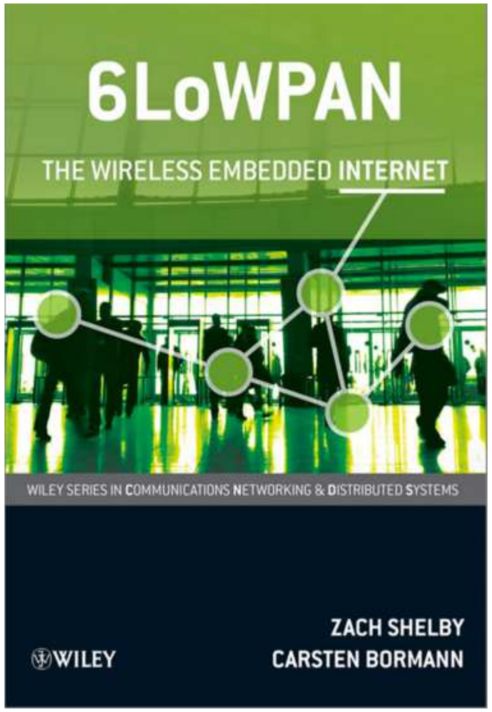
University of Duisburg-Essen
Networked Embedded Systems Group
Institute for Computer Science and
Business Information Systems (ICB)
Schützenbahn 70
D-45127 Essen, Germany

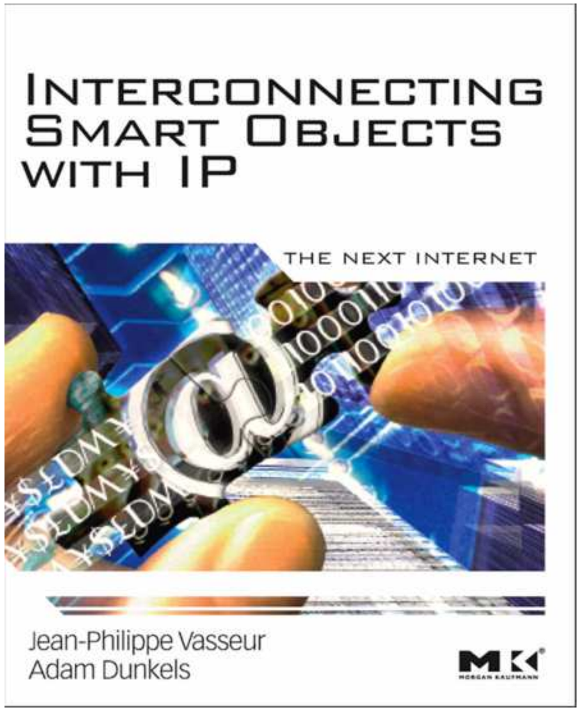
Kommunikationsnetze 2
9 – Internet of Things

Prof. Dr. Pedro José Marrón

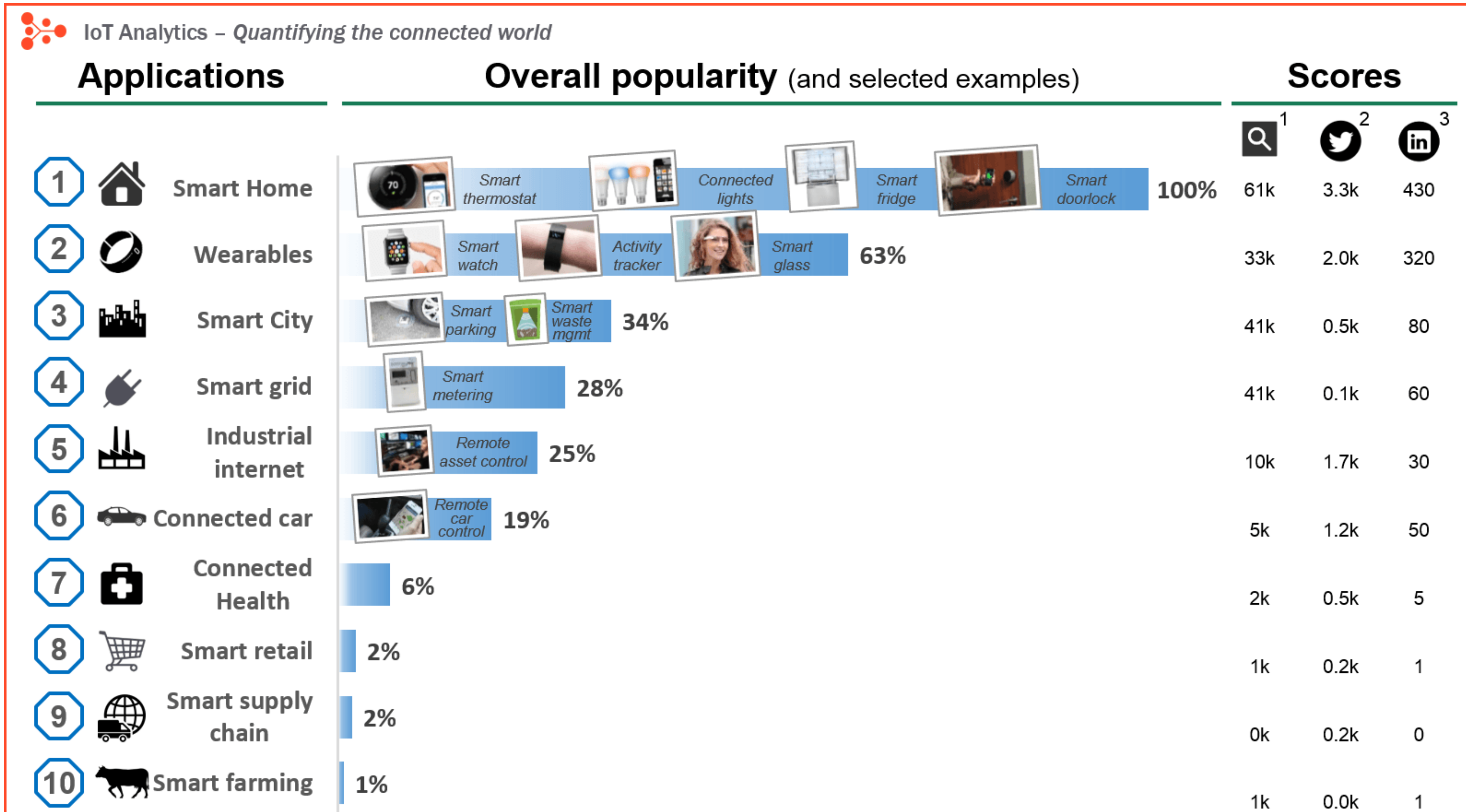
Literature and Acknowledgements

- Z. Shelby, C. Bormann. 6LoWPAN: Embedded Internet
  - PDF online available
- J.P. Vasseur, A. Dunkels. Interconnecting Smart Objects with IP – The Next Internet
  - PDF online available
- Part of the slides are a revision of the material from the Embedded Internet course at TU Graz, taught by Carlo Alberto Boano





What is the Internet of Things?



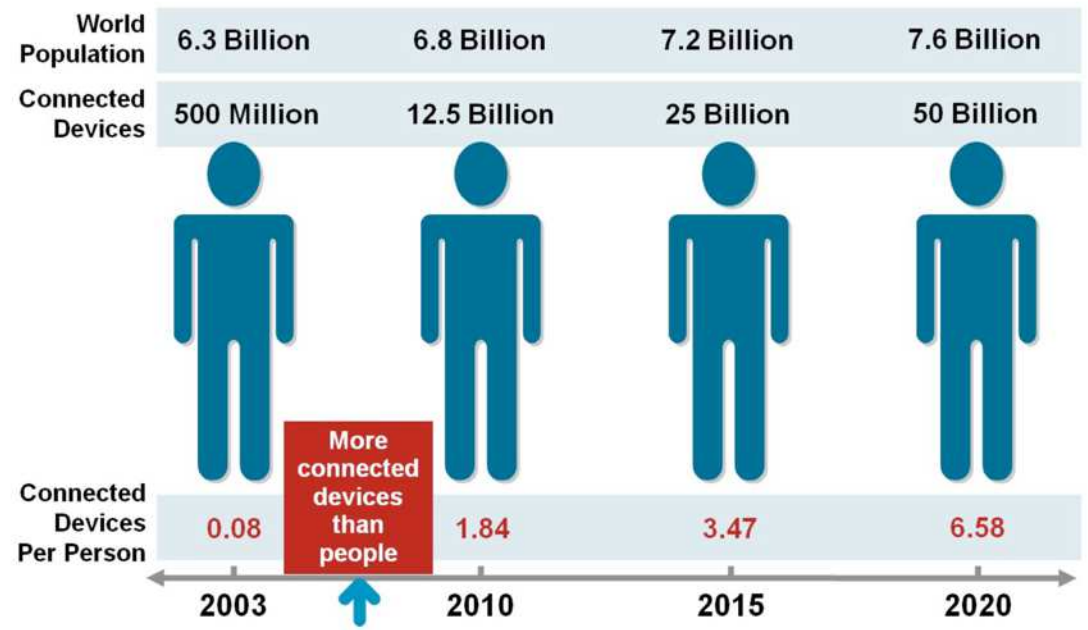
1. Monthly worldwide Google searches for the application 2. Monthly Tweets containing the application name and #IOT 3. Monthly LinkedIn Posts that include the application name. All metrics valid for Q4/2014.
Sources: Google, Twitter, LinkedIn, IoT Analytics

https://iot-analytics.com/10-internet-of-things-applications/

The Internet of Things

- From the Internet of People to the Internet of Embedded Devices
  - Everything is connected
- Enablers
  - Computing power
  - Progress in communication technologies
  - Better and cheaper sensors
  - New materials
- Obstacles
  - Batteries and their capacity
  - Scalability
  - Maintainability
  - Privacy and safety concerns

Cisco IBSG, April 2011



Many Names for the Same Paradigm

- Ubiquitous Computing (Marc Weiser, Xerox PARC, 1988)
  - “The most profound *technologies* are those *that disappear*. They weave themselves into the fabric of everyday life until they are indistinguishable from it”
- Internet of Things (Kevin Ashton, Auto-ID@MIT, 1999)
  - “We need an internet for things, *a standardized way for computers to understand the real world*”
- Ambient Intelligence (Emile Aarts, 1999)
  - “In an ambient intelligence world, *devices work in concert to support people* in carrying out their everyday life activities, tasks and rituals in an easy, natural way using information and intelligence that is hidden in the network”
- Internet of Everything (Cisco, 2013)
  - “The Internet of Everything (IoE) brings together people, processes, data, and things to make networked connections more relevant and valuable than ever before – *turning information into actions* that create new capabilities, richer experiences, and unprecedented economic opportunity for businesses, individuals, and countries.”
- …, Wireless Sensor Networks, Cyber-Physical Systems, …

History: From Early Visions to the First Connected Objects

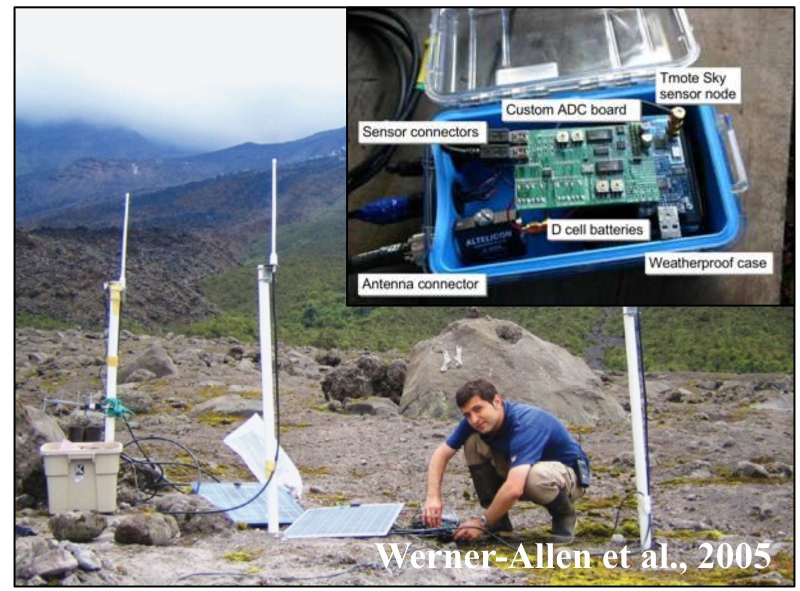
- 1990: A toaster connected to the Internet (Romkey and Hackett)
  - A human had to insert bread… until 1991
- 2000: LG Internet refrigerator
  - Unsuccessful: expensive and unnecessary
- 2001/2002: Mobile phones
- 2008: More connected devices than people

