Lecture 09
ZigBee

CS397/497 – Wireless Protocols for IoT

Branden Ghena – Spring 2022

Materials in collaboration
with Pat Pannuto (UCSD)

Updates

- Proposals
  - Are excellent and I’m super excited
  - I’ll get everyone detailed feedback throughout the week
    - Already sent an email if I was concerned

- Next lab
  - Will be released later this week

Today’s Goals

- Introduce ZigBee as another 802.15.4 implementation

- Discuss ZigBee application layer and interoperability

Outline

- **ZigBee overview**
- ZigBee PHY and MAC
- ZigBee application layer
- Interoperability

ZigBee goals

- Enable automatic communication between devices
  - Low complexity
  - Low power
  - Focus on home automation and industrial control/monitoring

- From our perspective
  - 802.15.4 PHY and MAC
  - Plus well-defined Server/Client interactions
    - Similar to BLE (actually, BLE is similar to ZigBee)
  - Designed for higher-power devices than Thread or BLE
    - Although still relatively low power

ZigBee history

- Intertwined with the creation of 802.15.4
  - Both are founded around the same time
  - ZigBee Alliance involved in the original 802.15.4 specification
    - Recently renamed: Connectivity Standards Alliance (CSA)
  - Original plan: 802.11/WiFi <-> 802.15.4/ZigBee

- Original specification 2004 (following 802.15.4 in 2003)
  - Updated 2006, 2007, 2015, (2017?)
  - 2015 version is also known as ZigBee Pro
  - We'll focus on 2015, but look at previous stuff too
    - Application layer stuff hasn't changed considerably

ZigBee resources

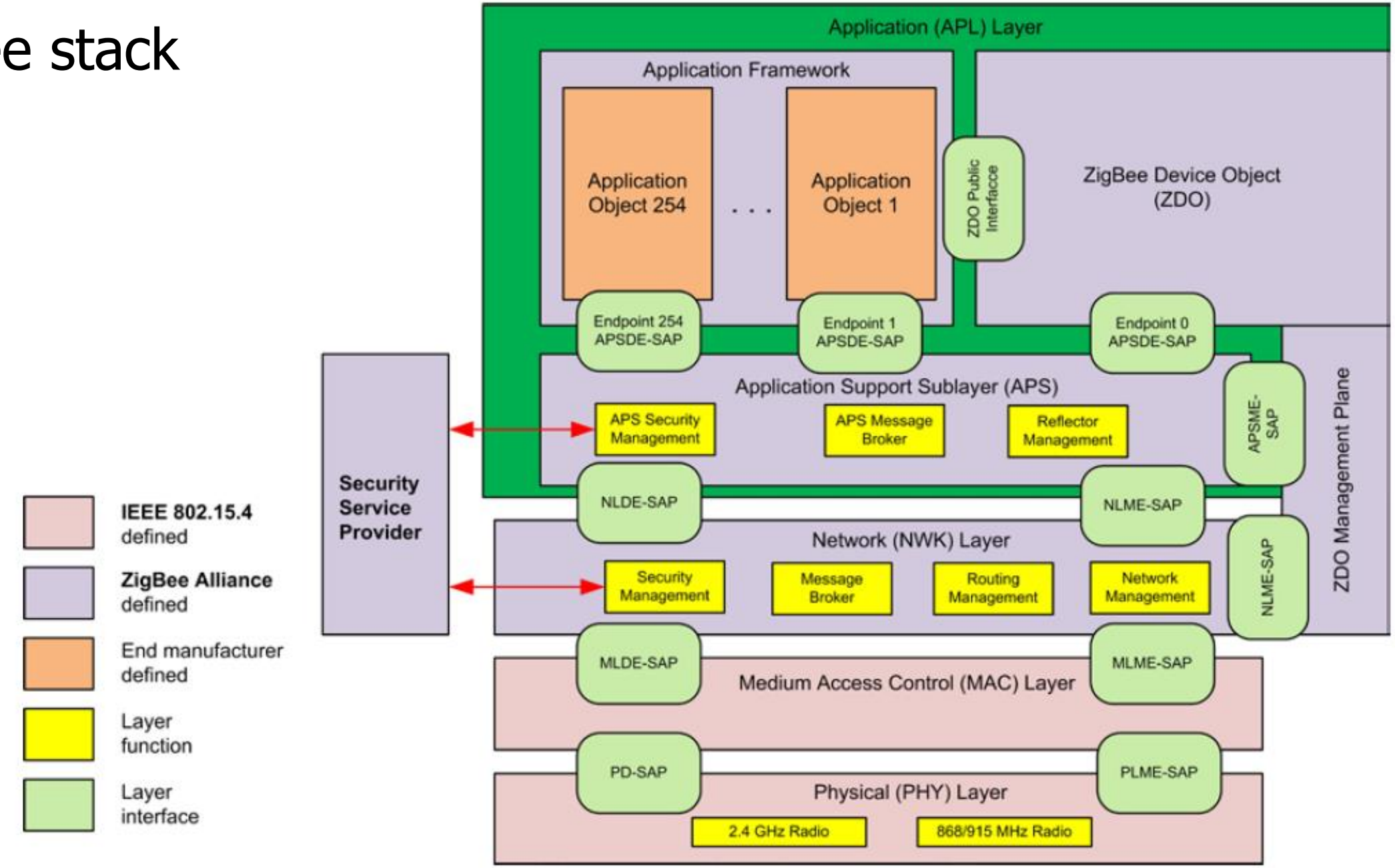
- ZigBee Specification (2015)
- ZigBee Cluster Library Specification (2016)

- Useful resources
  - ZigBee overview: https://www.cse.wustl.edu/~jain/cse574-14/ftp/j_13zgb.pdf
  - NXP library guides (include overview on ZigBee)
    - ZigBee Protocol: https://www.nxp.com/docs/en/user-guide/JN-UG-3113.pdf
    - ZigBee Cluster Library: https://www.nxp.com/docs/en/user-guide/JN-UG-3115.pdf
    - ZigBee Home Automation: https://www.nxp.com/docs/en/user-guide/JN-UG-3076.pdf

