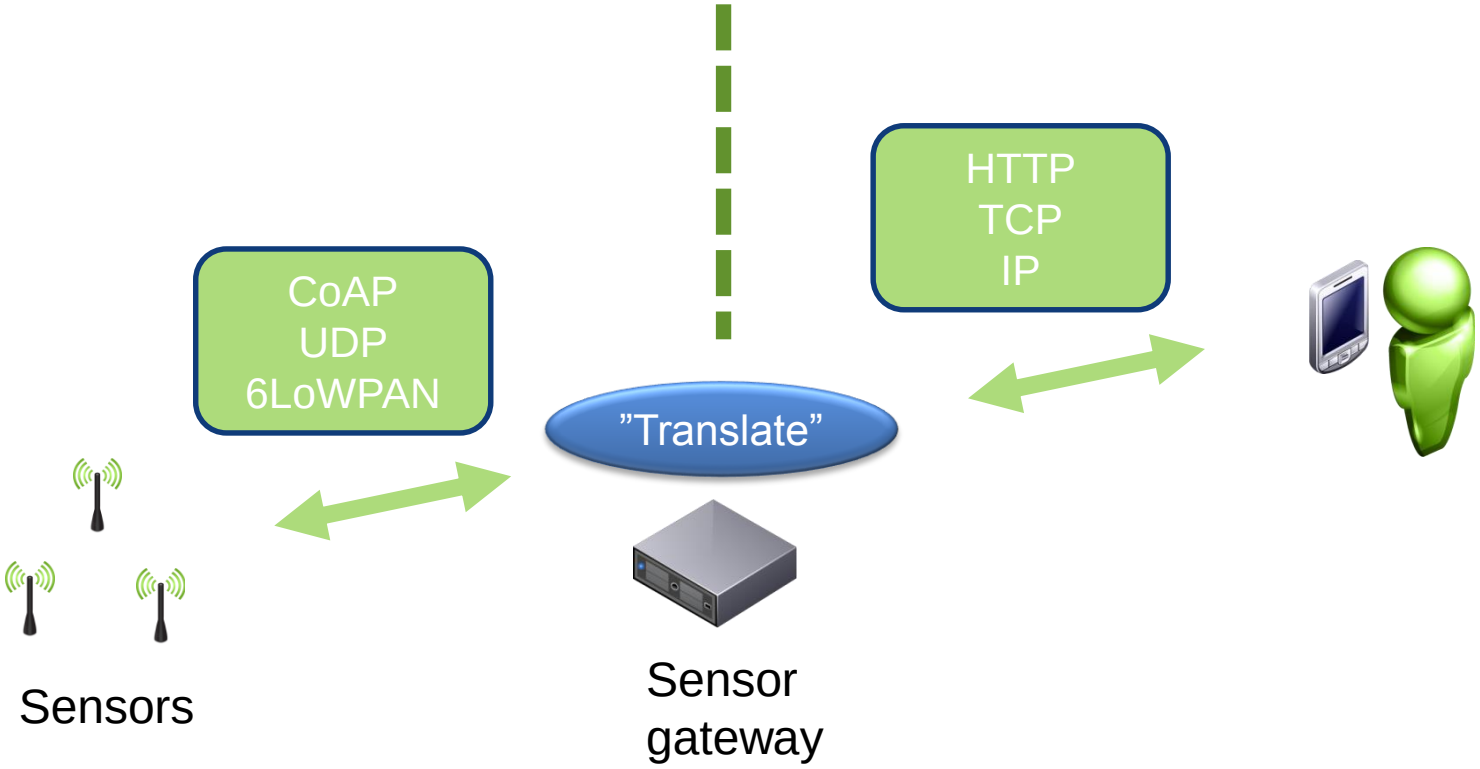
- Always-on host
- Permanent IP address

Clients:

- Communicate with server
- May be intermittently connected
- May have dynamic IP addresses



Sensor Network—Application Level



CoAP Overview

- Motivated by Internet-Of-Things (IoT)
- CoAP (Constrained Application Protocol)
  - Application protocol for constrained devices
- IETF standard RFC 7252
  - Z. Shelby (ARM), K. Hartke (University of Bremen), C. Bormann (University of Bremen)

CoAP Overview, cont'd

- Machine-to-machine communication a driving force
- Very small footprint, RAM, ROM
- URI (Uniform Resource Identifier)
  - coap://example.se:5683/~sensors./temp1.xml
  - User-agent/plugin for Firefox Copper (Cu), ETH
- CoAP implementation
  - Operating system Contiki-2.6, ETH Zurich
    - 8.5 kB ROM
    - 1.5 kB RAM

CoAP Overview, cont'd

- Resource Discovery
  - new feature to find resources (URI)
- Observe
  - new feature, report without WGET.
- UDP Port 5683 (mandatory)
- Reliable unicast
- Best effort multicast

CoAP Message Types

- Confirmable message
- Non-confirmable message
- ACK message
- Reset message
- Responses
  - Piggy-backed
  - Separate

CoAP Protocol Header (from RFC)