- 2001 – 201X: Many prototypical deployments

History: First Deployments for Scientists

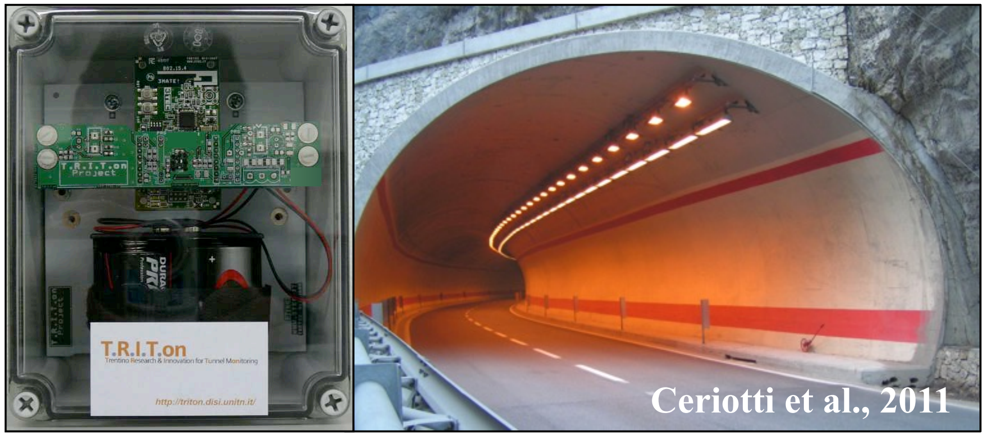

Mainwaring et al., 2002


Martinez et al., 2004



Werner-Allen et al., 2005

History: First Deployments for Society



Ceriotti et al., 2011


Feltrin et al., 2013


Thingsquare, 2014

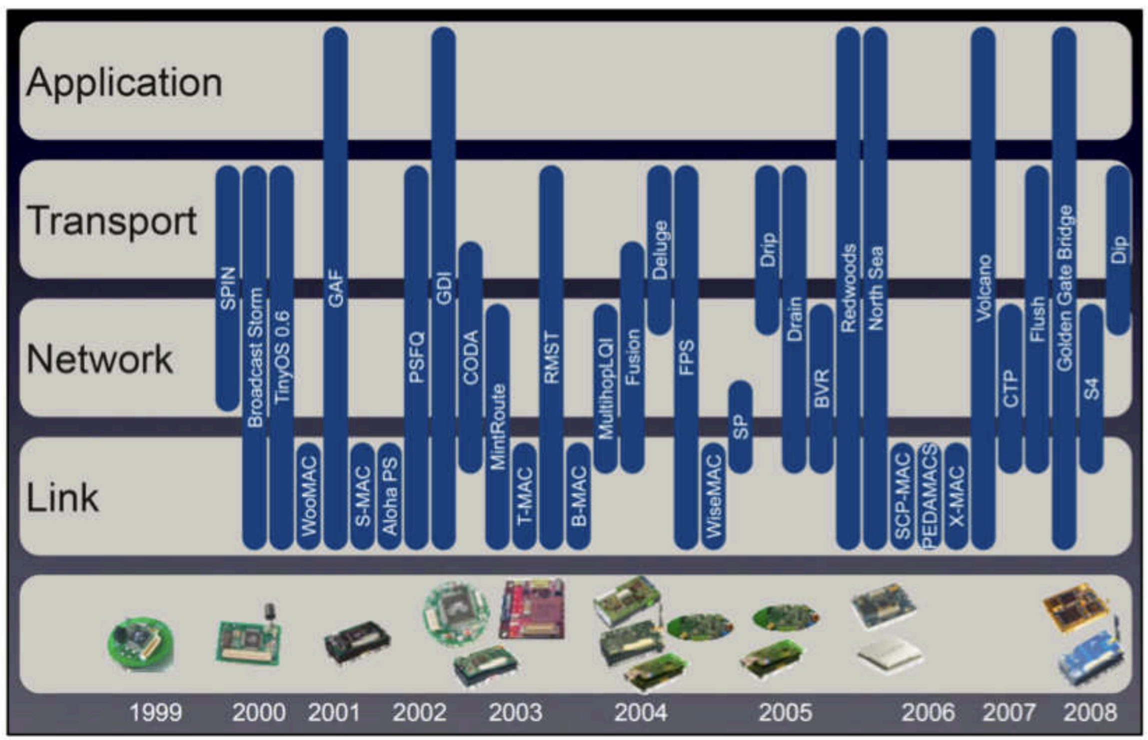
Today


63


Keith Smiley
@SmileyKeith
Follow
Got in our @Zipcar we've been driving for 3 days. It doesn't start. Call Zipcar and get told that the car can't access the internet so it won't start. The only option is to have a tow truck come and tow it to somewhere where it has internet.
2:28 PM - 27 Dec 2018
1,100 Retweets 3,280 Likes
141 1.1K 3.3K
Tweet your reply
Keith Smiley @SmileyKeith · 28 Dec 2018
Update: we moved the car 4 feet and it started
8 36 359

Towards IoT Networking: First Solutions

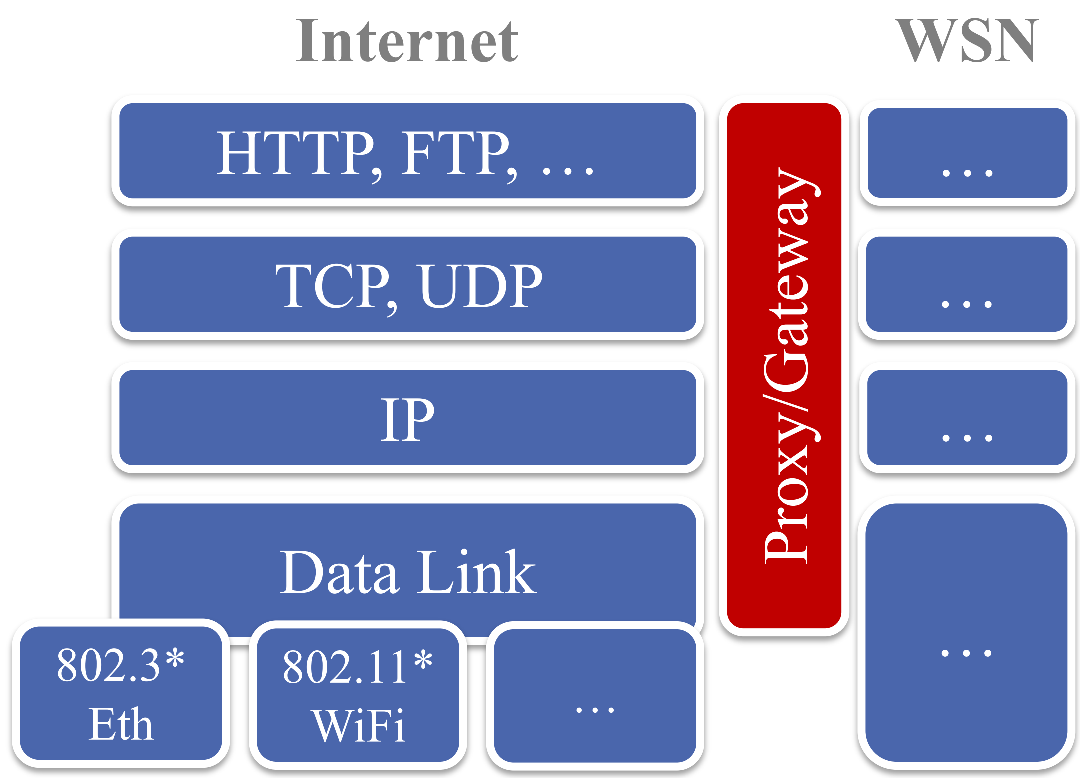
- The alphabet soup of different solutions
  - Isolated networks
  - Application and scenario specific solutions
  - Focus: constrained resources and network-level challenges



https://cms.hit.edu.cn/pluginfile.php/90655/mod_resource/content/1/8-ipv6.pdf

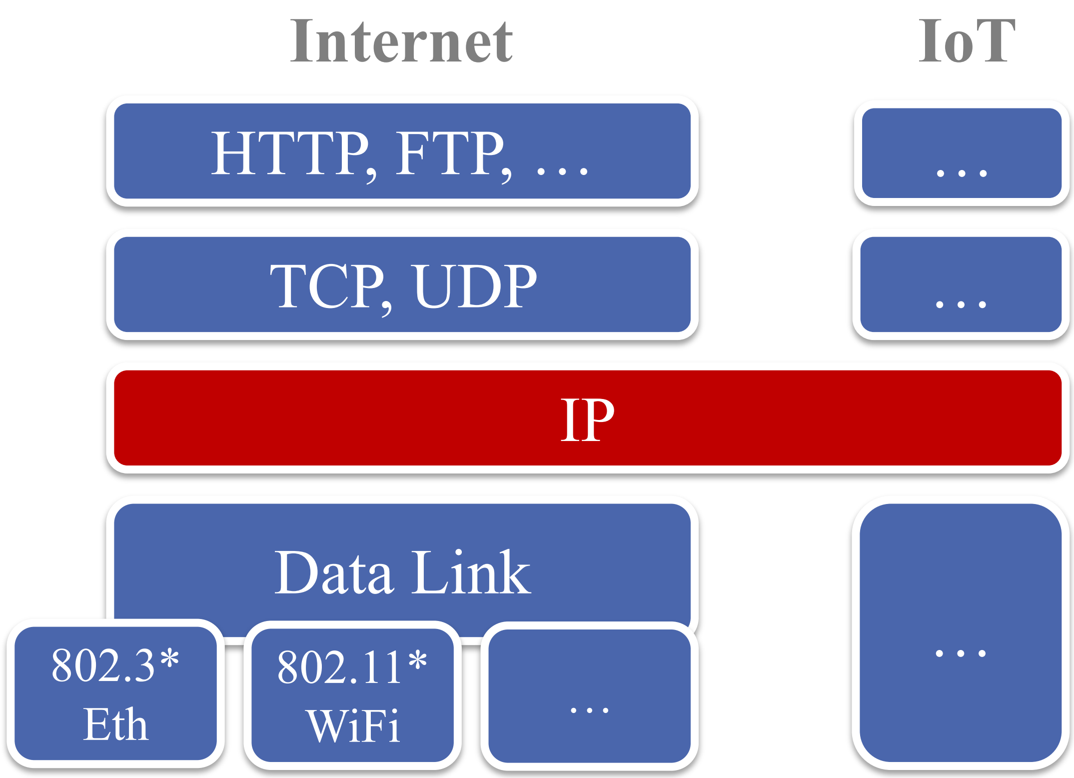
Towards IoT Networking: The Edge Network

- Use proxy/gateway to bridge low-power wireless networks and the Internet
- Problems:
  - Gateways break the networking model
  - Lack of flexibility, scalability and reliability
  - Inherent complexity



Why Not Using IP Instead?

- IP is ubiquitous
- From vertical, ad-hoc solutions to a common network and service infrastructure
- Standards foster adoption and provide interoperability
- Easy learning curve
- End-to-end connectivity



Why not? i.e. Challenges

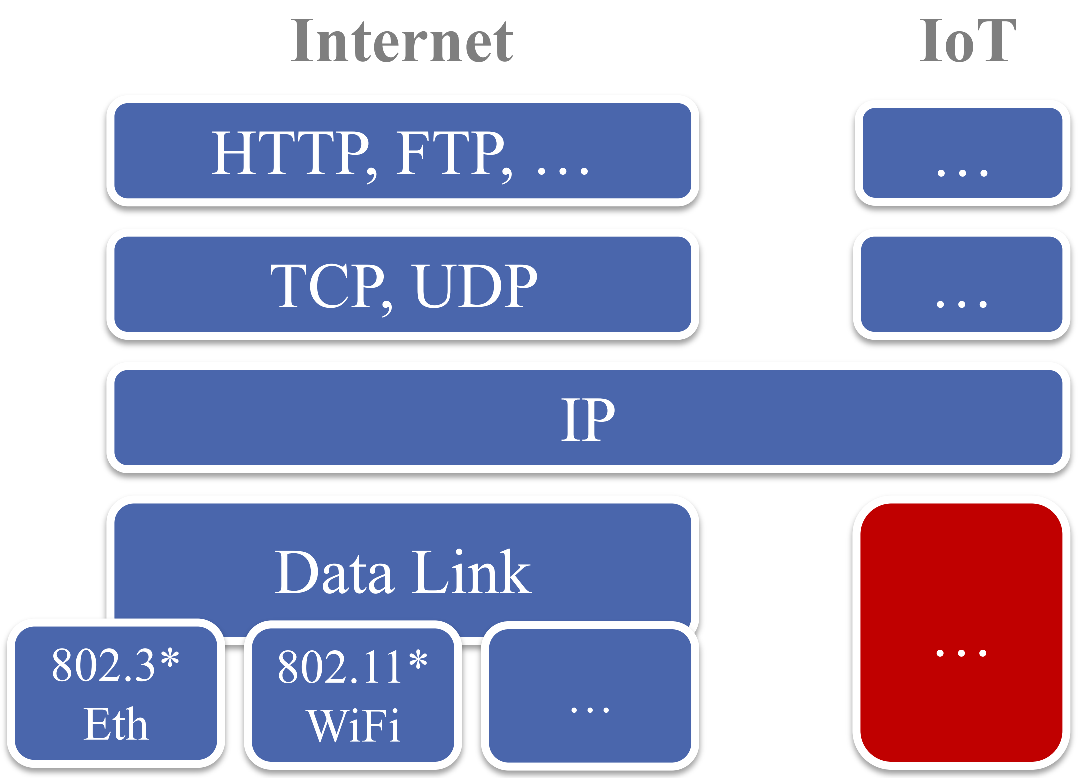
- Minimum IPv6 MTU: 1280 bytes
- Small packet size (802.15.4)
  - Frame size: 127 bytes (Ethernet 1500 Bytes)
  - Payload 102 bytes
  - Payload with security: 81 bytes
- Mismatched header sizes
  - IPv6: 40 bytes
- 64 bit MAC addresses (support for 16 bit)
- Data rate up to 250 kbps
- RAM/ROM constraints on node
- Lossy links
- Dynamic changes in topology

μIP

- Support for IP, UDP and TCP (minimal functionality only)
- Developed for 8-bit and 16-bit devices
- First release 2001
- 3-5 KB code, 100 bytes-2KB RAM (configurable)
- How
  - Shared packet buffer
  - Low throughput
  - No sockets, instead proprietary event-driven API
- Later extended for IPv6
  - A few KB of RAM, a few tens of KB of ROM

Link-Layer Technologies for Smart Objects

- Wireless
  - IEEE 802.15.4
  - Bluetooth LE
  - IEEE 802.15.7 (VLC)
  - Low-Power WiFi
  - Long-Range Wireless
- Wired
  - Powerline Communication