Outline

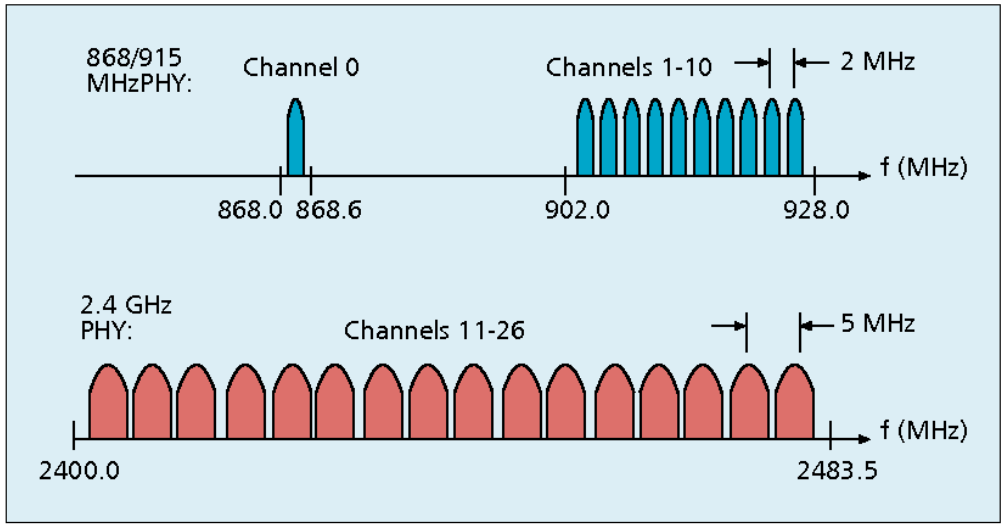
- ZigBee overview
- **ZigBee PHY and MAC**
- ZigBee application layer
- Interoperability

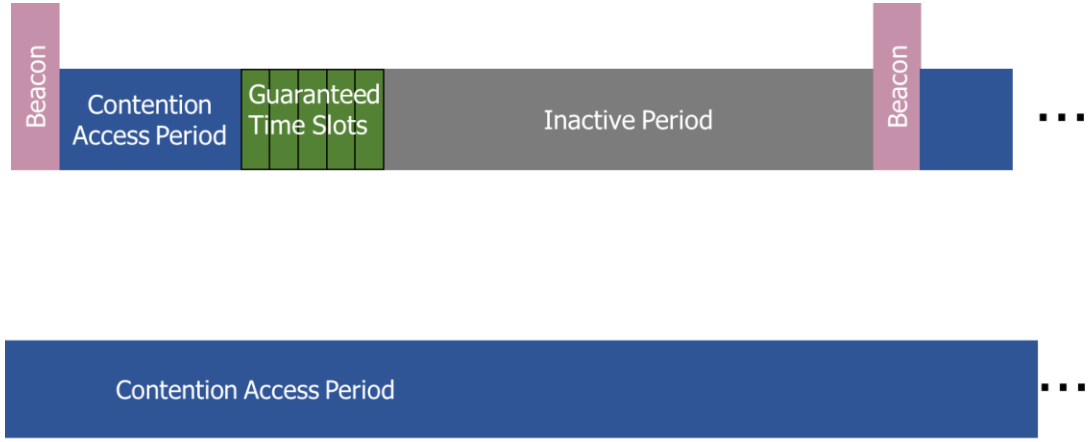
ZigBee stack



Use of 802.15.4

- Basic answer: everything
  - Reuse all of PHY (including non-2.4 GHz channels)
  - Reuse all of MAC (including beacon-enabled network and GTS)
    - Same CSMA/CA mechanism





ZigBee devices (same roles as 802.15.4 defines)

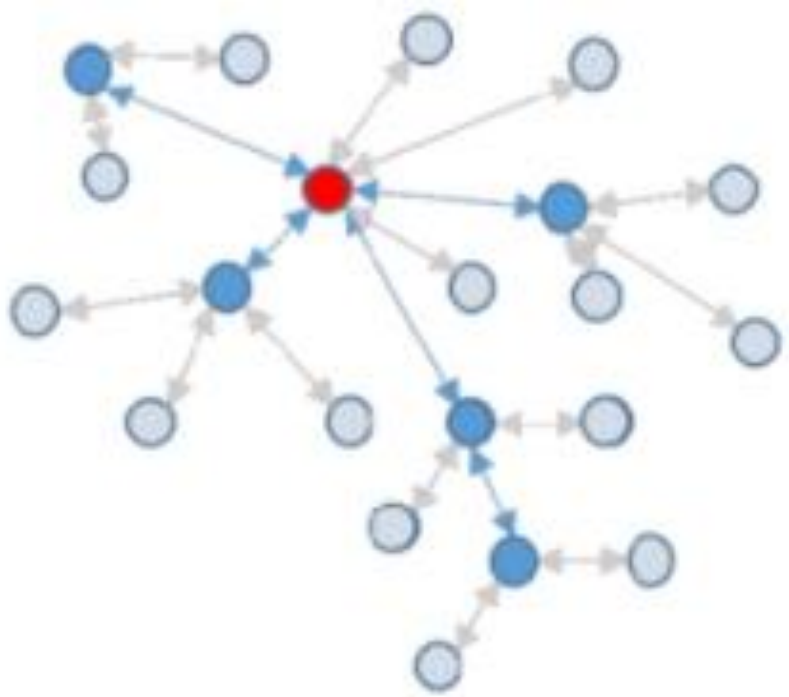
- ZigBee Coordinator (ZC)
  - Starts the network and decides on key parameters
  - Is also a Router

- ZigBee Router (ZR)
  - Higher-power, more-capable devices
  - Radios always on (except during inactive superframe)
  - Connect to one or more children
  - Connect to one or more routers

- ZigBee End Device (ZED)
  - Lower-power, less-capable devices
  - Always a child of one router

Older ZigBee - tree networks

- Original preferred topology



- Uses beacon-enabled network
  - Synchronization via beacon superframes
  - Can reduce power requirements for routers
- Some things get simpler
  - Address assignment is simple
    - If you restrict network size
  - Routing is straightforward
    - But likely more hops for router-to-router communication