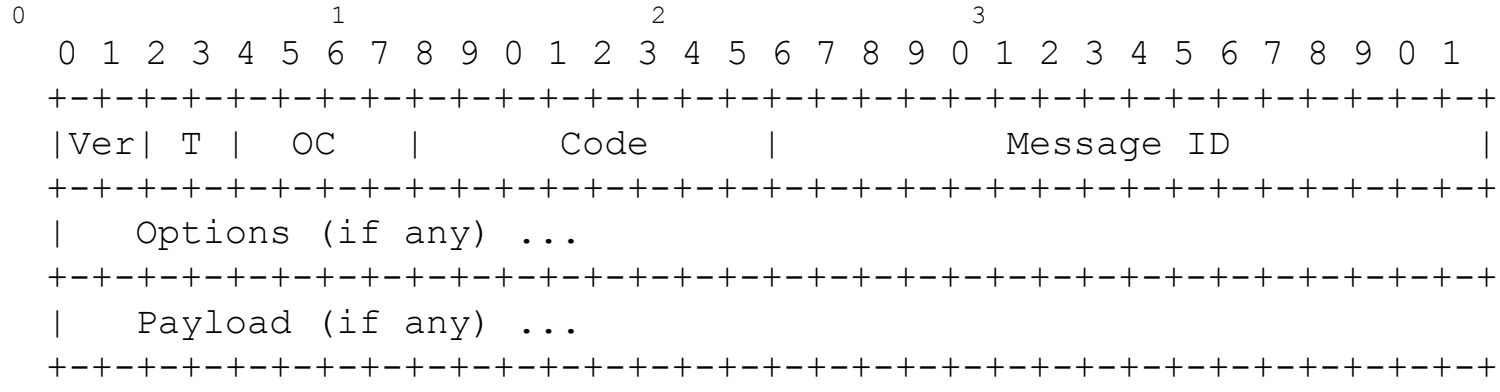


Figure 7: Message Format

3.1. Header Format

The fields in the header are defined as follows:

Version (Ver): 2-bit unsigned integer. Indicates the CoAP version number. Implementations of this specification MUST set this field to 1. Other values are reserved for future versions.

Type (T): 2-bit unsigned integer. Indicates if this message is of type Confirmable (0), Non-Confirmable (1), Acknowledgement (2) or Reset (3). See Section 4 for the semantics of these message types.

Option Count (OC): 4-bit unsigned integer.

CoAP Example (from RFC)

```
Client  Server
       |       |
       |       |
       +------>|     Header: GET (T=CON, Code=1, MID=0x7d34)
       | GET   |   Uri-Path: "temperature"
       |       |
       |       |
       |<------+     Header: 2.05 Content (T=ACK, Code=69, MID=0x7d34)
       | 2.05  |    Payload: "22.3 C"
       |       |
```