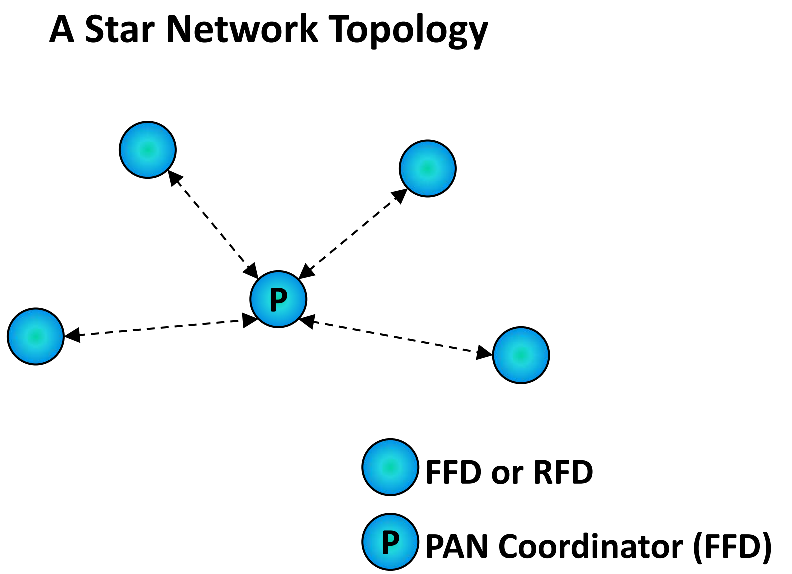


IEEE 802.15.4

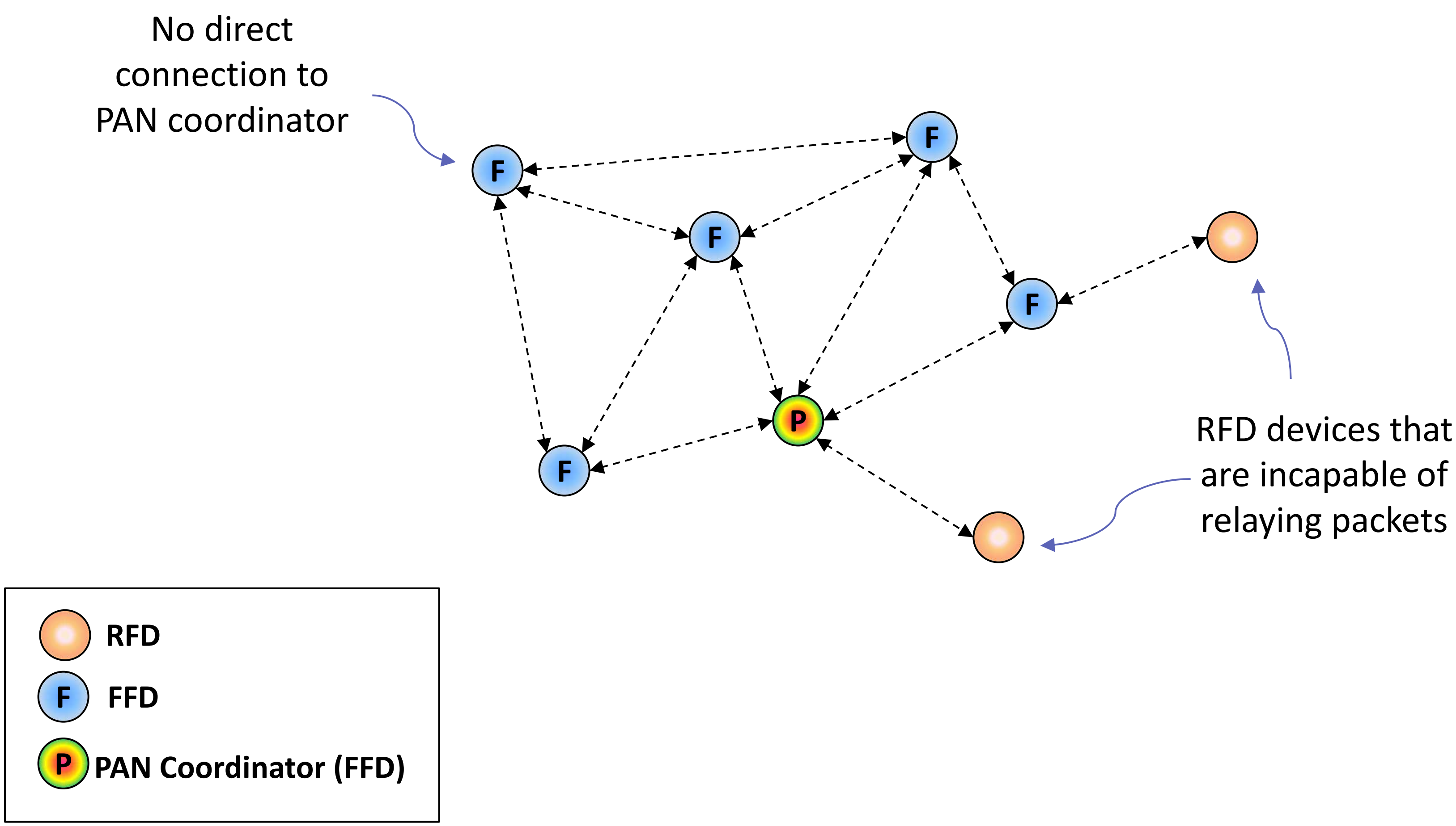
- Still one of the reference standard to build IoT
  - Target: low-power & low-data-rate applications
  - Low-cost radio transceivers
  - Data rates up to 20/40 kb/s (868/915 MHz), 250 Kb/s (2.4 GHz)
  - Specifies PHY & MAC layer
  - Main features
    - Transceiver management
    - Channel access
    - PAN management

IEEE 802.15.4: Device Types

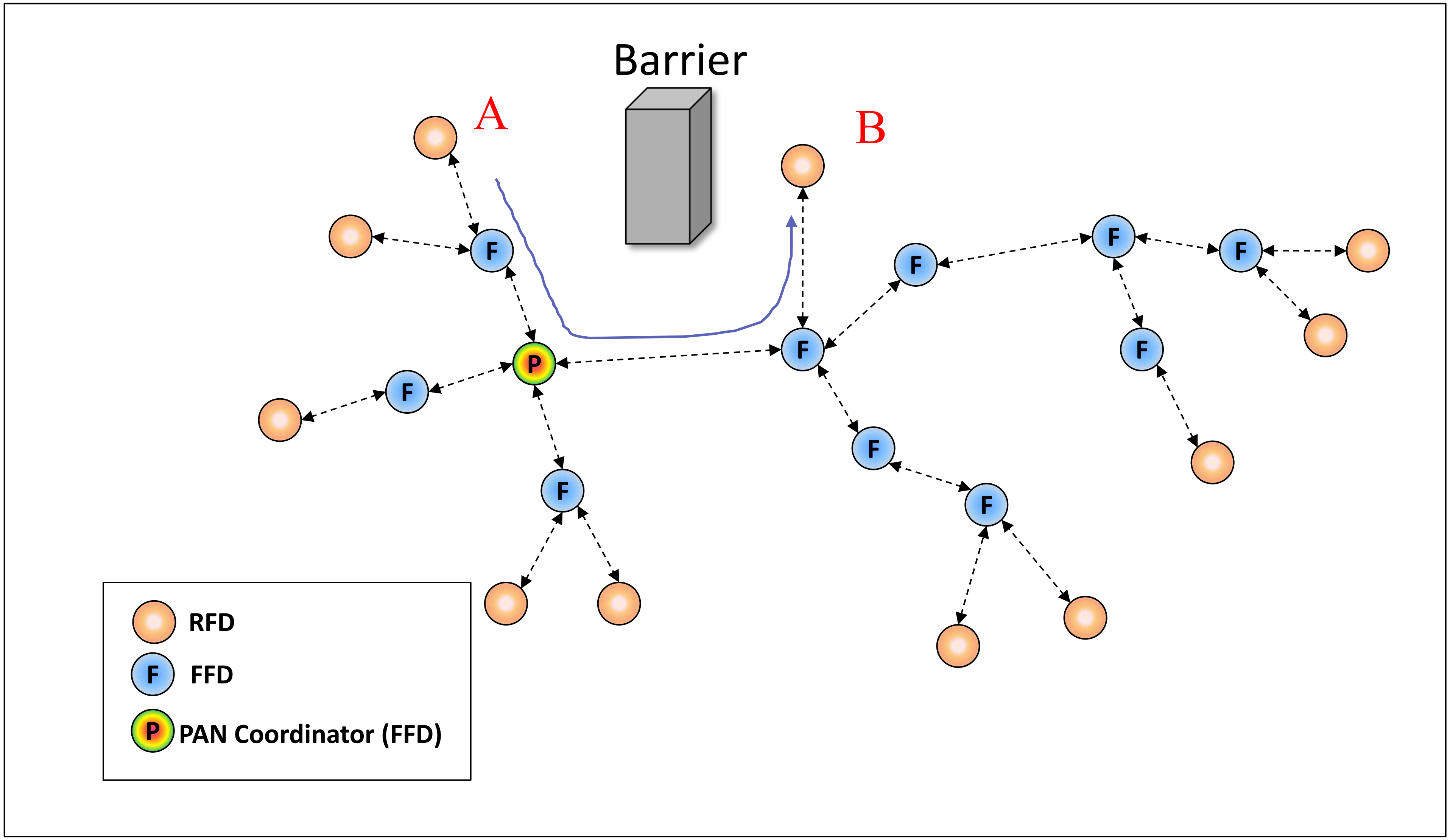
- Two Device types:
  - Full-Function Devices (FFDs)
    - They can perform all duties described in IEEE 802.5.4 standard
    - They can accept any role in the network
    - They can communicate with any other devices
    - Typically always-on with permanent power supply
  - Reduced-Function Devices (RFDs)
    - They have limited capabilities
    - They can only communicate with an FFD device
    - Typically on batteries



IEEE 802.15.4: Peer-to-Peer Topology – Mesh

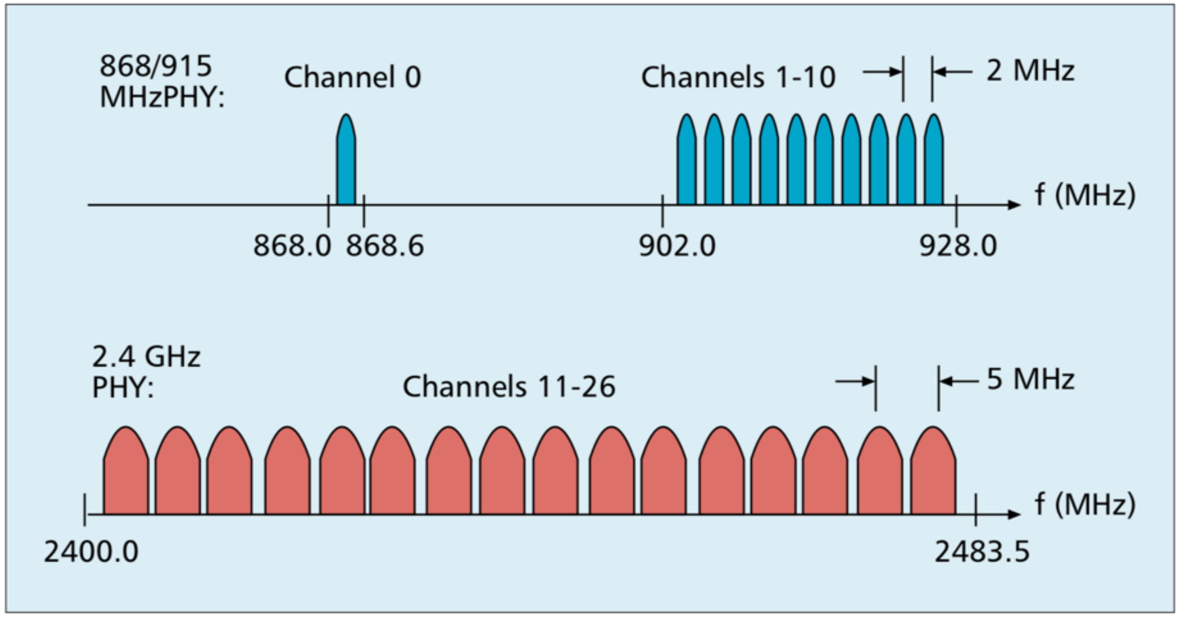


IEEE 802.15.4: Peer-to-Peer Topology – Tree



IEEE 802.15.4: PHY

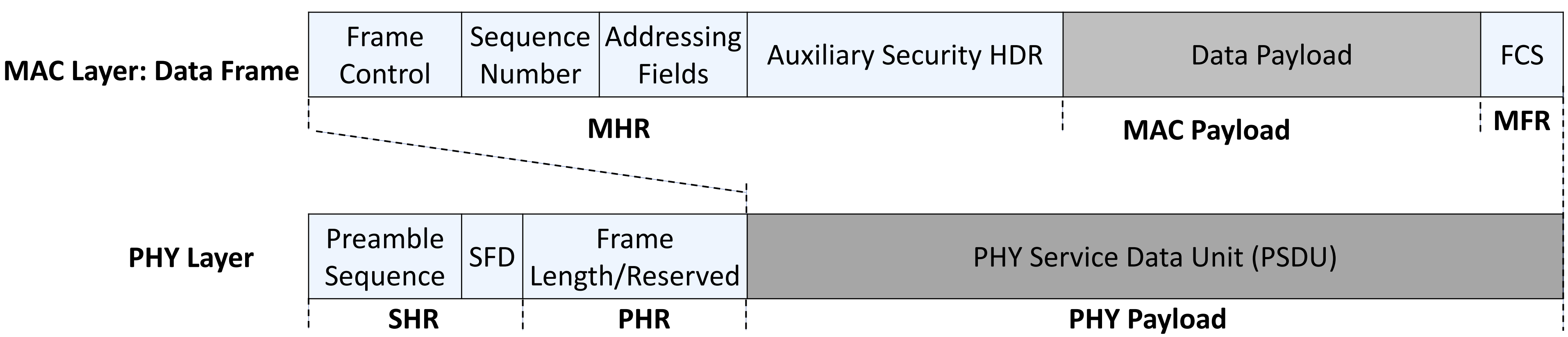
- Main features
  - Activation and deactivation of the radio transceiver
  - Energy detection within the current channel
  - Channel frequency selection