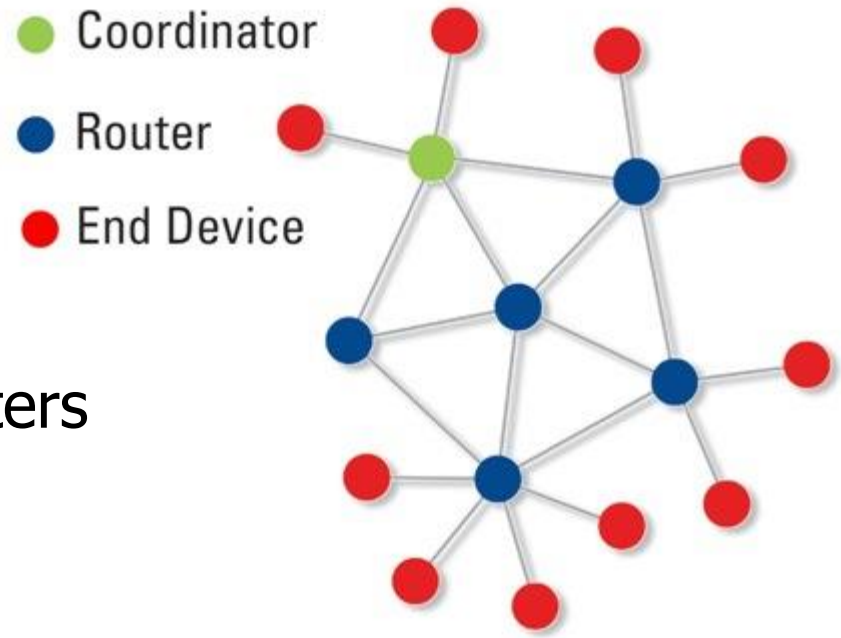
ZigBee tree network complications

- Distributed routing scheme limits topologies
  - There is a limit on number of routers
  - Each router has a maximum number of children
  - There is a maximum limit for router depth
  - Note: Thread has device count limits too!

- Needs a beacon scheduling mechanism
  - Each parent must both participate in a superframe
  - And also send their own superframe beacons
  - Need to keep inactive period large if there is significant router depth
  - Each beacon includes a TX offset field specifying parent beacon time
    - Helps prevent hidden terminal problem

Modern ZigBee – mesh networks

- Presently preferred topology



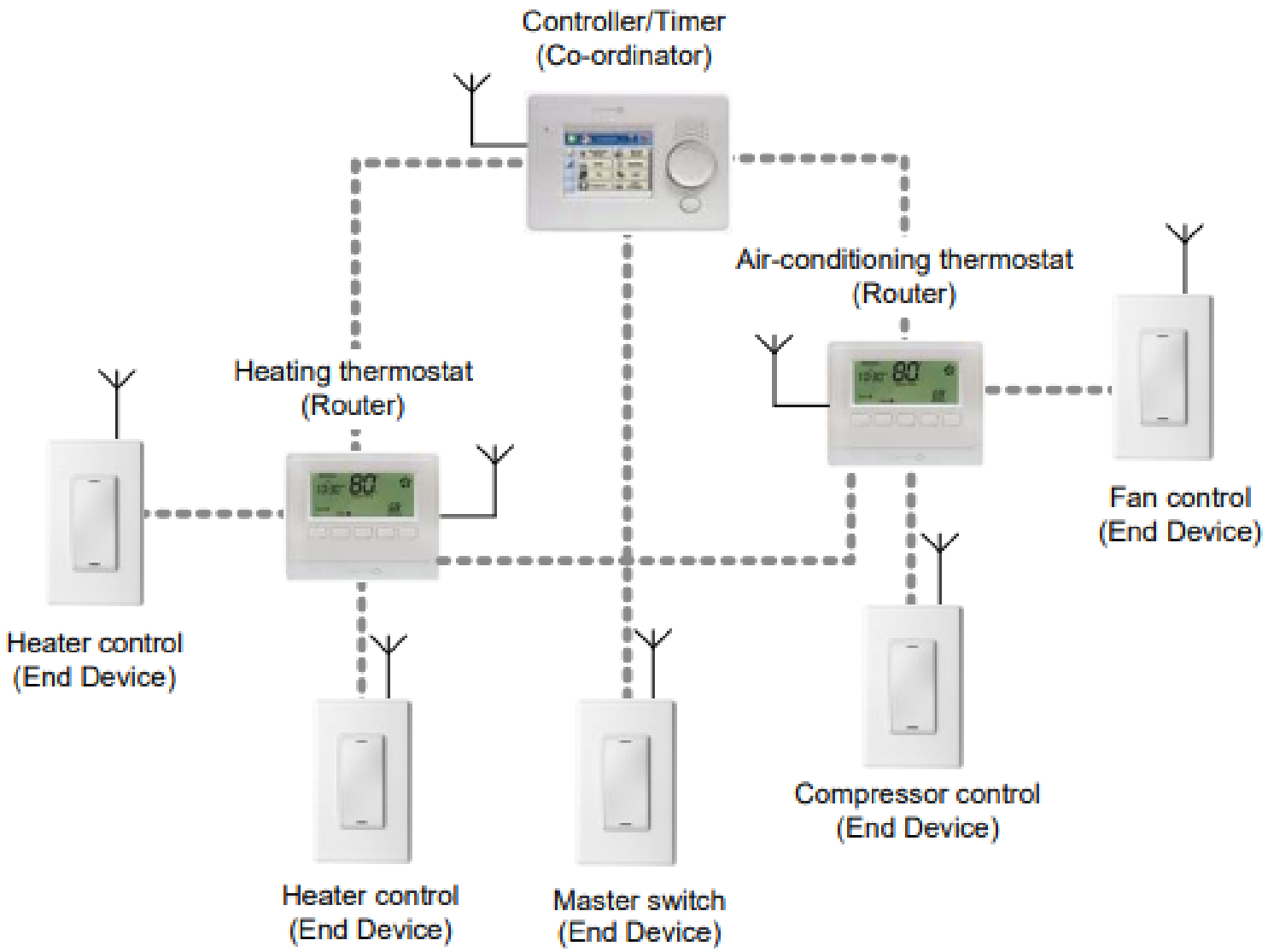
- Uses non-beacon-enabled network
  - All routers are always-on devices
  - Allows arbitrary communication between routers
- Some tradeoffs
  - Higher power routers
  - Routing more complicated (potentially better algorithms though)
  - Addressing more complicated
    - Assign random addresses to each node
    - Include a method for address conflict resolution

ZigBee End Device polling

- Packets are held in ZigBee Routers for up to 7.68 seconds
  - Compare to undefined duration for Thread (at least minutes)
  - Reduction in “low energy” capability for end devices
  - Limiting timeouts makes Router design simpler

- ZigBee codifies polling behavior for End Devices
  - Long Polling - steady state polling period, example: 7.5 seconds
  - Short Polling – polling period while waiting on data, example: 1 second

Example ZigBee network



Break

Outline

- ZigBee overview
- ZigBee PHY and MAC
- **ZigBee application layer**
- Interoperability

ZigBee application-layer terms

- Devices act as servers and clients

- Profiles – details application-level features
  - Includes network configurations
    - For example: security or reliability

  - Includes definitions of various Device Types
    - Specify a collection mandatory and optional Clusters
    - Clusters – collection of Attributes and Commands
      - Attributes – information, readable and/or writable
      - Commands – control, writable, may elicit a response