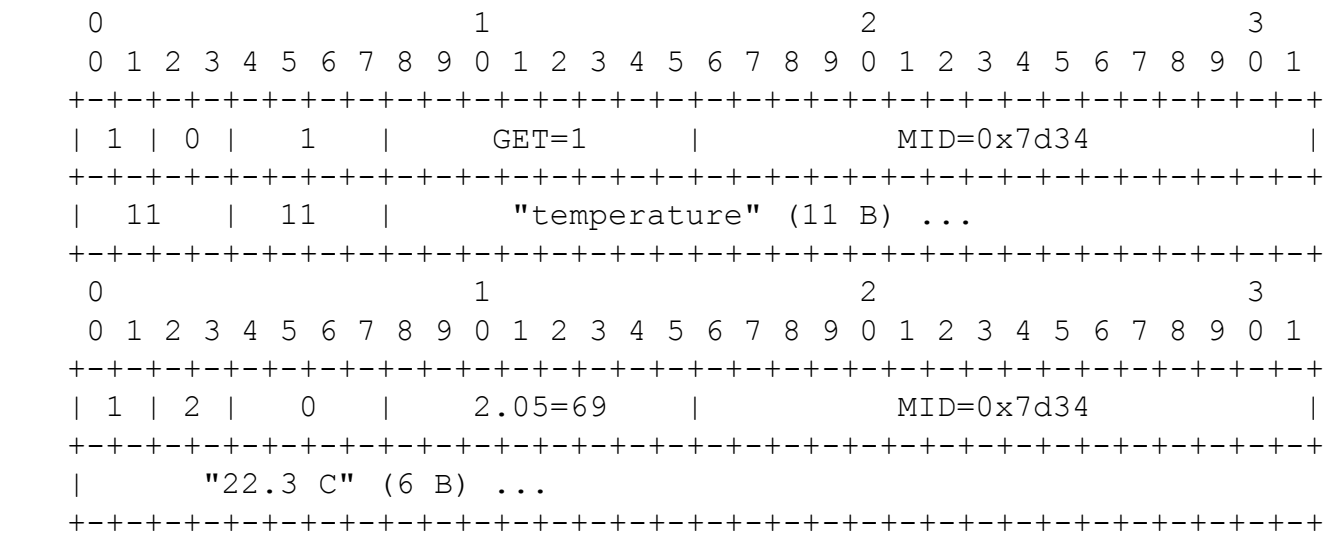
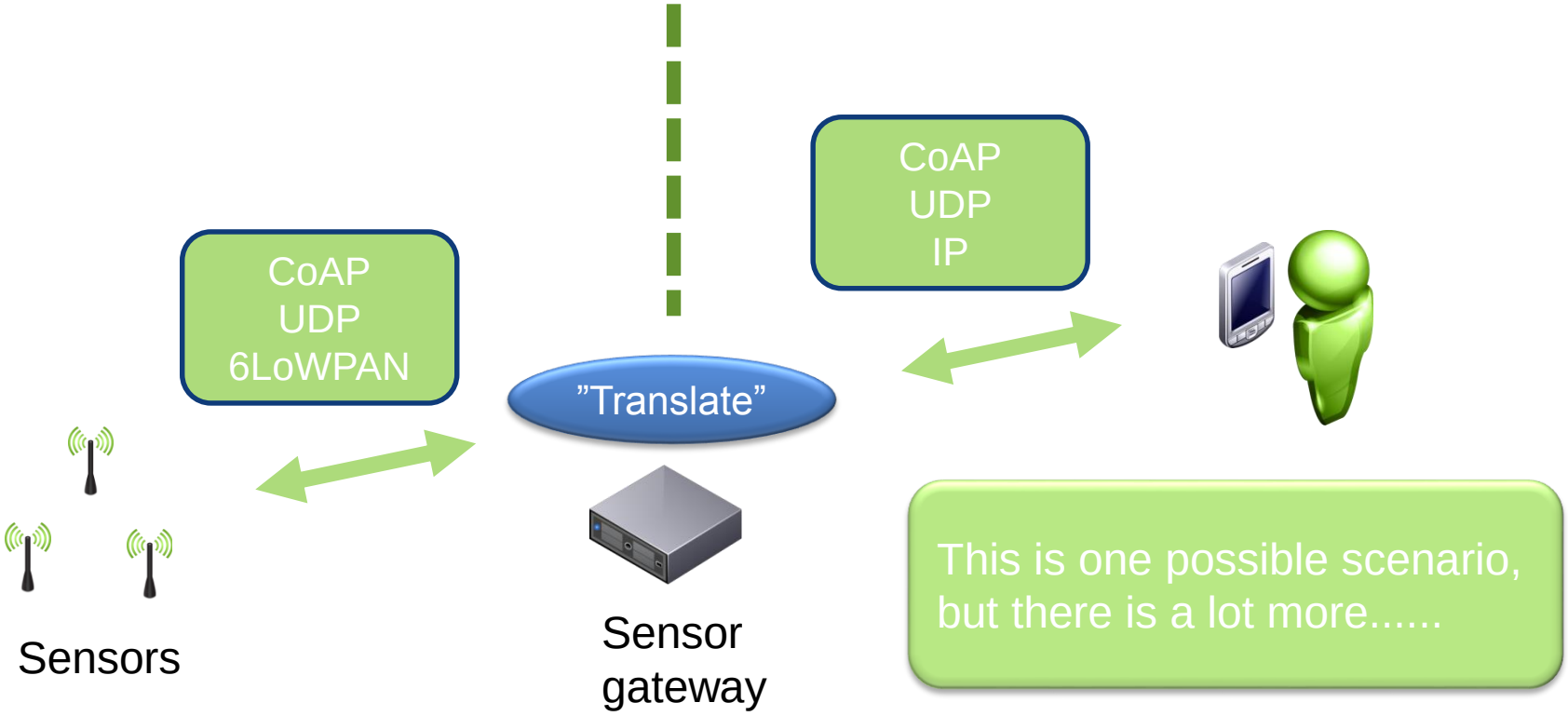
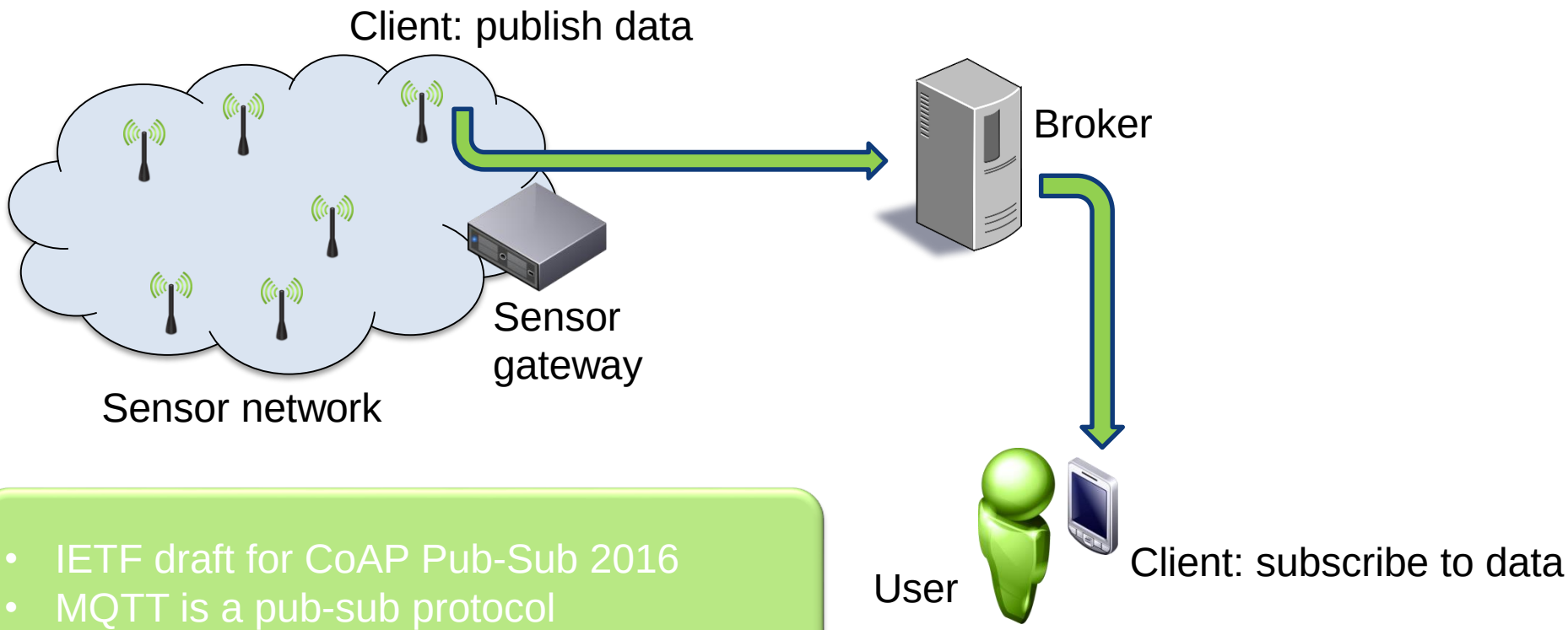