E. Callaway *et al*., "Home networking with IEEE 802.15.4: a developing standard for low-rate wireless personal area networks," in *IEEE Communications Magazine, 2002*

IEEE 802.15.4: Frame Format

- IEEE 802.15.4 is intended for devices with low data rates
  - Packets relatively small
  - Preamble (4 bytes) to achieve synchronization
  - SFD (1 byte): Start frame identifier
  - Frame length (7 bits): length of the PHY payload (up to 125 bytes)
  - MAC Frame types: data, beacon, ACK, command
- Addressing
  - unique 64-bit long address
  - short 16-bit address (only valid within a PAN)

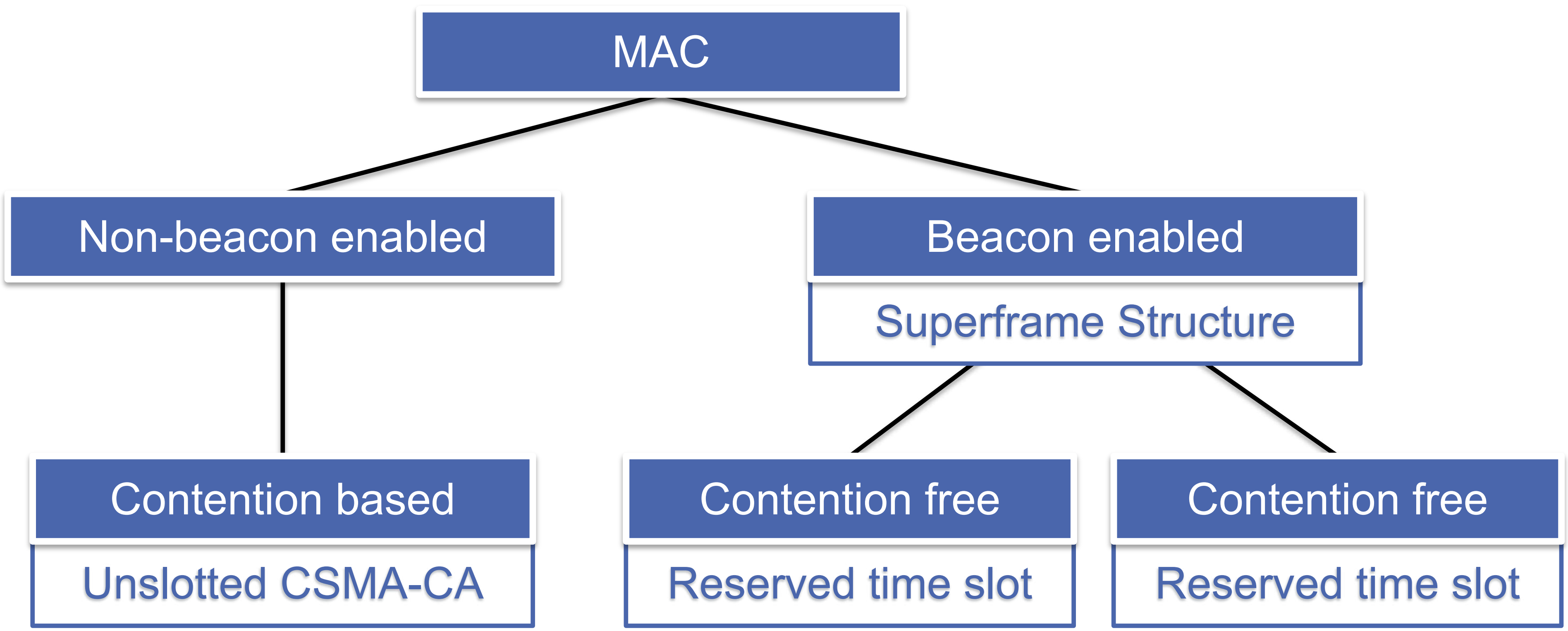


SHR: Synchronization header
PHR: PHY header

MHR: MAC header
MFR: MAC footer

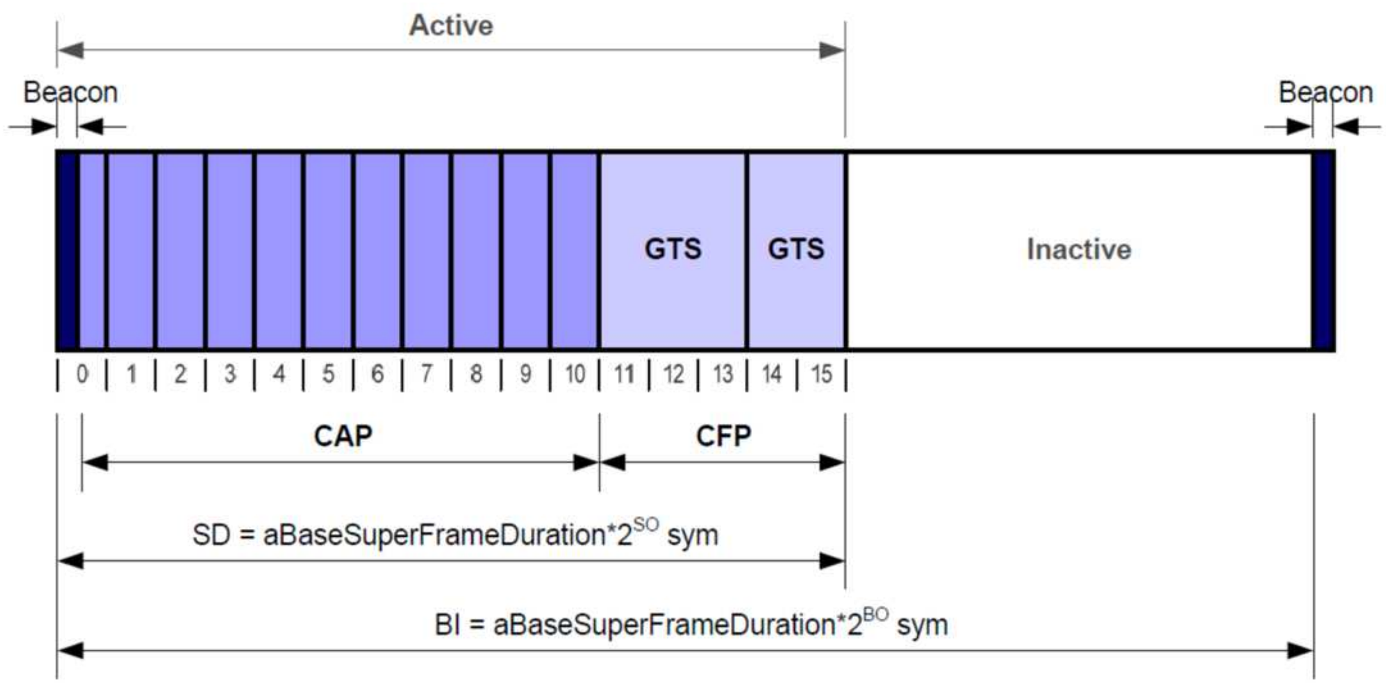
IEEE 802.15.4: MAC

- Two different channel access methods
  - Beacon enabled duty-cycled mode
  - Non-beacon enabled mode



IEEE 802.15.4: Beacon Enabled MAC

- Beacon enabled mode (Superframe)
  - CAP = Contention access period with simplified CSMA/CA
  - CFP = Contention-free period (GTS: Guaranteed Time Slot)
  - BI = Becaon Interval
  - SD = Superframe duration

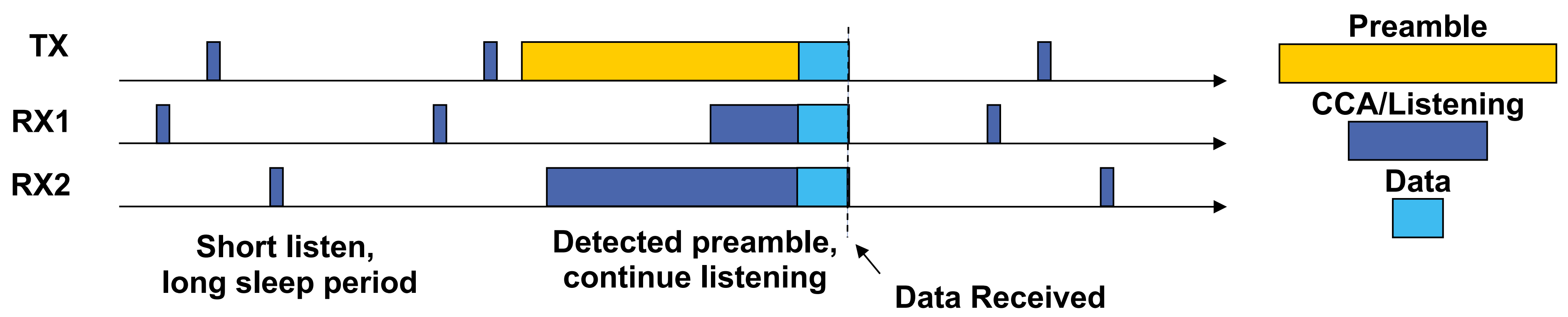


G. Anastasi, M. Conti and M. Di Francesco, "A Comprehensive Analysis of the MAC Unreliability Problem in IEEE 802.15.4 Wireless Sensor Networks," in *IEEE Transactions on Industrial Informatics, 2011*

IEEE 802.15.4: Non-beacon Enabled MAC

- Carrier Sense Multiple Access with Collision Avoidance
  - As in IEEE 802.11, with RTS/CTS/ACK
  - No need for synchronization, more flexible
- Many (many) alternatives
  - Idea: use CCA also at receiver to check for ongoing transmission

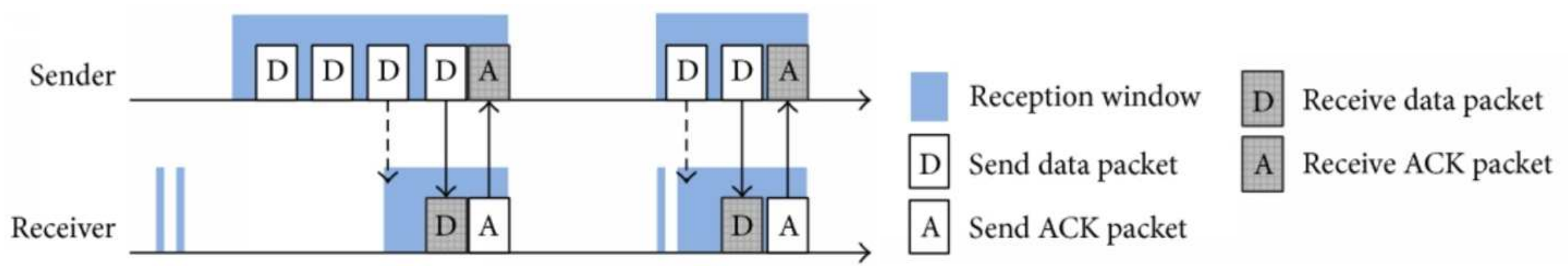
General Scheme



IEEE 802.15.4: Non-beacon Enabled MAC

- Carrier Sense Multiple Access with Collision Avoidance
  - As in IEEE 802.11, with RTS/CTS/ACK
  - No need for synchronization, more flexible
- Many (many) alternatives
  - Idea: use CCA also at receiver to check for ongoing transmission