Analogies between BLE and ZigBee

- BLE Profile
- BLE Service
- BLE Characteristic

- ZigBee Profile + Device Type
- ZigBee Cluster
- ZigBee Attribute
- Also ~ZigBee Commands

ZigBee profiles

- Broad classes of device purposes
  - Contains multiple Device Type definitions

Profile ID	Profile Name
0101	Industrial Plant Monitoring (IPM)
0104	Home Automation (HA)
0105	Commercial Building Automation (CBA)
0107	Telecom Applications (TA)
0108	Personal Home & Hospital Care (PHHC)
0109	Advanced Metering Initiative (AMI)

- Define more features of device than the profiles from BLE
  - Pick various optional network/MAC features, like security or commissioning

ZigBee Device Types

- A collection of Clusters
  - Some mandatory and some optional

- Lists Clusters as Server side or Client Side
  - Server side Cluster is an *input*
  - Client side Cluster is an *output*

- Example: light bulbs implement server, switches implement client

Example ZigBee profile: Home Automation Device Types

Generic Devices
- On/Off Switch
- On/Off Output
- Remote Control
- Door Lock
- Door Lock Controller
- Simple Sensor
- Smart Plug

Intruder Alarm System Devices
- IAS Control and Indicating
- IAS Ancillary Control
- IAS Zone
- IAS Warning Device

- Lighting
- On/Off Light
- Dimmable Light
- Colour Dimmable Light
- On/Off Light Switch
- Dimmer Switch
- Colour Dimmer Switch
- Light Sensor
- Occupancy Sensor

HVAC Devices
- Thermostat

Each bullet point is a **Device Type**

Which is a list of mandatory and optional Clusters

ZigBee Clusters

- A collection of Attributes and Commands
  - Analogous to BLE Services
  - Can be optional or mandatory

- ZigBee Cluster Library defines standard Clusters
  - Lists Attributes and Commands for each
  - Attributes
    - Type – uint8, enum, bitmap, string, etc.
    - Permissions - Read/Write/Report (receive automatic updates)
    - How to interpret meaning of value
  - Commands
    - Field(s), Type of each, Interpretation of each

Example Device Types: door lock and door lock controller

Server (Input) Side	Client (Output) Side
Mandatory	
Basic	
Identify	
Door Lock	
Scenes	
Groups	
Optional	
See Table 1 on page 26	See Table 1 on page 26
Alarms	Time
Power Configuration	OTA Bootload
Poll Control	

Table 6: Clusters for Door Lock

Server (Input) Side	Client (Output) Side
Mandatory	
Basic	Door Lock
Identify	Scenes
	Group
	Identify
Optional	
See Table 1 on page 26	See Table 1 on page 26

Table 7: Clusters for Door Lock Controller

Example Cluster: door lock attributes

Identifier	Name	Type	Access	Def	M/O
0x0000	*LockState*	enum8	Read Only Reportable	-	M
0x0001	*LockType*	enum8	Read Only	-	M
0x0002	*ActuatorEnabled*	bool	Read Only	-	M
0x0003	*DoorState*	enum8	Read Only Reportable	-	O
0x0004	*DoorOpenEvents*	uint32	Read/Write	-	O
0x0005	*DoorClosedEvents*	uint32	Read/Write	-	O
0x006	*OpenPeriod*	uint16	Read/Write	-	O
0x0010	*NumberOfLogRecordsSupported*	uint16	Read Only	0	O
0x0011	*NumberOfTotalUsersSupported*	uint16	Read Only	0	O
0x0012	*NumberOfPINUsersSupported*	uint16	Read Only	0	O
0x0013	*NumberOfRFIDUsersSupported*	uint16	Read Only	0	O
0x0014	*NumberOfWeekDaySchedulesSupportedPerUser*	uint8	Read Only	0	O
0x0020	*EnableLogging*	bool	Read*Write Reportable	0	O
0x0021	*Language*	string (3bytes)	Read*Write Reportable	0	O
0x0022	*LEDSettings*	uint8	Read*Write Reportable	0	O

Table 7-10. *LockType* Attribute Values

Value	Definition
0x00	Dead bolt
0x01	Magnetic
0x02	Other
0x03	Mortise
0x04	Rim
0x05	Latch Bolt
0x06	Cylindrical Lock
0x07	Tubular Lock
0x08	Interconnected Lock
0x09	Dead Latch
0x0A	Door Furniture

Example Cluster: door lock commands (client side)

Command ID	Description	M/O
0x00	Lock Door	M
0x01	Unlock Door	M
0x02	Toggle	O
0x03	Unlock with Timeout	O
0x04	Get Log Record	O
0x05	Set PIN Code	O
0x06	Get PIN Code	O
0x07	Clear PIN Code	O
0x08	Clear All PIN Codes	O
0x09	Set User Status	O
0x0A	Get User Status	O
0x0B	Set Weekday Schedule	O
0x0C	Get Weekday Schedule	O
0x0D	Clear Weekday Schedule	O
0x0E	Set Year Day Schedule	O
0x0F	Get Year Day Schedule	O

Octets	Variable
Data Type	octstr
Field Name	PIN/RFID Code

- Server-side
  - Performs actions when it receives these commands
- Client-side
  - Capable of sending these commands

Example ZigBee profile: Smart Energy

- Interactions with energy providers for efficiency and cost savings

- Devices
  - Energy service interface
  - Metering device
  - Load control device

- Clusters
  - Demand response
  - Metering
  - Price
  - Key establishment (e.g. security)

Server Side	Client Side
Mandatory	
	Demand Response and Load Control
	Time
Optional	
	Price
	Calendar
	Device Management
	MDU Pairing
Energy Management	
Alarms	
Tunneling	Tunneling

Example: demand response cluster

- No attributes, only commands

Command Identifier	Description	M/O
0x00	Load Control Event	M
0x01	Cancel Load Control Event	M
0x02	Cancel All Load Control Events	M

Load Control Command Payload

Octets	4	2	1	4	2	1	1
Data Type	uint32	map16	uint8	UTC	uint16	uint8	uint8
Field Name	Issuer Event ID (M)	Device Class (M)	Utility Enrollment Group (M)	Start Time (M)	Duration in Minutes (M)	Criticality Level (M)	Cooling Temperature Offset (O)

Octets	1	2	2	1	1	1
Data Type	uint8	int16	int16	int8	uint8	map8
Field Name	Heating Temperature Offset (O)	Cooling Temperature Set Point (O)	Heating Temperature Set Point (O)	Average Load Adjustment Percentage (O)	Duty Cycle (O)	Event Control (M)