Figure 16: Confirmable request; piggy-backed response

Sensor Network—Application Level Revisited



Publish-Subscribe Model

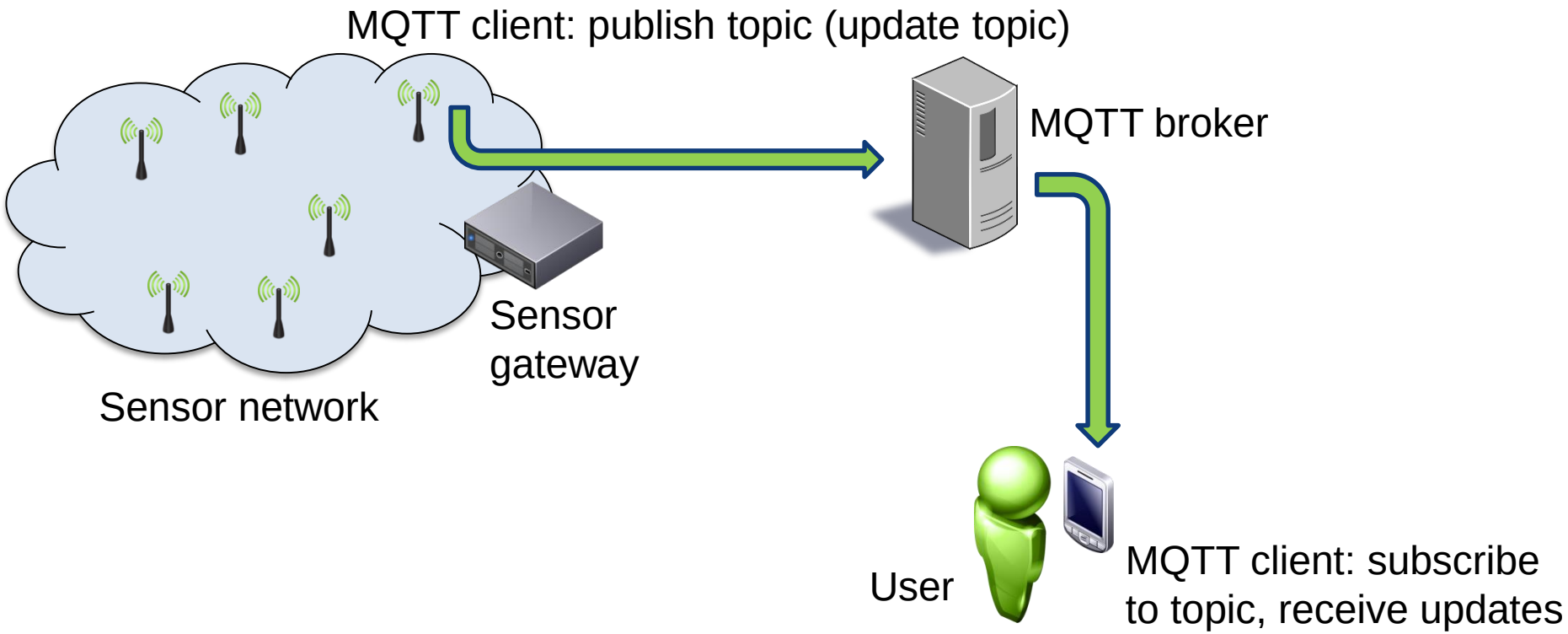


- IETF draft for CoAP Pub-Sub 2016
- MQTT is a pub-sub protocol

MQTT: Message Queue Telemetry Transport

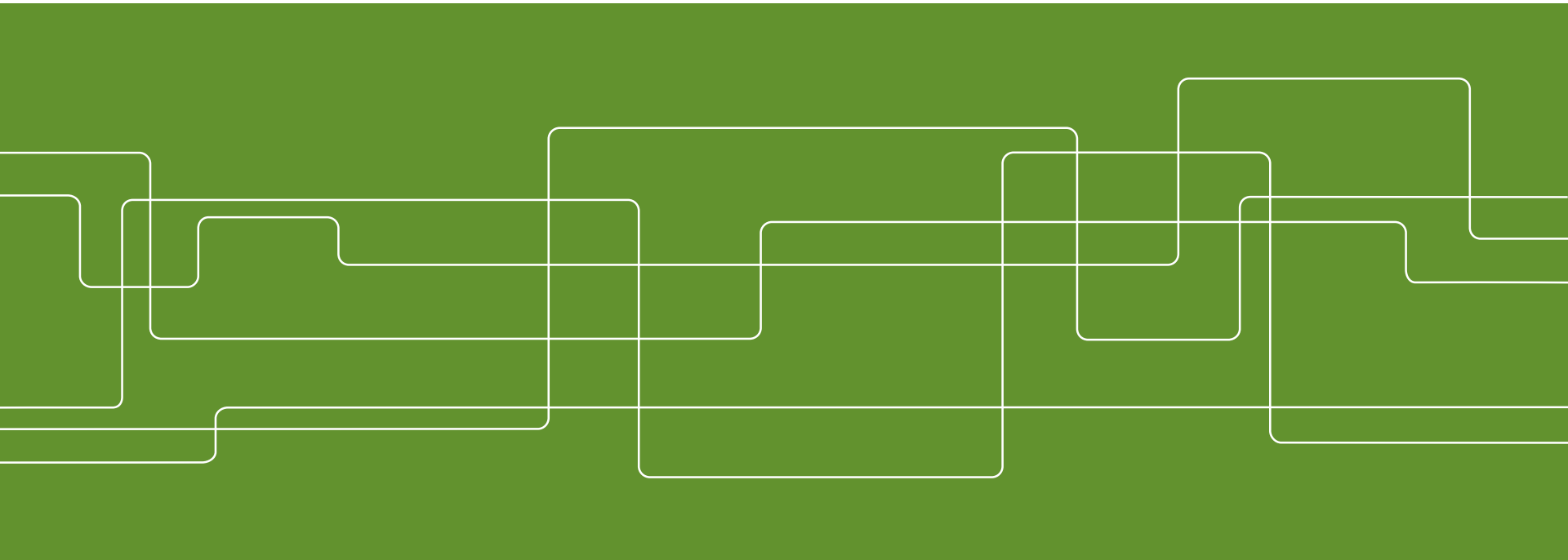
- Invented by Andy Stanford-Clark and Arlen Nipper
  - IBM, 1999
- Open source publish/subscribe protocol
  - Runs over TCP
- Data is relayed by a broker
- Devices can publish and subscribe to different ”topics”
  - Wildcard # can be used to subscribe to all
- Once a topic is updated, subscribers get notified and get data via the broker