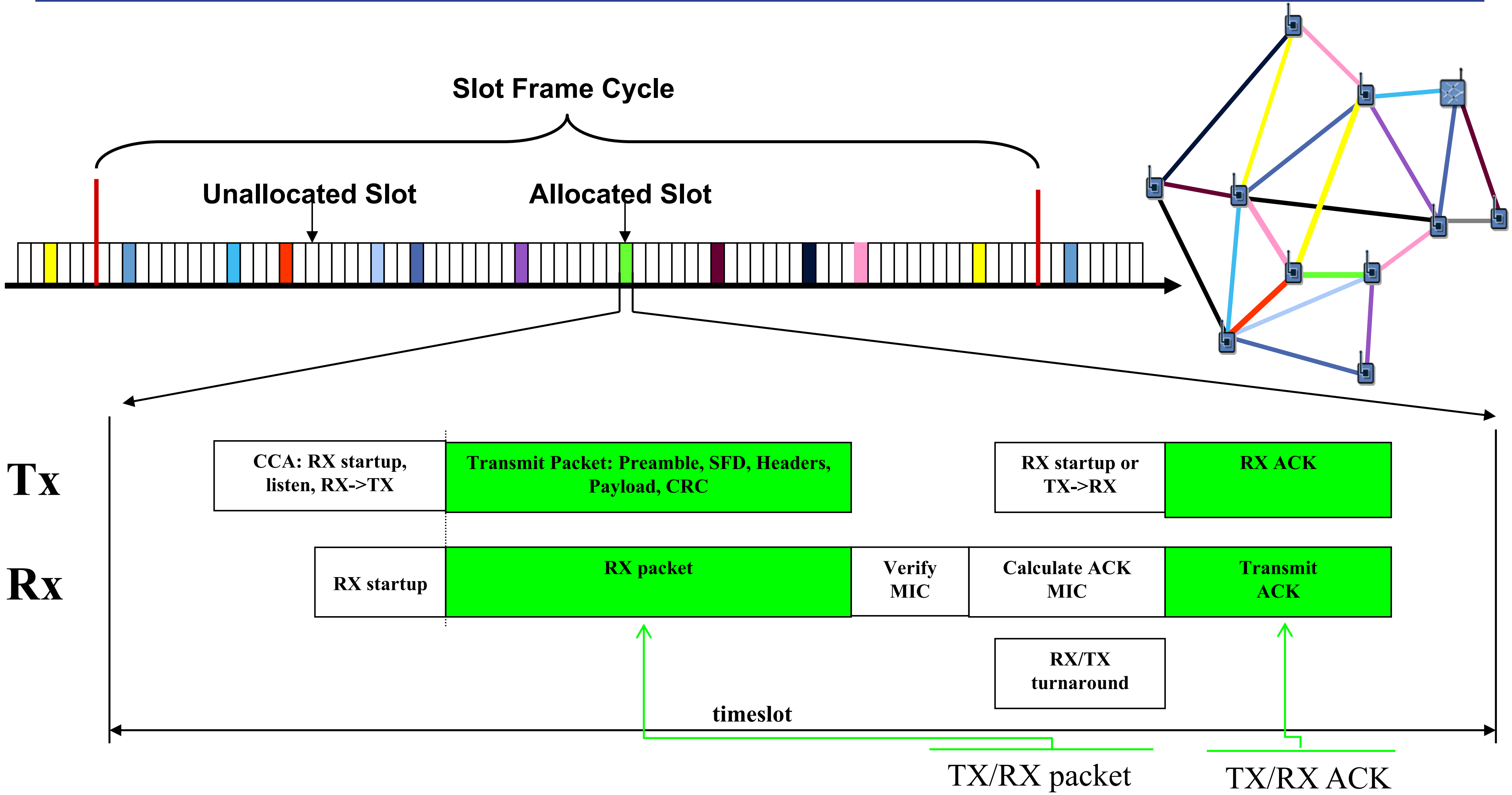
Specific Scheme:
A. Dunkels, "The ContikiMAC Radio Duty Cycling Protocol," Tec.Rep., 2011



IEEE 802.15.4e

- IEEE 802.15.4 limitations
  - Unbounded latency
  - No guaranteed bandwidth
  - Prone to interference (and other factors, like temperature)
- In 2012, amendment with Time Slotted Channel Hopping (TSCH)
  - Slotted access
  - Shared and dedicated slots
  - Multichannel communication
  - Frequency hopping

IEEE 802.15.4e: Slotted Access



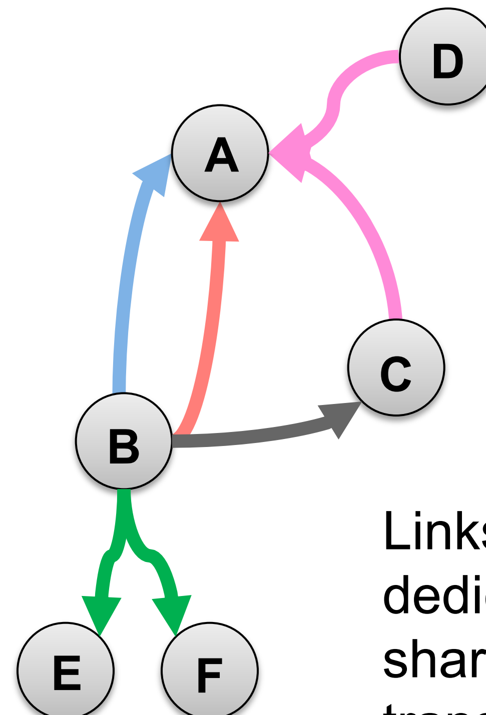
http://www.ieee802.org/15/pub/TG4e.html

IEEE 802.15.4e: Slotted Access

- Devices are configured with a slot frame and timeslots to communicate with each other
- Devices use timeslots to:
  - Schedule when they wakeup to transmit or listen
  - Keep time synchronized
  - Time the sequence of operations
    - Allow source and destination to set frequency channel
    - Listening for a packet
    - Sending a packet
    - Listening for an ACK
    - Generating an ACK
  - Synchronize channel hopping

IEEE 802.15.4e: Link Types

- Dedicated link
  - Assigned to one device for transmission and to one or more devices for reception
    - Dedicated broadcast link assigned to all devices for reception
- Shared link
  - Assigned to more than one device for transmission
    - ACK failures detect collisions
    - A back-off algorithm resolves collisions

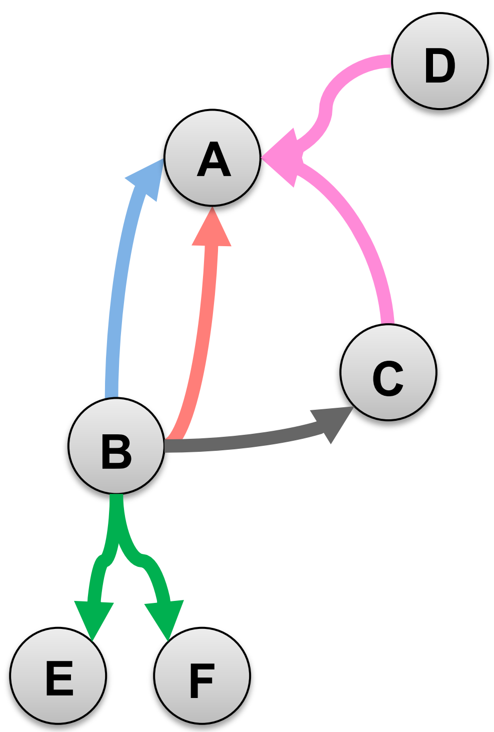


Links from B to A are dedicated; D and C share a link for transmitting to A

http://www.ieee802.org/15/pub/TG4e.html

IEEE 802.15.4e: Link Scheduling

- Link = (Timeslot, Channel offset)
  - The two links from B to A are dedicated
  - D and C share a link for transmitting to A
  - The shared link does not collide with dedicated links



Time →

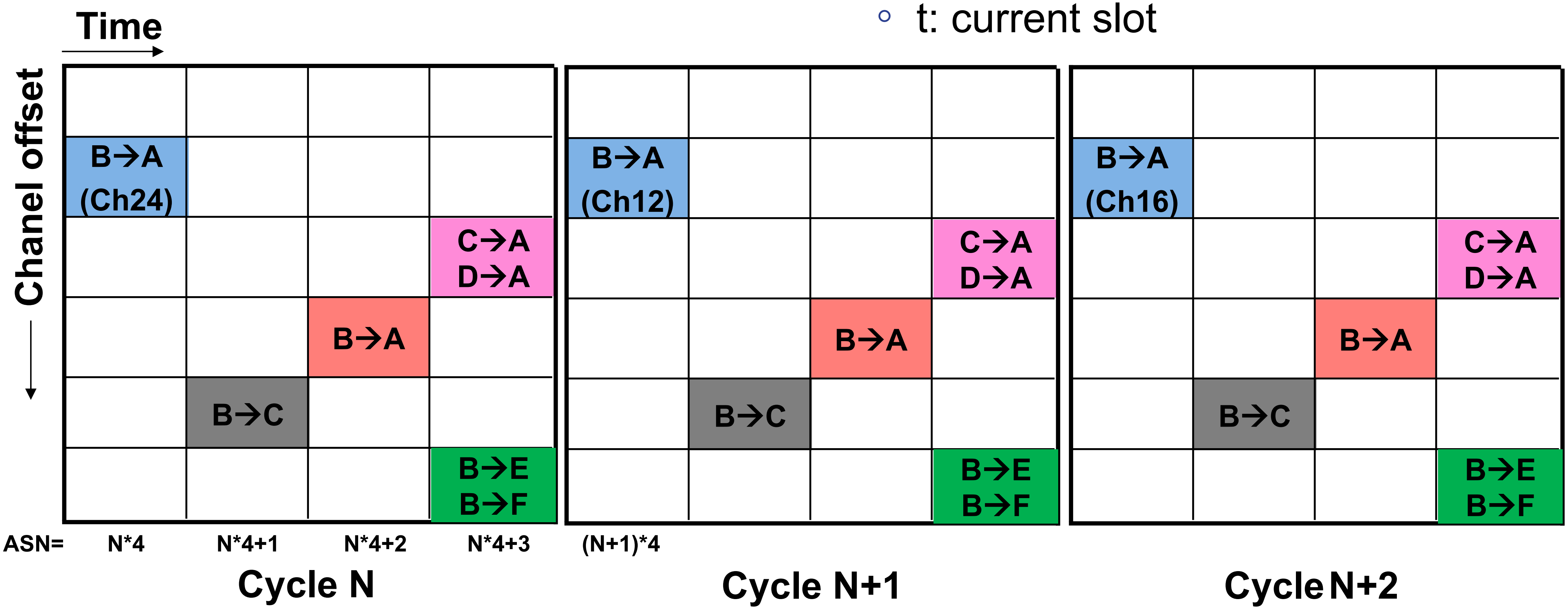
One Slot

Chan. offset ↓

B→A			
			C→A D→A
		B→A	
	B→C		
			B→E B→F

IEEE 802.15.4e: Channel Hopping

- $f = F\{(ASN + chOf) \bmod n_{ch}\}$
  - ASN: Absolute Slot Number
  - n_{ch}: number of used channels
- ASN = (N x S + t)
  - S: slotframe size
  - N: slotframe cycle
  - t: current slot



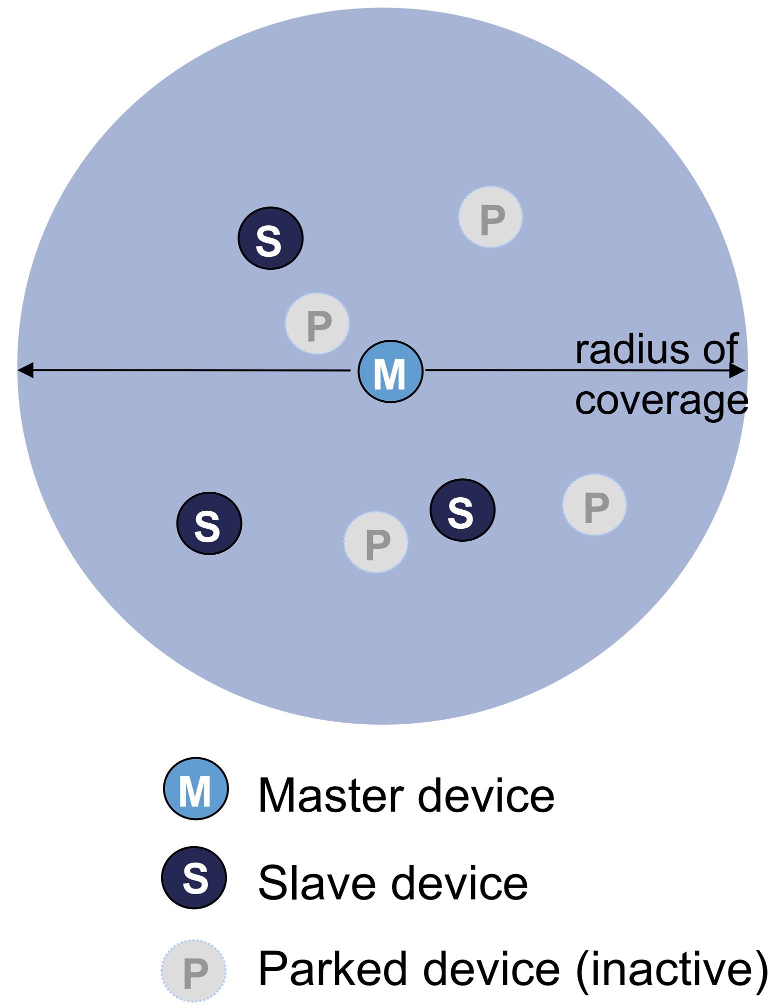
IEEE 802.15.4e: Schedule Formation

- Network formation
  - Difficult due to frequency mismatch
  - Based on beacons
    - Special frames containing
      - ASN and timeslot information
      - Synchronization and channel hopping information
    - Regularly broadcasted by the PAN coordinator
      - And by FFD already synchronized with the network
- How to build a schedule is not defined by the standard
  - Centralised and distributed policies are possible

http://info.iet.unipi.it/~anastasi/talks/2016-MST.pdf
https://ans.disi.unitn.it/inw2015/presentations/s6-3-vogli-scheduling.pdf

IEEE 802.15.1: Bluetooth

- Wireless technology for cable replacement
  - Low-power, cheap with reduced range
- RF radio layer
  - 2.4 GHz, 79 frequency channels, 1 MHz wide
  - Data rate: 1 Mbps (nominal)
  - Radio frequency hopping with a common sequence for all devices of a piconet
- Piconet: Bluetooth units sharing a single (frequency-hopping) channel
  - Single master with max 7 slaves
  - Master gives slaves clock and identifiers
  - All devices hop together in unison



IEEE 802.15.1: Communication and Limitations

- Basic communication
  - Slots are dedicated to a master or a specific slave
  - Master “polls” slaves and slave uses next slot to answer
  - Control packets in addition to data packets
    - NULL packet for ACK (if slave has no payload to transmit)
    - POLL packet when master has no payload to send
- Limitations
  - Connection-oriented: a link is maintained even if no data
  - Long and complex discovery process
  - Scalability: max 7 devices
  - Data throughput is the focus!