Endpoints

- Each ZigBee device has a number of Endpoints (up to 240)
  - Number by which remote applications can contact it
  - Analogous to a Port in TCP/UDP

- Each Endpoint has one Device Type attached to it
  - Communication refers to the Endpoint number,
    - Then the Cluster ID within it,
    - Then the Attribute/Command ID within that
  - Endpoints can be queried to determine what they provide

- Special case: Endpoint 0 – ZigBee Device Object
  - All devices must implement the ZigBee Device Object
  - Attributes and Commands for controlling a network device
  - Network parameters are configured just like a light or door lock

Example Endpoints for a device

An example endpoint implementation:

Endpoint # - Profile Name: Device Type

0 - ZigBee Device Profile (ZDP): ZDO

1 - HA: Thermostat

2 - HA: On/Off Output

3 - SE: In-Home Display

4 - MSP: Proprietary vendor extensions

- Even simple devices hopefully have three endpoints:
  1. ZigBee Device Object
  2. <Their functionality>
  3. Over The Air Bootloader (code updates)

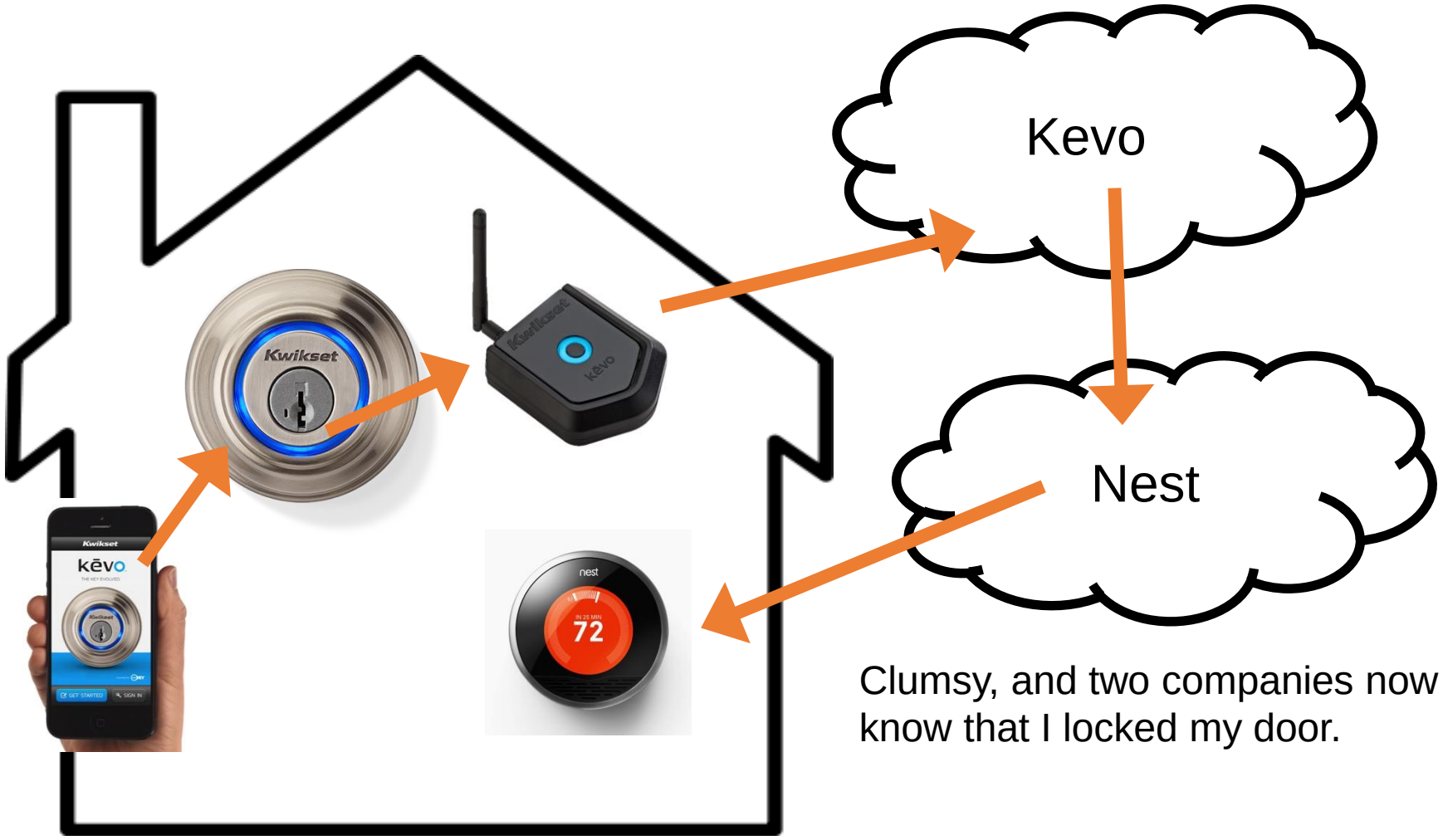
Break + Comparison

- ZigBee
  - Standardized services
  - Low-power end devices
  - Higher power routers
  - Mesh for extended range
  - No interface on most consumer hardware
- BLE
  - Standardized services
  - Low-power end devices
  - Higher power scanner/centrals
  - Star topology for simplicity
  - Compatible with smartphones

Outline

- ZigBee overview
- ZigBee PHY and MAC
- ZigBee application layer
- **Interoperability**