MQTT and Publish-Subscribe



Network level protocols

IPv6 and RPL



IPv6—Stateless Autoconfigurtion

Stateless Autoconfiguration (SLAAC)

- Initially developed for IPv6, but later designed also for IPv4
- Server keeps no state about hosts, only non-host state
  - Like global unicast prefix and subnet prefix
- Uses IPv6 concept of link local addresses to enable clients to communicate on local subnet before having a global address
- Client does not explicitly request address from server
  - Does not even explicitly inform the server of address selected

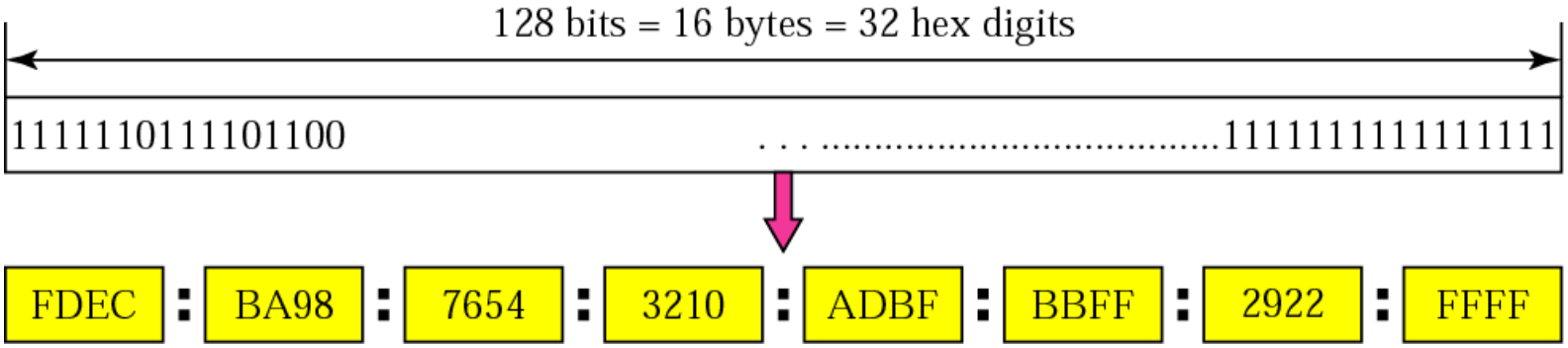
IPv6 Addressing, Revisited

An IPv6 unicast address identifies an interface connected to an IP subnet (as is the case in IPv4)

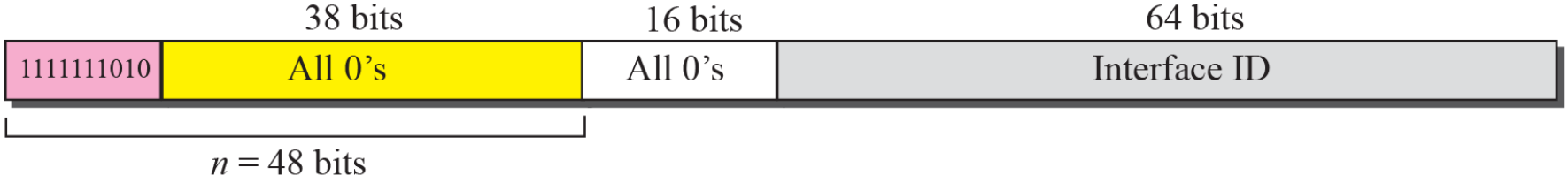
IPv6 routinely allows each interface to be identified by several addresses

- facilitates management

Colon hexadecimal notation (eight 16 bit hexadecimal integers)



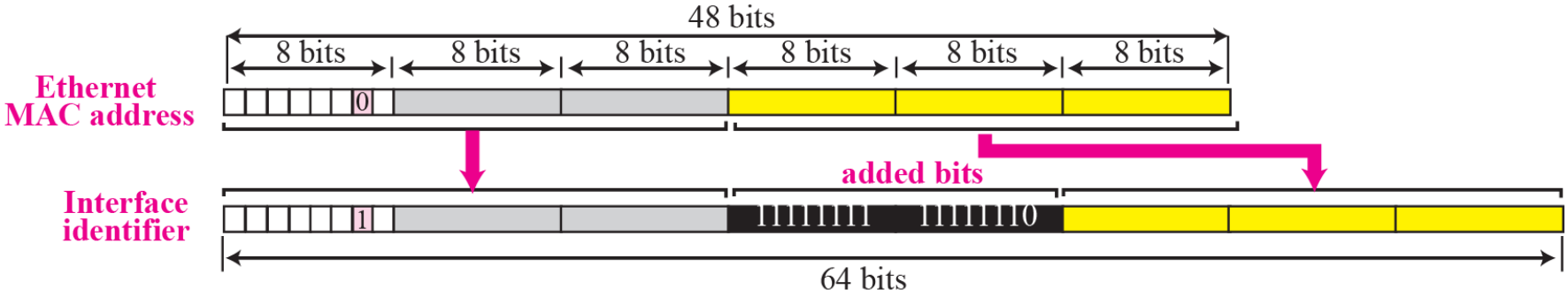
IPv6 Link Local Address in SLAAC



Host generates link local address (LLA)