Bluetooth Low-Energy (BLE)

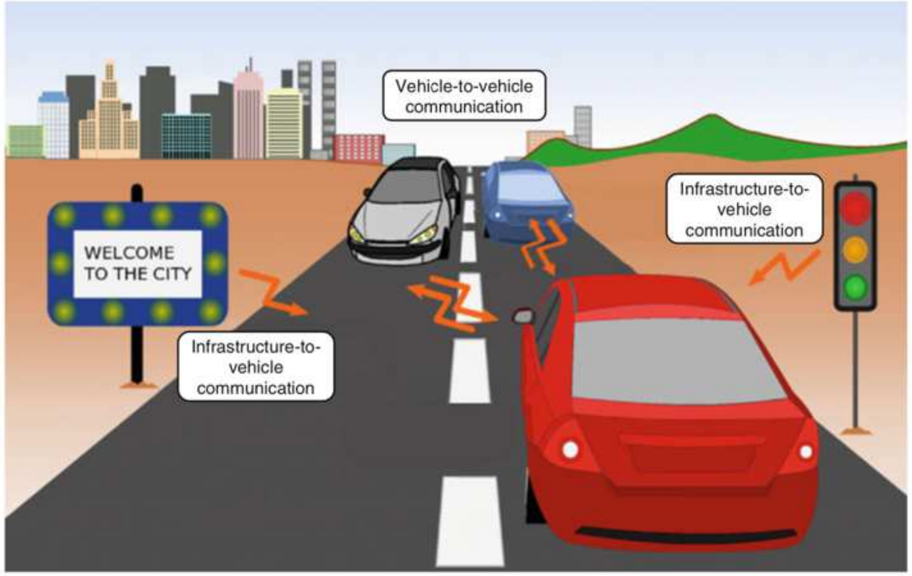
- New PHY layer
  - 2.4 GHz, 40 Channels with 2 MHz spacing with adaptive frequency hopping
  - 3 channels reserved for advertising and 37 channels for data
  - Not compatible with Bluetooth classic
- Four device roles
  - Peripheral device: connectable advertiser (slave)
  - Central device: connection initiator (master)
  - Broadcaster: non-connectable advertiser
  - Observer: scans for advertisements without connecting

Bluetooth Low-Energy (BLE)

- Data exchange
  - Connection-oriented
    - Connection request on advertising channels
    - Master informs slave about hopping sequence and wake-up times
    - Further transactions in the 37 data channels
    - Deep sleep allowed between transactions
  - Connection-less
    - Devices can broadcast advertisements outside a connection
    - Observer can scans for advertisements

Further Link-Layer Technologies

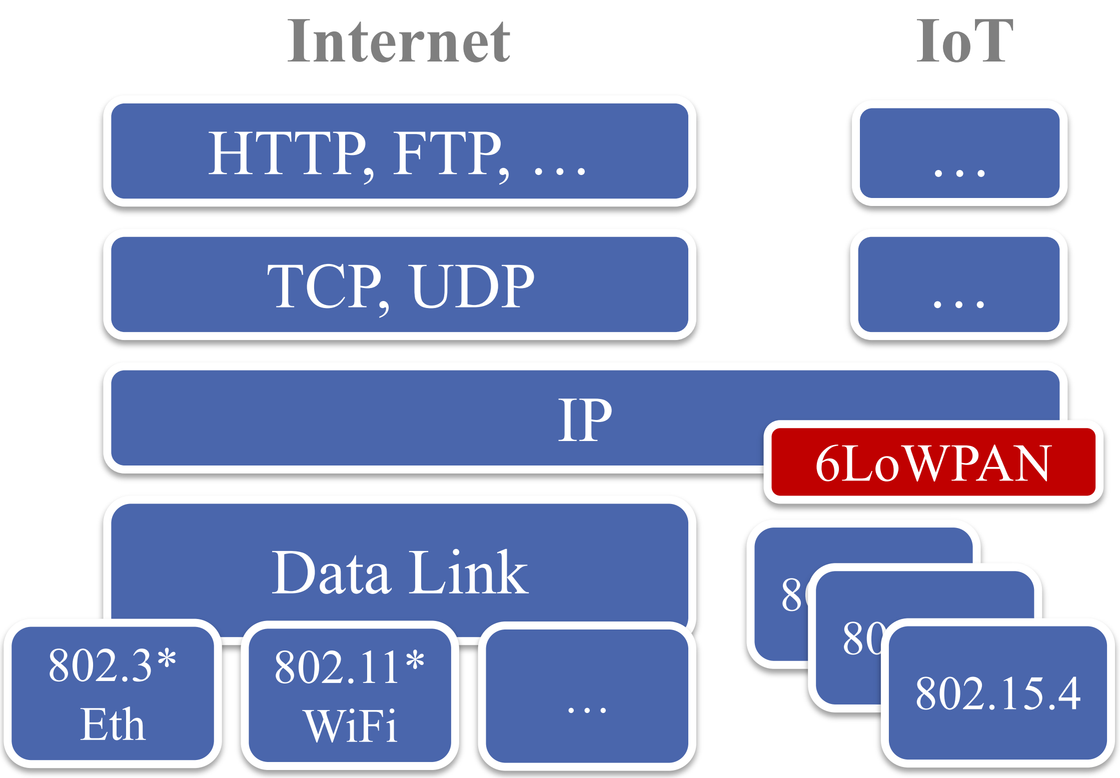
- IEEE 802.15.7 (VLC)
  - Short-range optical wireless communication using visible light spectrum
  - Problems: dimming support and flickering mitigation
  - Multiple light sources also possible
- Low-Power Wi-Fi
  - Optimised for low-power consumption
  - Standby mode with timings to prevent disassociation
- Long-Range Technologies
  - Allows to trade energy for throughput, reliability and range
  - Small amounts of data
  - Examples: LoRa ("long-range low-power WiFi") and NB-IoT ("cellular IoT")



http://tinyurl.com/hpf3fhv

6LoWPAN

- IPv6 over Low-power Wireless Personal Area Networks
  - An "adaptation layer"
  - Cover the mismatch between
    - IPv6
    - Link-layer technologies designed for low-power wireless embedded devices



6LoWPAN

- Goals
  - Adapt IPv6 to constraints of IoT
  - Transparent Internet integration
  - Scalability – consider large local area networks
  - End-to-end connectivity
  - Provide Standards
  - Close cooperation with 802.15.4 working group
- IETF 6lowpan working group
  - RFC4919: “"IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals”
  - RFC4944: “Transmission of IPv6 Packets over IEEE 802.15.4 Networks”

Features

- Provide adaptation layer
- Header compression
- Fragmentation: overcome payload restrictions for 1280 MTU
- Stateless auto configuration (from IPv6)
  - Use hardware MAC address as host address
  - Router solicitation multicasts
- Tight integration with underlying network (i.e. 802.15.4)
- Routing is not covered (complementing standardization efforts)

Header Compression

- Assumptions
  - Within the LoWPAN: flat address space
  - Unique MAC layer addresses (802.15.4: 16 bit or 64 bit)
- Approach
  - Omit network prefix: assume known within the WPAN
  - IID (src/destination)
    - Omit if link-local (i.e., within WPAN): addresses are part of 802.15.4 header
    - Compress for multihop
- Stacking of headers
  - Include only what is needed (e.g., for fragmentation)

Header Dispatch

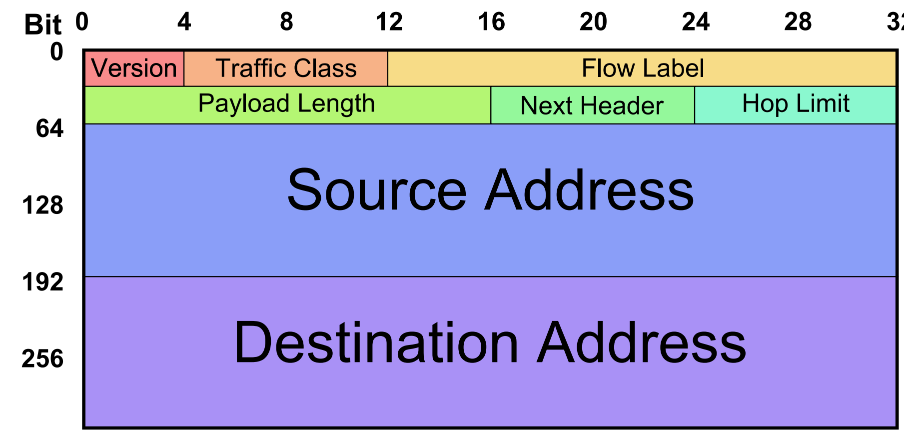
- Basic structure

```
+---------------+-------------+---------+
| IPv6 Dispatch | IPv6 Header | Payload |
+---------------+-------------+---------+
```

- Header dispatch: Identify what kind of header is following
  - 00: Not LoWPAN
  - 01: LoWPAN IPv6 addressing header
  - 10: LoWPAN mesh header
  - 11: LoWPAN fragmentation header

LoWPAN IPv6 addressing header

- Dispatch:
  - 000001: uncompressed IPv6 address (40 bytes)
  - 000010: HC1 (Fully compressed: 1 byte)
    - Source: MAC header
    - Destination: MAC header
    - Traffic Class + Flow Label: 0
    - Next Header: UDP/TCP/ICMP
- Hop limit is never compressed




```
                    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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0 1| Dispatch  |  type-specific header
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

HC1 compressed header

- Flags HC1 encoding (bit 0-7)
  - 0: Source prefix omitted (i.e., use link layer)
  - 1: Source interface ID omitted (i.e., use link layer)
  - 2: Destination prefix omitted (i.e., use link layer)
  - 3: Destination interface ID omitted (i.e., use link layer)
  - 4: Traffic and Flow Label omitted (assume 0)
  - 5-6: Next header (00 uncompressed, 01 UDP, 11 TCP, 11 ICMP)
  - 7: HC2 encoding follows (e.g., compressed UDP header)

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| HC1 encoding  |     Non-Compressed fields follow...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Compressed UDP header

- HC_UDP encoding (= HC2 for UDP)
  - 0: source port compressed to 4 bits follows ActualPort=61616+ShortPort (i.e., 61616-61631)
  - 1: destination port compressed to 4 bits follows (cf., above)
  - 2: length computed from link layer packet length and header lengths

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|HC_UDP encoding|     Fields carried in-line follow...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Fragmentation

- Approach: all fragments carry same "tag" (sequence number)
- Fragments may arrive out-of-order
- Datagram_size: size of complete datagram (11 bits)
- First fragment:

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1 1 0 0 0|    datagram_size    |         datagram_tag          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- Subsequent fragments

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1 1 1 0 0|    datagram_size    |         datagram_tag          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|datagram_offset|
+-+-+-+-+-+-+-+-+
```

6LoWPAN Neighbor Discovery

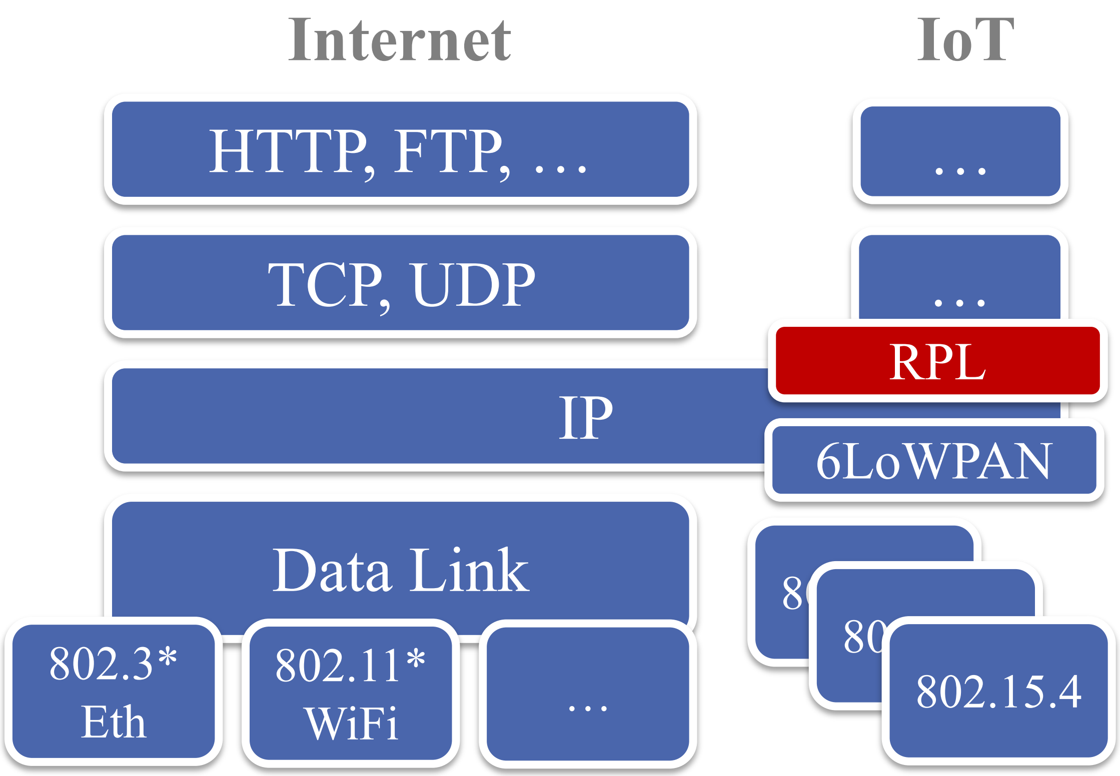
- Standard IPv6 neighbor discovery not suitable
  - Assumes single link for subnet (e.g., Ethernet cable)
  - Assumes nodes are always-on
  - Heavy use of multicast (would require flooding)
- 6LoWPAN ND
  - Standardization in progress
  - Minimize control traffic
  - Re-use MAC layer capabilities
  - Optimize host-router interface