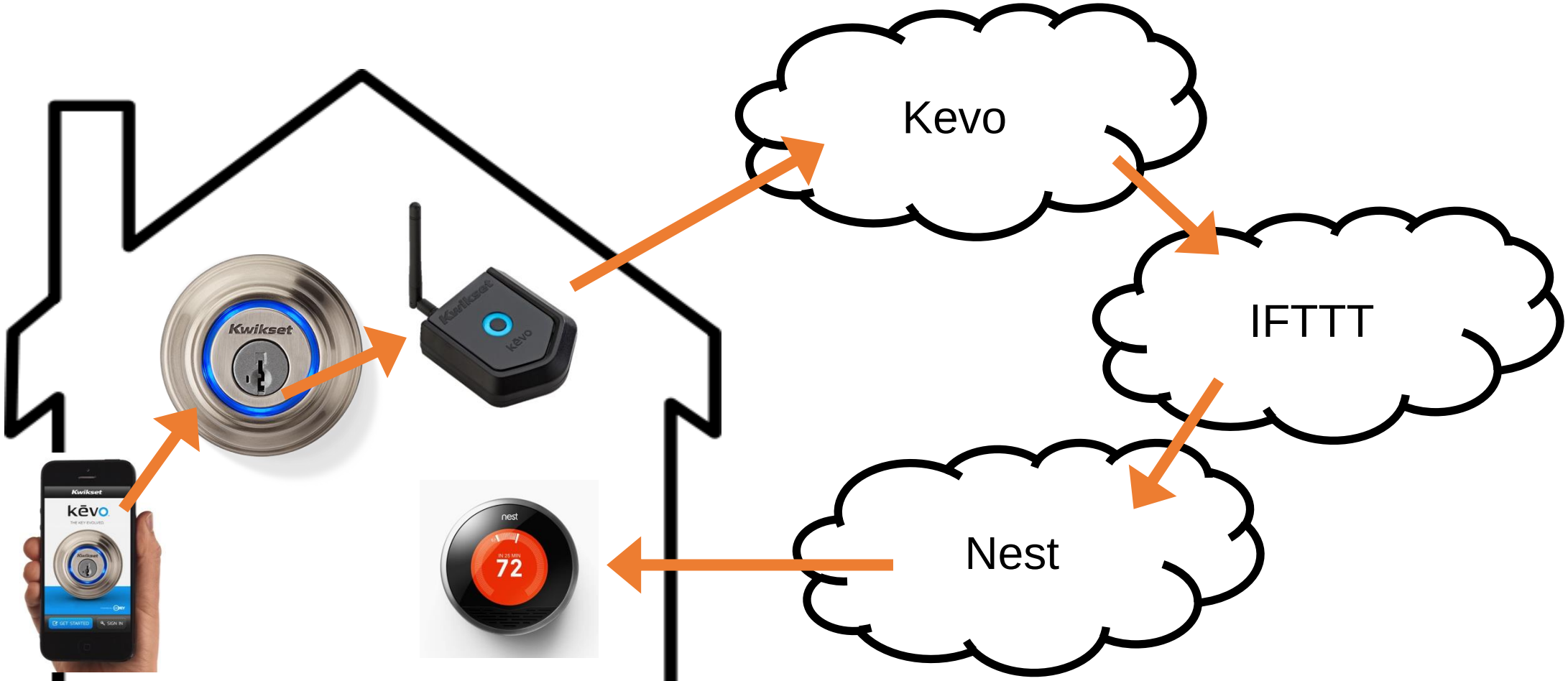
“When I leave, turn down the AC”



Clumsy, and two companies now know that I locked my door.

Slide courtesy of Prabal Dutta

“When I leave, turn down the AC”

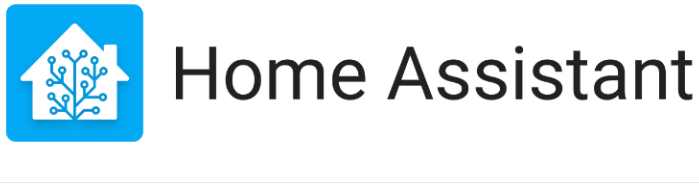


Clumsy, and ~~two~~ three companies now know that I locked my door.

Slide courtesy of Prabal Dutta

What does it look like without three different clouds?

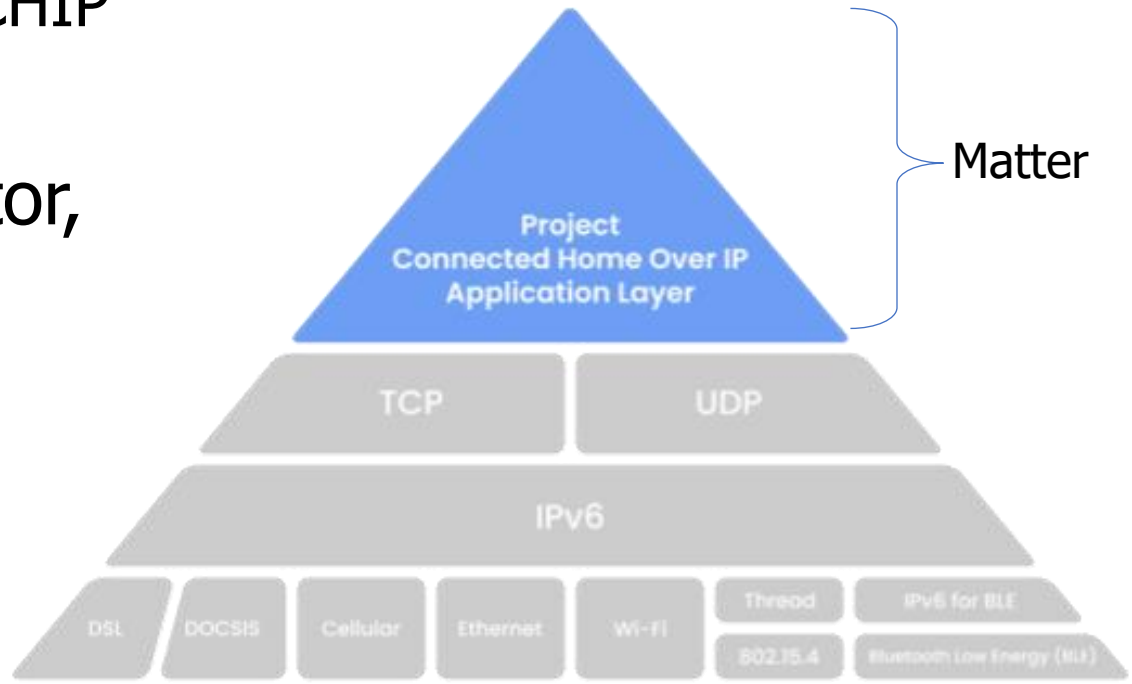
- ”Standardization” is the answer? Custom adaptations?