- From the interface MAC address
- Or using random number (IPv6 privacy extensions)

Use ICMP messages to probe local subnet and detect if generated LLA already in use



SLAAC, cont’d

Once unique LLA is established, this can be used to communicate with other hosts on same subnet

Routers on local subnet broadcast periodic Router Advertisements (RA) ICMP messages

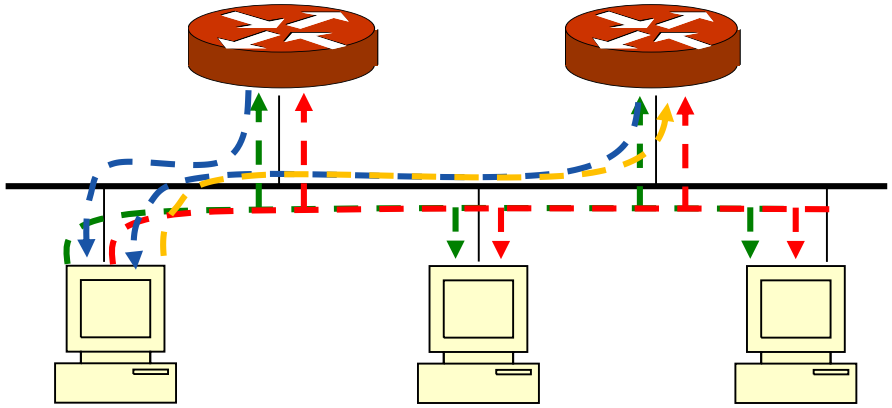
- Hosts can also send an ICMP message to solicit announcements from routers on local subnet
- RA contains the global unicast prefix and subnet prefix

Client combines LLA with prefixes to create its unique global address and directly starts using it

Stateless Autoconfiguration—Example

1 (green). Host forms link-local tentative address.
Sends neighbor solicitation: Duplicate address detection

4 (orange). Optionally: proceed with stateful configuration



2 (red). Assign link-local address to interface.
Send router solicitation to all routers

3 (blue). Get router advertisement from Routers. Get prefixes, form addresses, default nexthop. Duplicate address detection

Routing over Low-power Lossy Networks (LLN)

- Challenging long history
- Routing Over Low power and Lossy networks (ROLL)
- Radio intended to cover only 5-10 m—routing essential
- Different metrics needed
  - Power, Radio Quality, Latency, etc
- Current protocol: RPL RFC 6550 IETF 2012 (New)

RPL Introduction

- RFC 6550 (157 pages)
- RPL: IPv6 Routing Protocol for Low-Power and
  - Lossy Networks (LLNs)
- RPL is a distance vector protocol
- Runs in host and routers
  - RPL messages encapsulated in ICMPv6
- From 10s to thousands of routers
- Separate routing optimization (power, latency loss etc)
  - via Objective Function (OF) from packet handling.
- RPL node often combines host and router

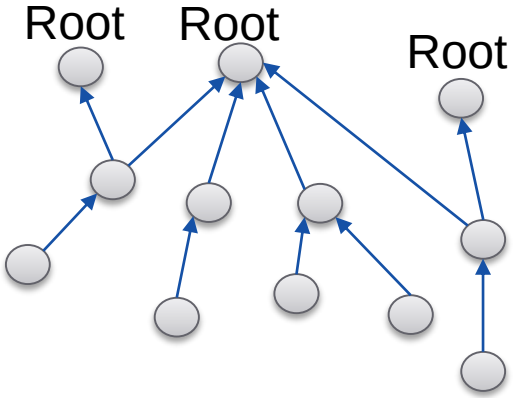
RPL Introduction, cont'd

- OF gives a ”rank” used for tree/parent selection and loop prevention
  - Rank decreases towards destination
- RPL is not designed for a specific link layer
- Typically lossy wireless networks or Power Line Communication (PLC)
- IPv6 Neighbor Discovery (ND) may by replaced by RPL

RPL Tree Terminology

DAG:

Directed Acyclic Graph. A directed graph having the property that all edges are oriented in such a way that no cycles exist. All edges are contained in paths oriented towards and terminating at one or more root nodes.

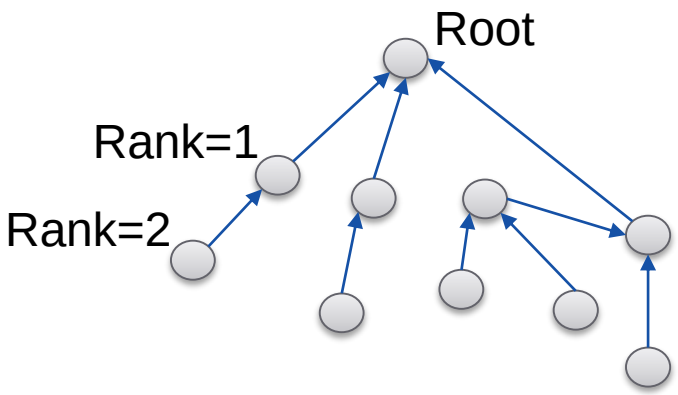


Reworked based on original from Prof Raj Jain, Washington University in St. Louis

RPL Tree Terminology, cont'd

Destination-Oriented DAG (DODAG):

A DAG rooted at a single destination, i.e., at a single DAG root (the DODAG root) with no outgoing edges.