6LoWPAN Neighbor Discovery (2)

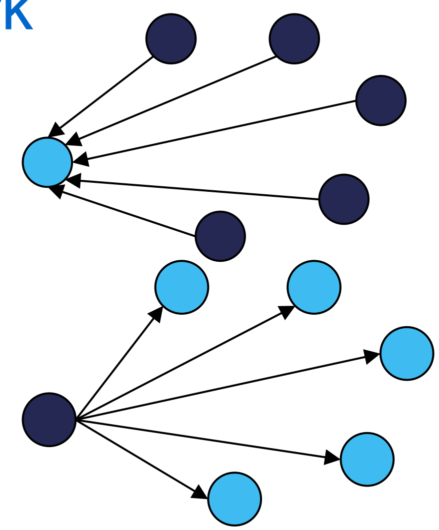
- Address Auto-Configuration
  - Prefix disseminated in network
  - Interface ID: from 64-bit 802.15.4 address
- Nodes exchange information with neighboring nodes
  - Neighborhood tables
  - Duplicate address detection
  - Address resolution

Routing for 6LoWPAN

- IETF ROLL working group
  - "Routing Over Low power and Lossy networks" (LLN)
  - Separate from (but cooperating with) 6LoWPAN
  - One possible routing implementation on top of 6LoWPAN
- RFC 6550: "RPL: IPv6 Routing Protocol for Low power and Lossy Networks"



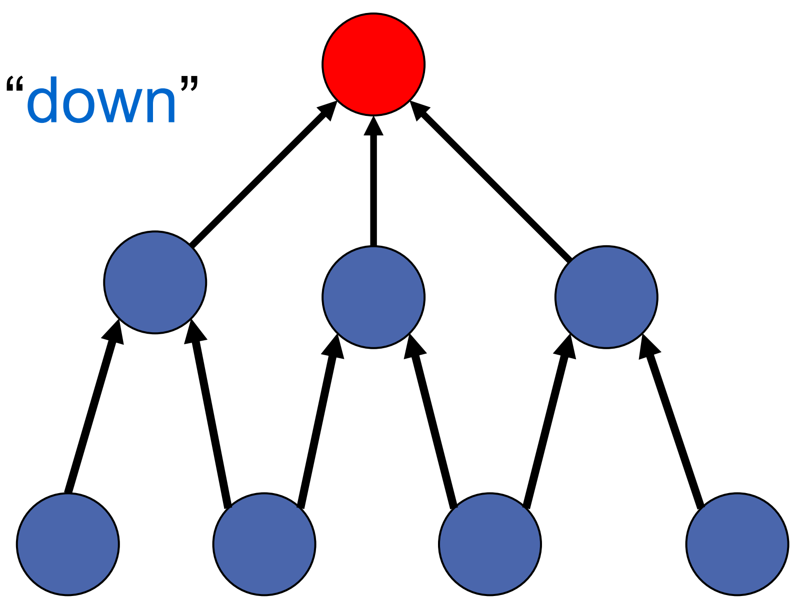
RPL – Assumptions



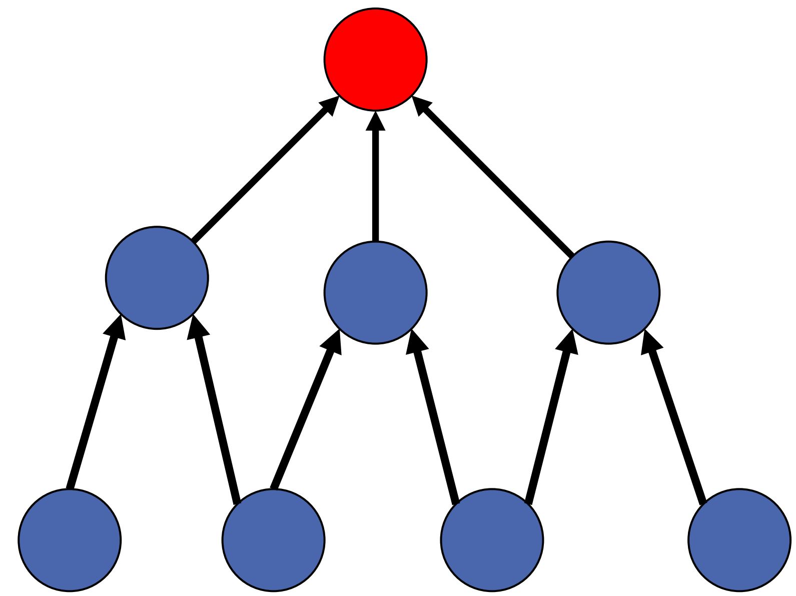
- Dominant traffic pattern differs from traditional network
  - Many-to-one (Convergecast)
  - One-to-many (Network broadcast, dissemination)
- Most traffic flows through few nodes
  - Sinks
  - Gateways to different physical networks (e.g., 802.15.4 ←→ Ethernet)
- Traffic out of the network into the internet important
- Experiences from research
- Requirements gathered from industry

Approach

- Build DAG(s) rooted at “important” nodes
  - Usually border routers (gateways)
- Consider 2 directions:
  - Down: from root in direction to the leafs
  - Up: in direction to the root
- Point to point connections by “up” then “down”
- DAG more robust than tree
  - Multiple possible parents



Terminology



- DAG root
  - Node with no outgoing edges
  - 1 or more per DAG
  - All paths terminate at a DAG root
- DODAG “Destination Oriented DAG”
  - A DAG with exactly 1 root
  - Can offer routing to set of destination prefixes
- Rank of a node (in a DODAG)
  - The position with respect to a root
  - Topological distance or more advanced metrics
- DODAG Parent
  - Immediate successor on path to root
  - Lower Rank
- DODAG Sibling
  - Neighboring node with same rank
  - May or may not share common parents

Terminology (2)

- Goal
  - Is a node or a set of nodes satisfying an application objective
- Goal nodes within the network become roots
- Nodes with access to external goal nodes become roots
- A DODAG is
  - Grounded: it provides access to its goal
  - Floating: otherwise (e.g., during repair phases)

Objective Function

- Metrics
  - Compare possibilities (ranking)
- Constraints
  - Reduce number of possibilities
- Node metrics and constraints & Link metrics and constraints
- Metrics can be used as constraint (maximum/minimum requirements)
- Objective Function (OF): specifies goal based on metrics and constraints
  - E.g.: “Avoid battery operated links and compute the path that optimizes reliability”
  - Is well known in the network
  - Multiple OFs may be present and used for constructing multiple DODAGs

Node Metrics and Constraints

- Node state and attributes
  - Overloaded
  - Traffic aggregator: node is able to reduce traffic by aggregation
  - Extensible (but currently no other properties defined)
- Node Energy Object
  - Support for different levels of detail
  - Simplest: Powered vs. scavenger vs. battery
  - Mid: Energetic happiness
    - Scavenger: H=PowerGenerated/PowerConsumed in percent
    - Battery: H=EstimatedLifetime/DesiredMinimumLifetime
  - Extensible: definition of more possibilities underway
- Hop Count
  - Number of hops to root

Link Metrics and Constraints

- Throughput (bytes/second)
- Latency (milliseconds)
- Link reliability
  - Link Quality Level: unknown or 1-3 (1=best)
    - As constraint: require minimum value
    - As metric: aggregate over DAG; either sum or minimum
  - ETX (cf. chapter Routing): as (additive) metric or constraint
- Link color
  - Static, administrative attribute (e.g., bit 1: encryption)
  - 10 bits →10 colors; links can have multiple colors
  - Metric: count number of links of each color, i.e. 10 counters
  - Constraint: exclude/include links with certain color

RPL Instance

- Routing topology for a given objective function
- 1 or more DODAG roots
- Node disjoint DODAG(s) (1 per root)
- Each root has a DODAGID
  - Multiple roots can have the same ID
  - Different DODAGs but, e.g. offer routing to same set of destination prefixes

```
+----------------------------------------------------------------+
|  +--------------+                                              |
|  |              |                                              |
|  |     (R1)     |              (R2)                  (Rn)      |
|  |     /  \     |              /| \                  / |  \    |
|  |    /    \    |             / |  \                /  |   \   |
|  |  (A)    (B)  |           (C) |  (D)     ...   (F) (G)  (H)  |
|  |  /|\     |\  |           /   |   |\           |   |    |    |
|  |  : : :   : : |          :   (E)  : :          :   :    :    |
|  |              |               / \                            |
|  +--------------+              :   :                           |
|       DODAG                                                     |
+----------------------------------------------------------------+
```

DODAG Iteration

- A DODAG root can request a reconstruction of a DODAG by increasing the DODAGSequenceNumber

```
+----------------+                +----------------+
|                |                |                |
|      (R1)      |                |      (R1)      |
|      /  \      |                |      /         |
|     /    \     |                |     /          |
|   (A)    (B)   |         \      |   (A)          |
|   /|\     |\   |    ------\     |   /|\          |
|  : : (C)  : :  |           \    |  : : (C)       |
|                |           /    |        \       |
|                |    ------/     |         \      |
|                |         /      |         (B)    |
|                |                |          |\    |
|                |                |          : :   |
|                |                |                |
+----------------+                +----------------+
        Sequence N                       Sequence N+1
```

DAG Construction

- Similar to tree based routing
- Uses link-local broadcasts
- Root starts broadcasts: DODAGID + metric values
- Nodes join DAG when suitable parent is found
  - Based on objective function
- Nodes rebroadcast DAG information with local metric values
- Rank is calculated based on position in DAG
- Information is periodically rebroadcasted to detect inconsistencies → Trickle Timer

The Trickle Algorithm

- Density-aware local broadcast
- Used for dissemination
- Goals
  - Quick resolving of inconsistency
  - Energy saving in stable states
  - Energy saving in dense topologies
- Originally used for code dissemination
  - Disseminate meta data (incl. version numbers) for code updates
- Share information with neighbors
  - Detect + repair loops
  - React to topology changes (e.g., node failures, link failures)

Trickle Algorithm

- Information is broadcasted locally “from time to time”
- Parameters:
  - Minimum Interval (e.g., 100 ms): Imin
    - Should be based on link parameters (e.g., 4* worst-case latency)
  - Maximum Interval: Imax
    - In doublings of the minimum interval
    - E.g.: 16 → 100ms * 2^16 = 6553.5 s ~ 109 minutes
  - Redundancy constant: k
- Variables
  - I: current interval
  - t: time within current interval
  - c: counter

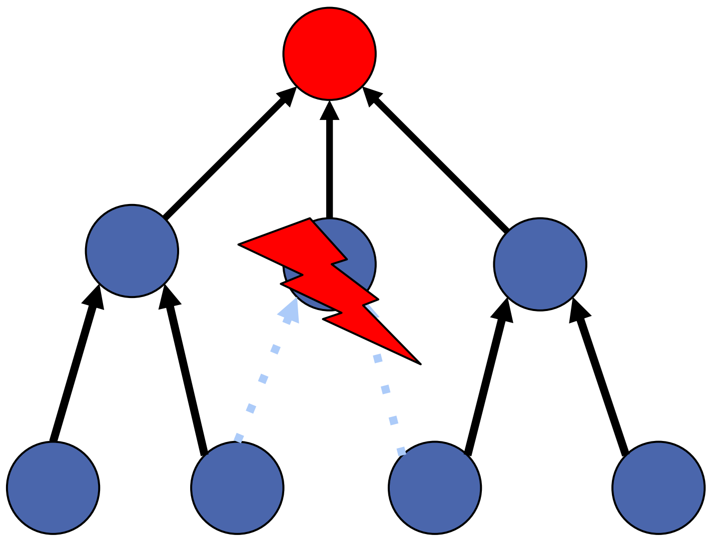
Trickle Algorithm Rules

1. Start of interval
   - c := 0
   - t := random(I/2, I)
2. On receive transmission
   - If consistent => c++
   - If inconsistent =>
     - resetTimer => If I != Imin => I := Imin; goto 1
3. At time t: if c < k => broadcast
4. At end of interval: I := max(I*2, Imax)
5. On external event (i.e. change of value): resetTimer

RPL Traffic Flows

- MP2P: Multipoint-to-Point Traffic
  - Similar to Convergecast: goals are the possible destinations
  - Traffic routed to root
  - But root can be intermediate node on path to final destination
- P2MP: Point-to-Multipoint Traffic (optional feature)
  - Routing towards destination prefixes, away from root
- P2P: Point-to-Point Traffic (optional feature)
  - One hop (neighbors)
  - Multi-hop over DAG
  - Possibility (but no specification) for additional routing mechanisms