~~Zigbee~~ Connectivity Standards Alliance today

- 2021 rebrand (why?)
  - Zigbee fading in relevance, utility
  - Zigbee group created/creating new standard: Matter
    - Announced Dec 2019; “first products expected early 2022”
    - Previously known as “Project CHIP”

- Setting up as a Thread competitor, focused on nailing Smart Home
  - Details still a bit murky

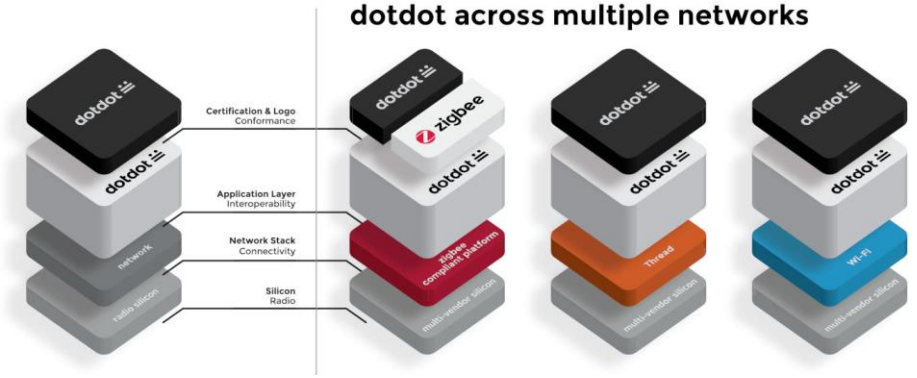


Interoperability, the lack thereof, and CSA's proposed solution

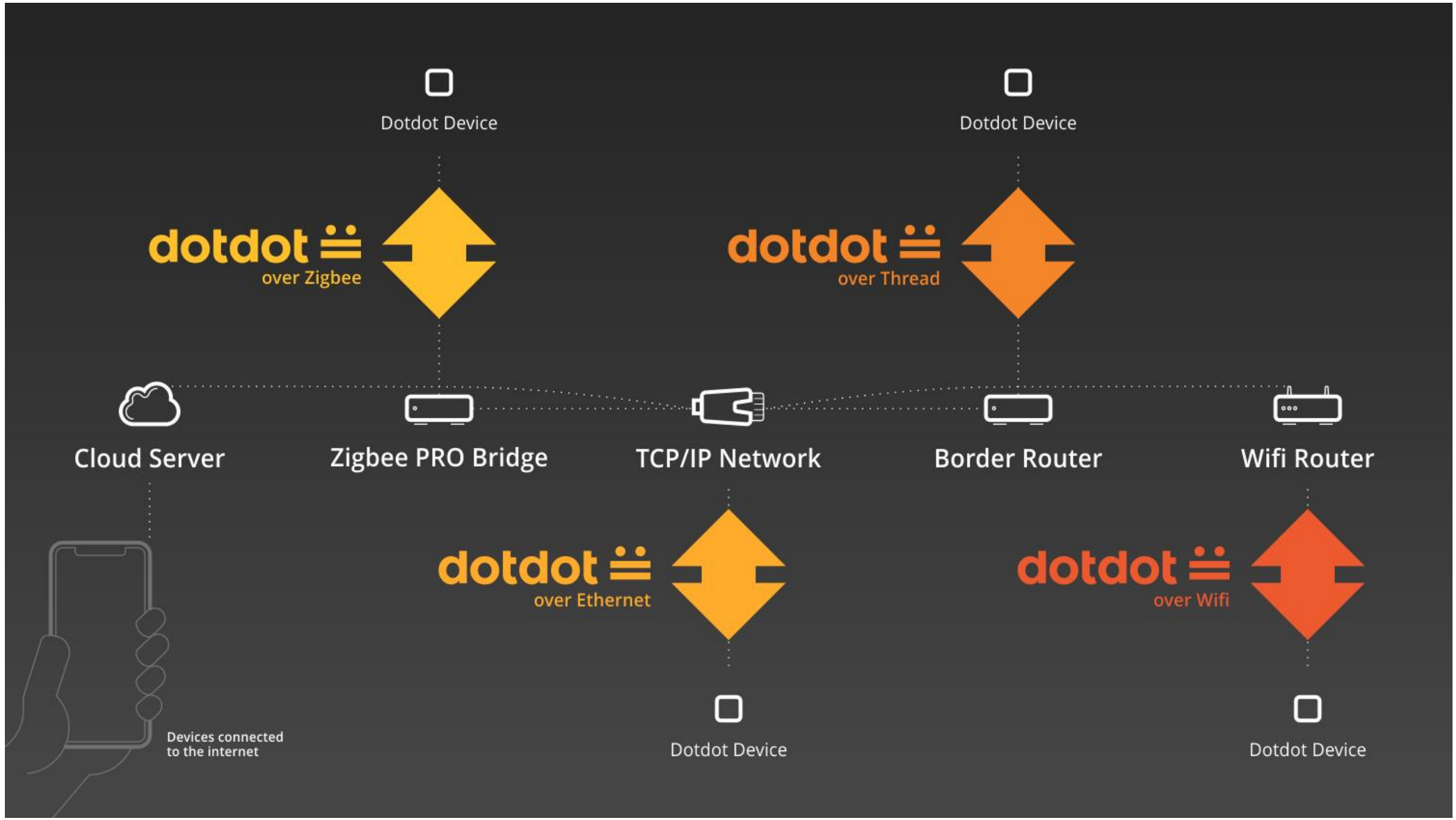
- Zigbee's device provisioning set themselves up for this problem, but also meant they were early to ideas of how to fix it

- The specification for how to interact with devices is far above anything network-specific