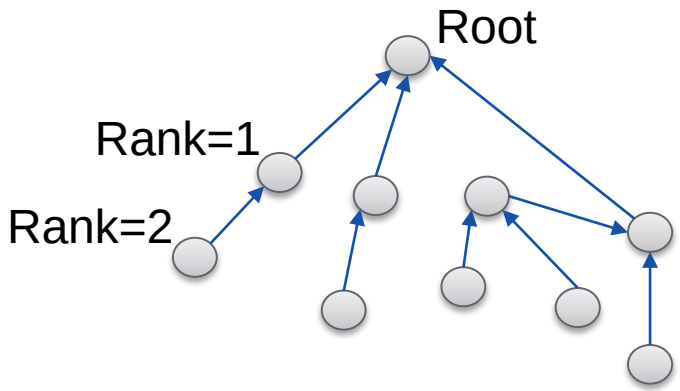


Reworked based on original from Prof Raj Jain, Washington University in St. Louis

RPL Tree Terminology, cont'd

DODAG root:

A DODAG root is the DAG root of a DODAG. The DODAG root may act as a border router for the DODAG; in particular, it may aggregate routes in the DODAG and may redistribute DODAG routes into other routing protocols.



Reworked based on original from Prof Raj Jain, Washington University in St. Louis

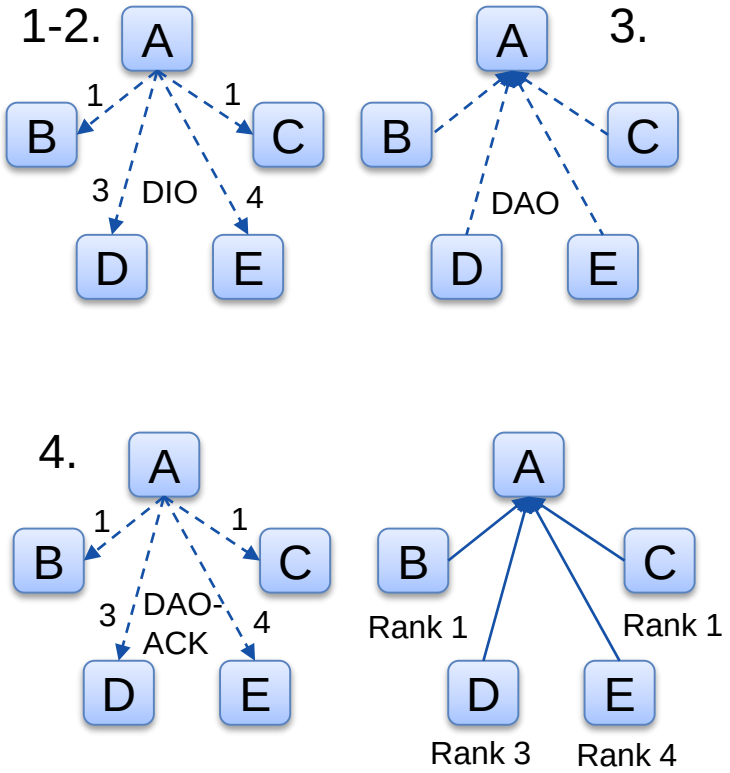
RPL Core Object Messages

- DIO: DODAG Information Object
- DAO: Destination Advertisement Object
- DAO-ACK: Destination Advertisement Object Acknowledgement
- DIS: DODAG Information Solicitation
- CC: Consistency Check (Not covered here)

RPL Message Exchange—DODAG Formation

1. A multicasts DIOs, itself with Rank 0
2. B, C, D, E receive DIOs and determine their rank (1, 1, 3, 4)
3. B, C, D, E send DAOs to A
4. A accepts DAOs

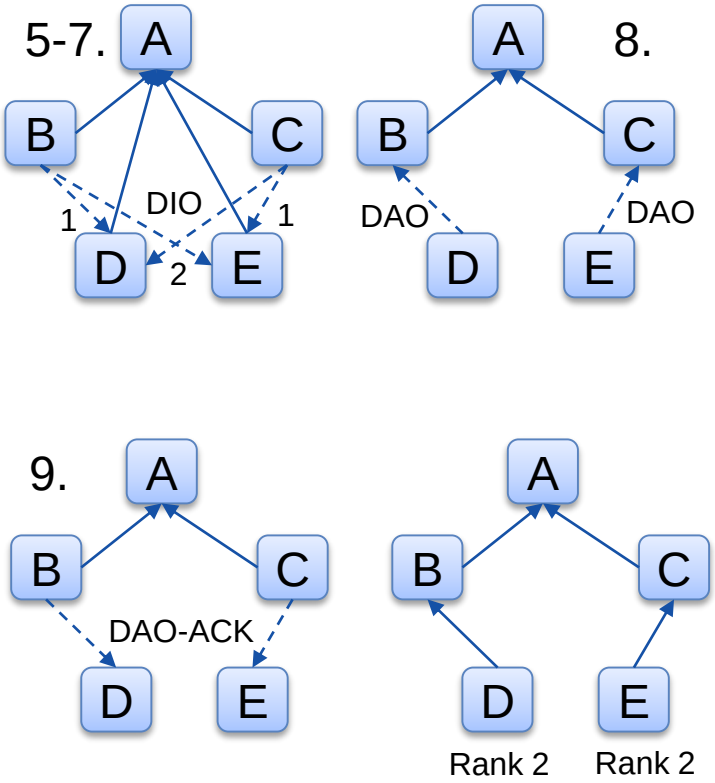
.... to be continued...