Upward Routes



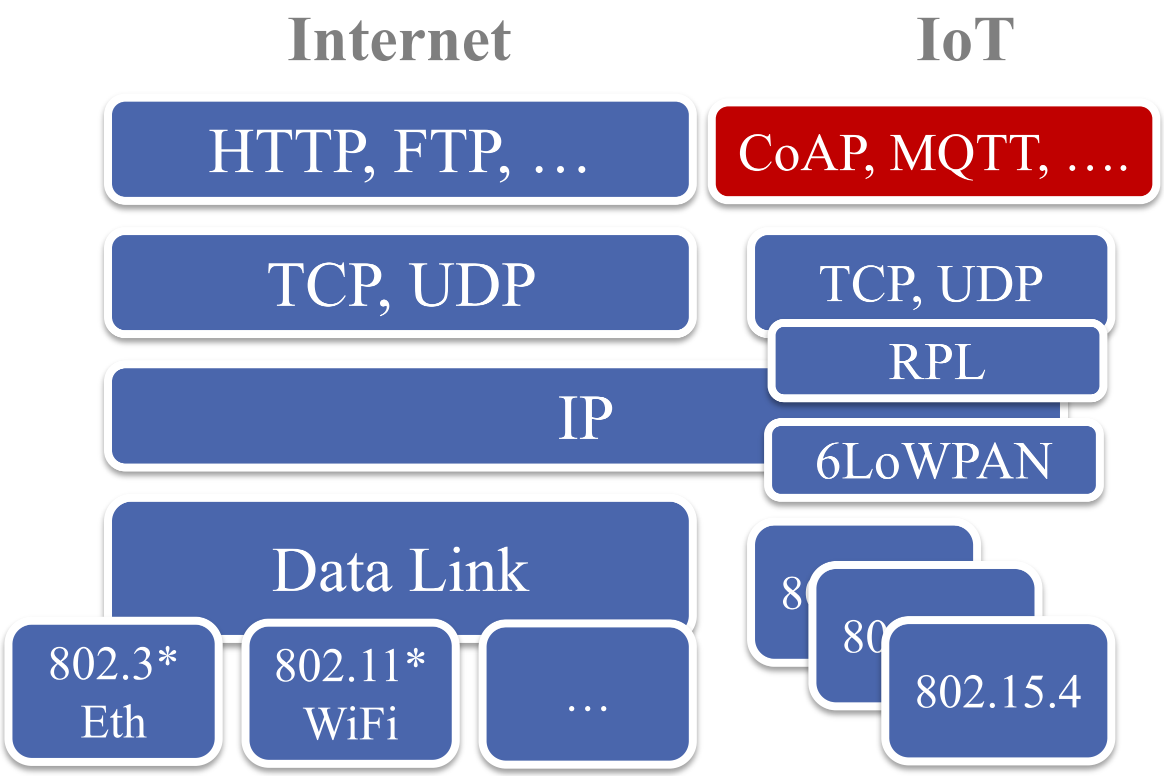
- By using DODAG
  - Send to parent
- Nodes store preferred parent
- Nodes can store one or more parents
  - Increase path diversity
  - Load balancing
  - By-pass faulty nodes
- Links to siblings can be used for increased diversity
- Messages are forwarded upwards until…
  - Destination reached or …
  - A node with a route to destination is reached, e.g.,
    - Destination
    - Gateway node
    - Neighbor of destination
    - Intermediate node with route to destination (start downward routes)

Downward Routes

- Optional feature if P2MP or P2P traffic is required
- Nodes can send Destination Advertisement Objects (DAO)
  - Contain destination prefix
  - Sent to parent
  - Parent adds own address to routing stack and forwards
  - May store information for later use
  - “no-DAO” clears previous established downward route
- Source routing is used for downward routes

Web of Things

- Web of Things:
  - Approaches, software, architectures and programming patterns that allow smart objects to be part of the World Wide Web
  - Web Services for Resource-constrained devices



MQTT: Message Queuing Telemetry Transport

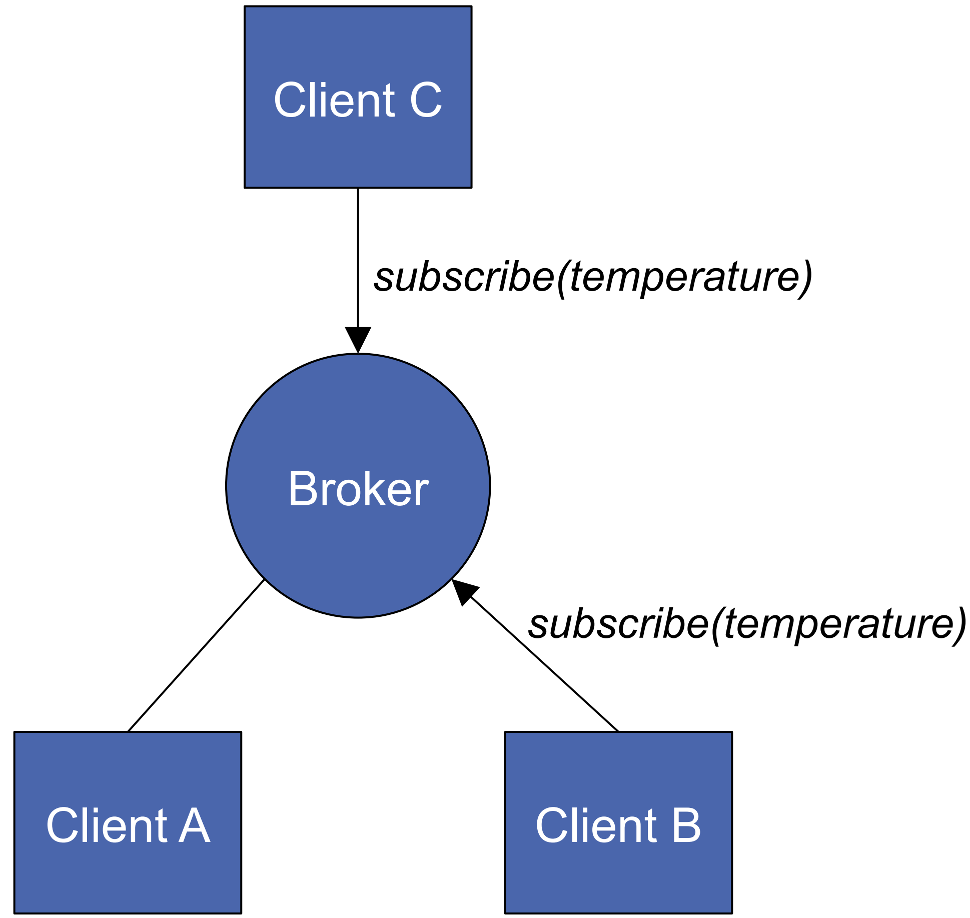
- Publish/Subscribe messaging protocol
  - Explicitly designed for lightweight machine-to-machine (M2M) communications
  - ISO standard
  - Not a new IoT protocol
    - First version in 1999
    - Version ~~3.1.1~~ 5.0
    - Not specifically designed for IoT

MQTT: Protocol

- Client/Server model
- Every sensor is a client and connects to a server (known as broker)
  - Connection using TCP
- Every message is published to an address (known as topic)
  - Clients may subscribe to multiple topics
  - Every client subscribed to a topic receives every message published to the topic
- Allows one-to-one, one-to-many, many-to-one communication

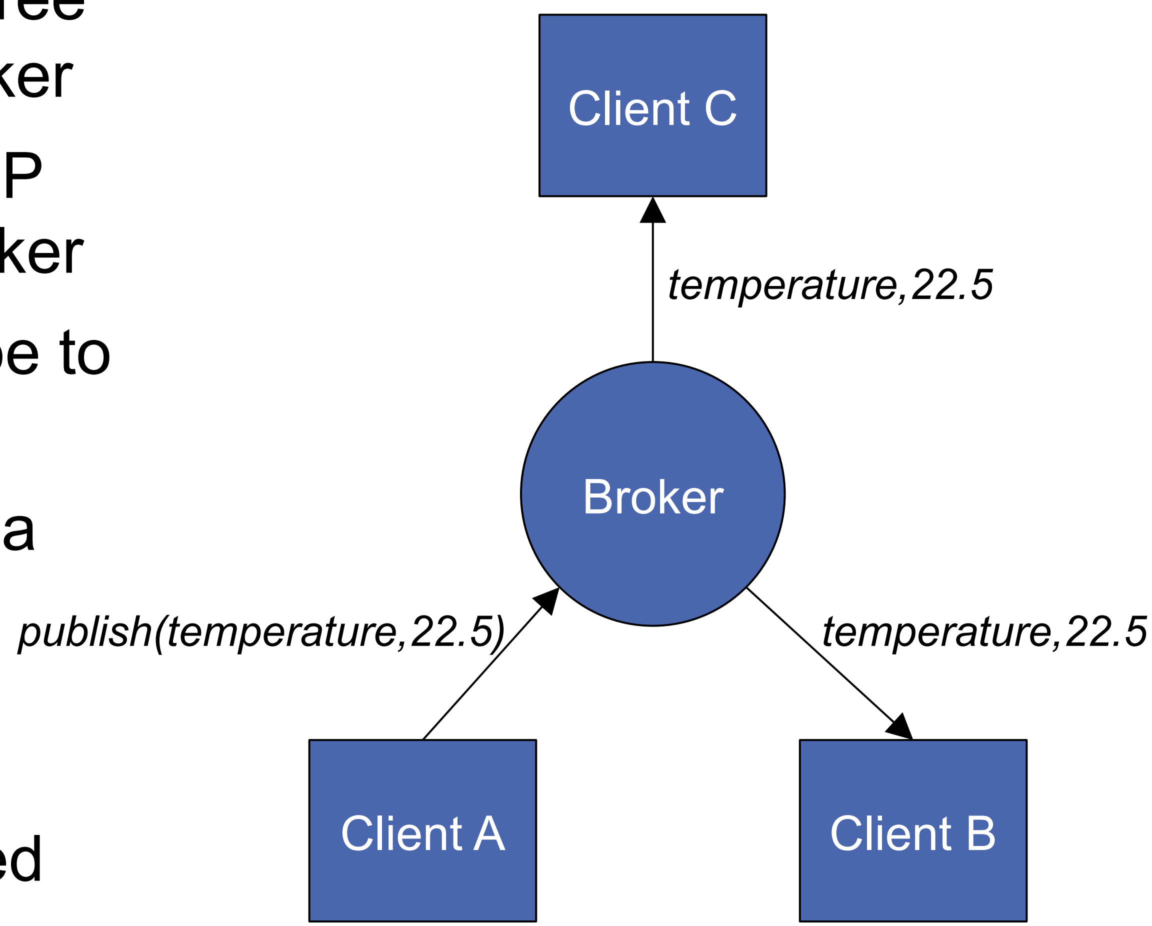
MQTT: Example

- A simple network with three clients and a central broker
- All three clients open TCP connections with the broker
- Clients B and C subscribe to the topic `temperature`



MQTT: Example

- A simple network with three clients and a central broker
- All three clients open TCP connections with the broker
- Clients B and C subscribe to the topic `temperature`
- Later, client A publishes a value of 22.5 for topic temperature
- The broker forwards the message to all subscribed clients

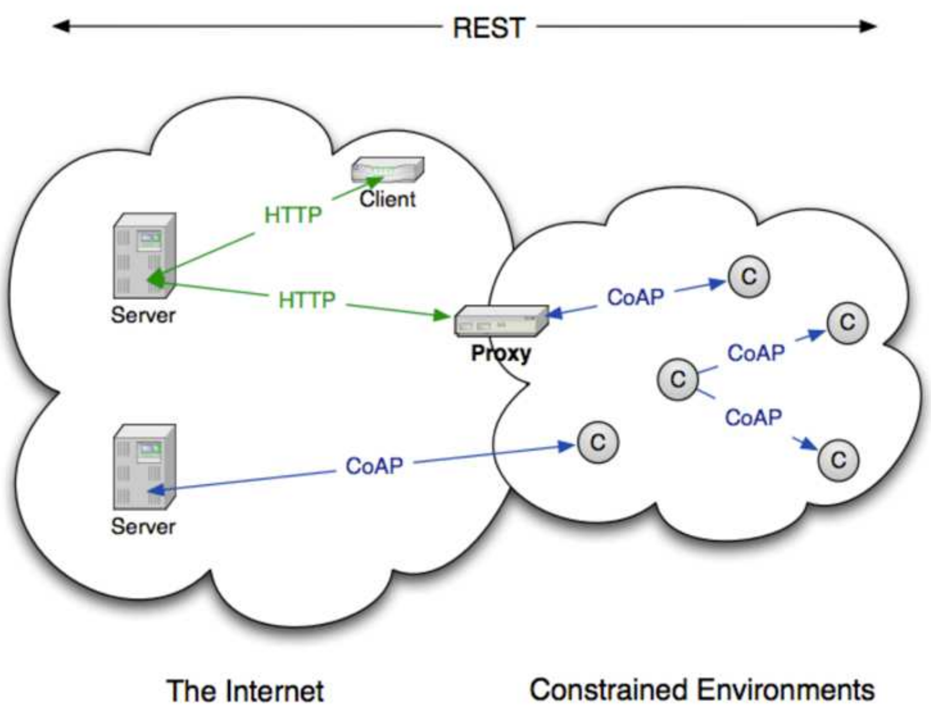


MQTT: Topics and QoS

- Topics are hierarchical
  - Like a file system (e.g., `kitchen/oven/temperature`)
  - Wildcards are allowed when registering a subscription (but not when publishing)
    - Example: `kitchen/+/temperature`
- Application-level QoS
  - Abilities of TCP exploited in different ways
  - Three quality of service levels supported
    - Fire and forget
    - Delivered at least once
    - Delivered exactly once
- Limitations: TCP!

CoAP: Constrained Application Protocol

- CoRE: Constrained ReSTful Environments
  - CoAP used withing the LoWPAN
  - Proxy maps CoAP to HTTP and viceversa



https://www.slideshare.net/zdshelby/coap-tutorial

CoAP: Constrained Application Protocol

- CoAP is similar to HTTP…
  - Request/response communication model
  - GET, POST, PUT, DELETE semantics (easy to map to HTTP)
  - URI support
- … but it is optimised for low-power and lossy networks
  - UDP binding with reliability and multicast support
  - Support for asynchronous message exchange
  - Low header overhead and parsing complexity
    - Small, simple 4 byte header

CoAP: Features

- Built on top of UDP
  - Reliable unicast
  - Best-effort unicast
  - Support for asynchronous message exchanges
- Message types
  - Confirmable (CON), for reliable transmission
  - Non-confirmable (NON), for unreliable transmission
  - Acknowledgement (ACK)
  - Reset (RST)
- Ongoing communication identified by message IDs and tokens