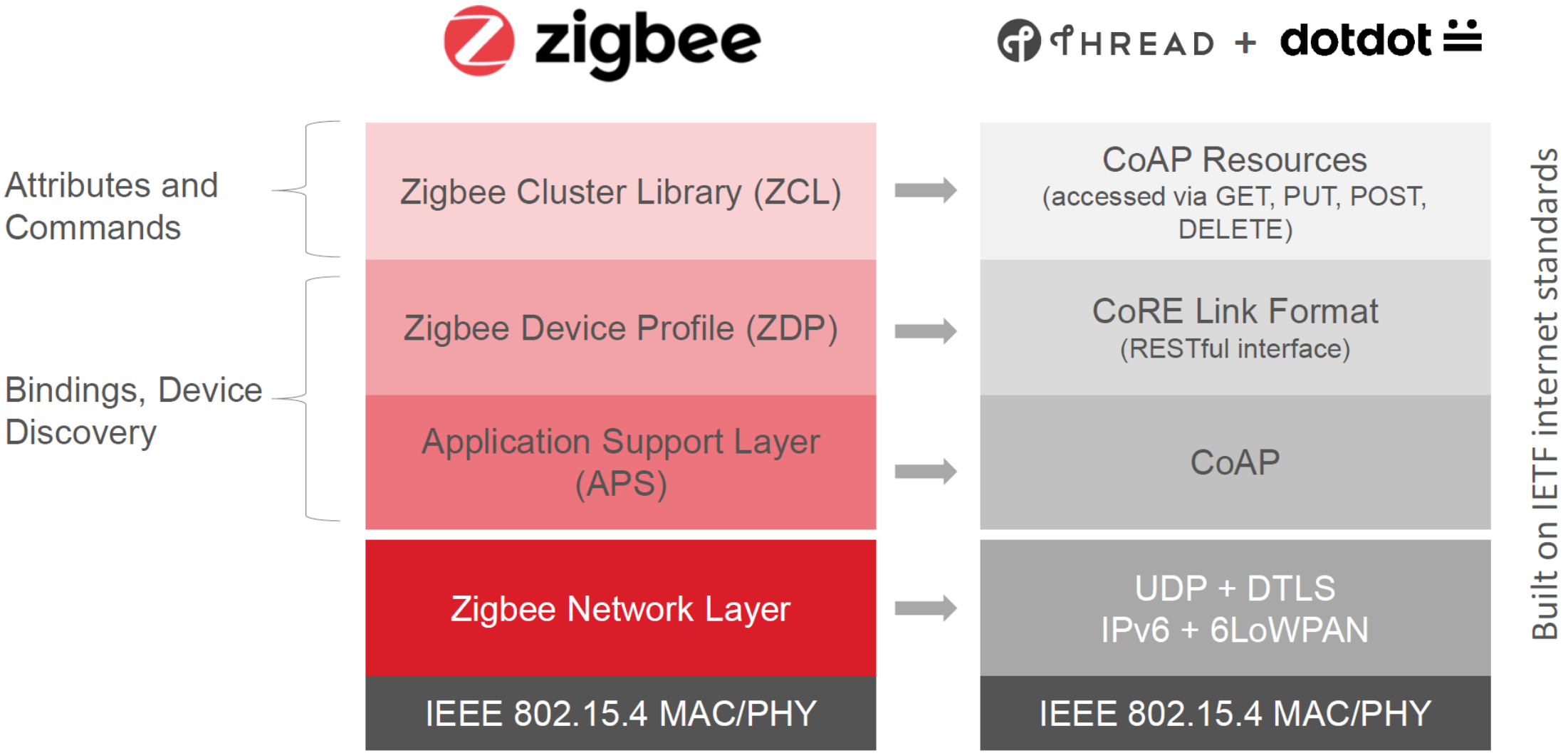
- dotdot is a recent effort to spread ZigBee Clusters more widely
  - Runs same application-layer on top of various lower layers
  - ZigBee, BLE, Thread, WiFi, Ethernet



dotdot provides ZigBee-style control over various networks



Example dotdot over Thread

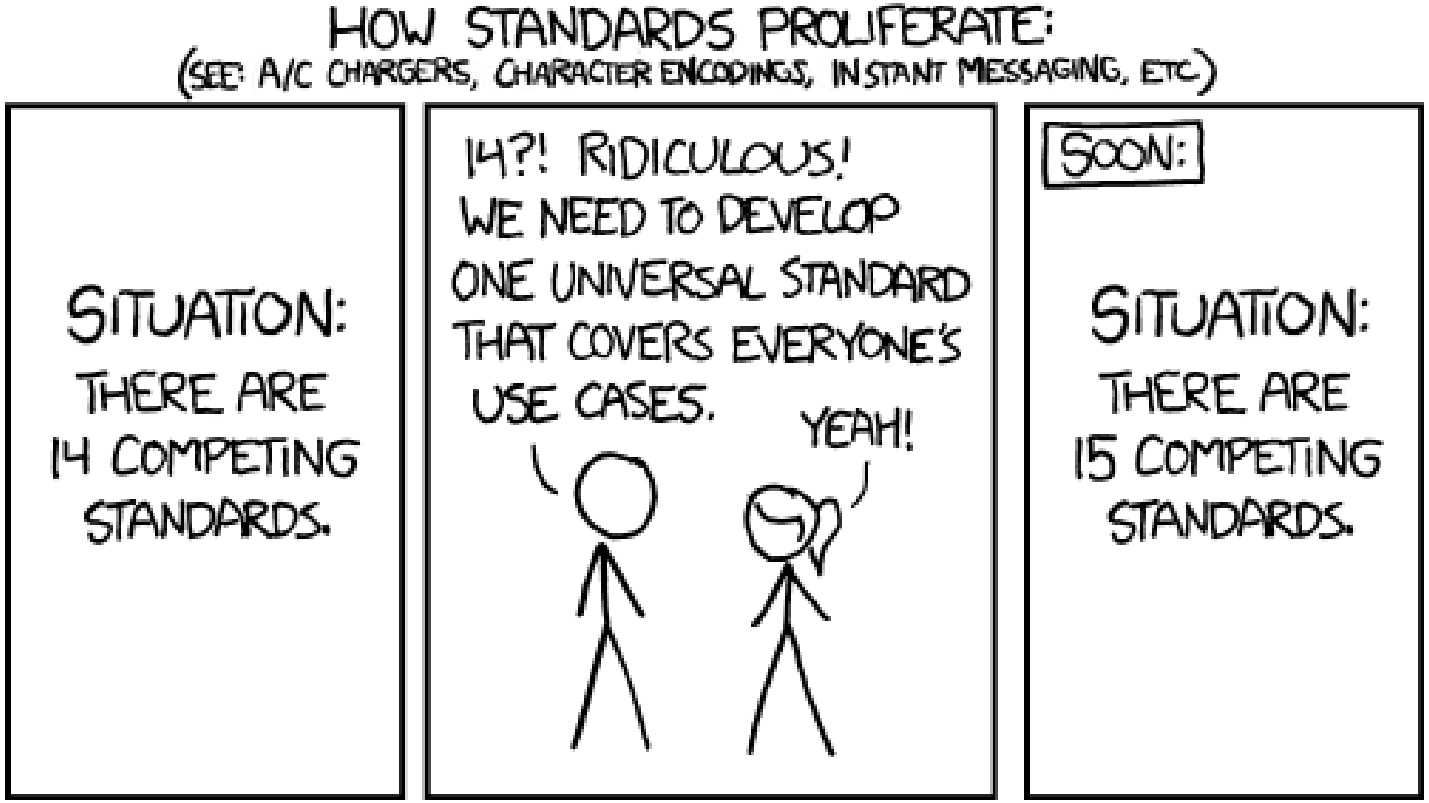


ZCL to CoAP mappings

Resource	Methods	URI
Resource discovery	GET	/zcl
Endpoints	GET	/e
Attributes	GET, PUT, POST	/a
Commands	GET, POST	/c
Bindings	GET, PUT, POST, DELETE	/b
Report Configuration	GET, PUT, POST, DELETE	/r
Report Notification	POST	/n
Group Notification	POST	/g
EZ-Mode Commissioning	GET, POST	/m

Is ZCL the right standard for device interactions?

- Seems better than making something new from scratch



https://xkcd.com/927/

Outline

- ZigBee overview
- ZigBee PHY and MAC
- ZigBee application layer
- Interoperability