Reworked based on original from Prof Raj Jain, Washington University in St. Louis

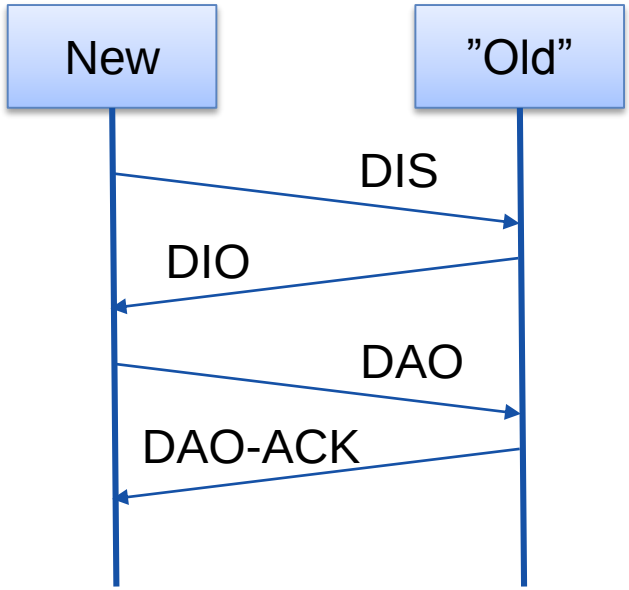
DODAG Formation, cont'd

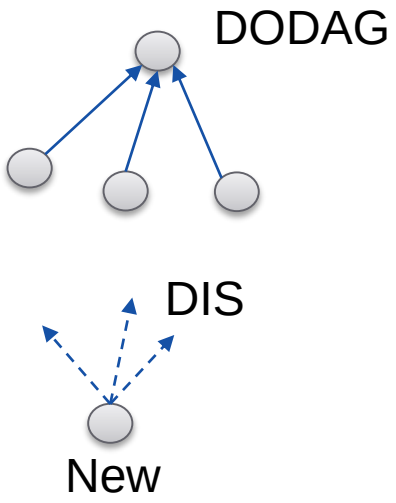
5. B and C multicast DIOs
6. D receives DIOs and determines rank is 1, 2
7. E receives DIOs and determines rank 2, 1
8. D sends DAO to B
   E sens DAO to C
9. B sends DAO-ACK to D
   C sends DAO-ACK to E



Reworked based on original from Prof Raj Jain, Washington University in St. Louis

RPL Message Exchange—New Node Joins DODAG

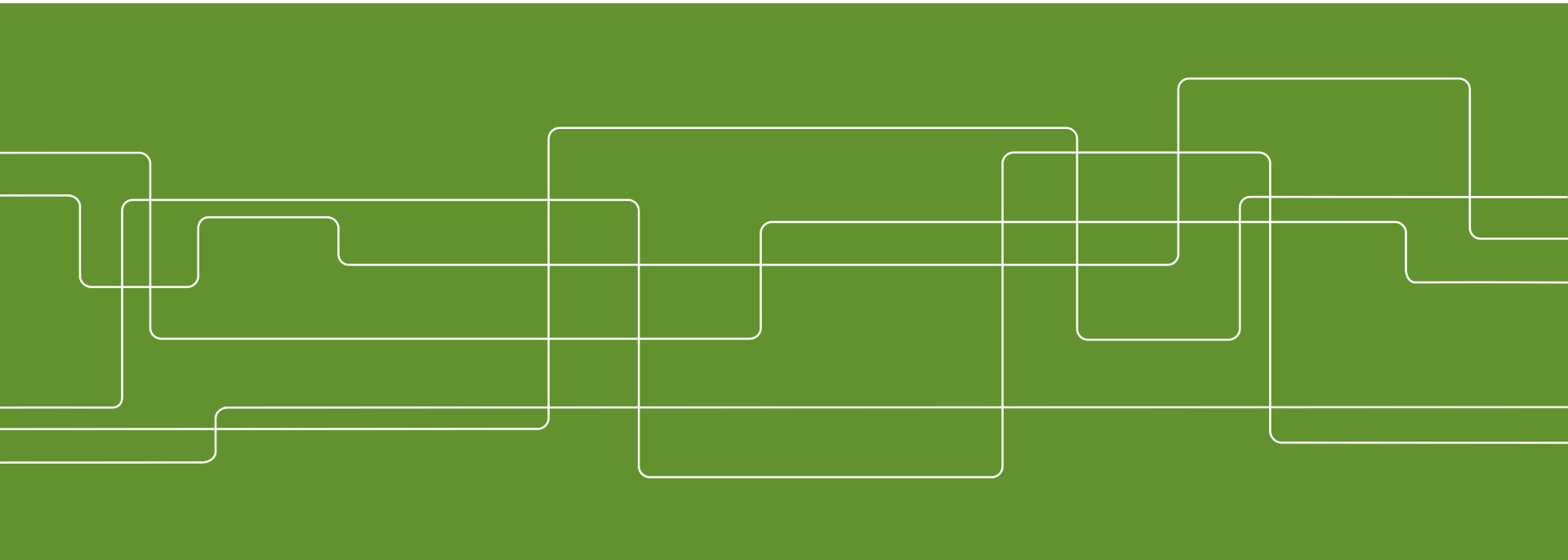




Reworked based on original from Prof Raj Jain, Washington University in St. Louis

Lower level protocols

6LowPAN, IEEE 802.15.4



6LowPAN Adaptation Layer

- IPv6 over Low Power Personal Area Networks
- RFC 4944 (original), RFC 6282 (header compression)
- Header compression
- Fragmentation and reassembly (small MTU)
- Stateless autoconfiguration
  - Duplicate address detection (DAD)

Makes it possible to run IPV6 on small constrained low-power devices!!

6LowPAN Encapsulation—Stacked Headers

- Mesh addressing header
  - L2 forwarding, several hops in 6LowPAN network
  - Hop limit, source address, destination address
  - IEEE 802.15.4 addresses (or short 16-bit addresses)
- Fragment header
  - When payload is too large for IEEE 802.15.4 MTU
  - Size, tag (ID for fragment set), offset
- IPv6 header compression

Thank You!