MOBILE & CYBER-PHYSICAL SYSTEMS

THE ZIGBEE STANDARD

STEFANO CHESSA

1

LEARNING OBJECTIVES

- ZIGBEE APPLICATION LAYERS
- SECURITY
- CLUSTERS AND PROFILES
- CLUSTER LIBRARY

2

ZIG-BEE

- Power to build a S.Env. (now supporting deployment/maintenance)
  ⇒ minimize intervention of humans
- 2 PARTS of specifications
  1) Messages / behaviours among devices to build a network
  2) Cluster library: standardized application to build with ZigBee
     ↳ specifies how a ZigB device operating in a S.E. should behave

provides build large ntw of sensor / actuators

No TCP/IP: internal mechanism, builds large ntwk → gateway → internet

ZIGBEE

- Standard for wireless sensor networks & IoT
  - Developed and promoted by the ZigBee alliance, of companies (non profit) → IEEE industrial alliance
- Applications: of zig bee (support large # of apps)
  - Home automation (domotics, ambient assisted living,...)
  - 1 Health care
  - 2 Consumer electronics
  - 3 Industrial automation
  - most complex of standardization of ... ...
  - MANS exploration

→ practical scope: on what already exist.

Collaborate to dev this: telecom companies, IoT manufacture, ICT companies. SID and fight on app level. COOPETITION

(1) → delivered 2003, bluetooth was used, before 2003 used STANDARD to enable connection between health device (adopting bluetooth). → when zigbee devices don't success in that market (was taken by bluetooth) → zig bee goes in 2, 3.

ZIGBEE

- Main requirements:
  - 1) Network completely autonomous, no human intervention
  - 2) Very long battery life (a device may run for years)
  - 3) Low data rate (effect of have long battery life) — with sensors you don't need very high data rate
  - 4) Interoperability of ZigBee devices from different vendors
- Main features:
  - Standard-based
  - Low cost devices
  - Can be used globally — frequencies are free world-wide
  - Reliable and self healing → devices can fix problems of network
  - Supports large number of nodes ← tens of thousand
  - Easy to deploy
  - Very long battery life
  - Secure

let you build a local network of IoT devices connected to internet via gateway ⇒ large network of devices with 1 NETWORK ADDRESS, 1 IPv6 address (gateway)
- devices will not use IPv6 (not yet)

IEEE 802.15.4 & ZIGBEE

IEEE 802.15.4 first delivered in 2003 and revised in 2006

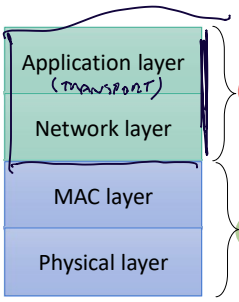
- The first revision of the standard is freely distributed

2003

end of 2004

ZigBee first delivered at the end of 2004, revised in 2006

- Proposed by the ZigBee Alliance, an association of companies
- The first revision of the standard is freely distributed



5

THE IEEE 802.15.4 STANDARD

Specification of the physical and MAC layers for low-rate Wireless Personal Area Networks (PAN).

Infrastructure-less

Short range

Support for star and peer-to-peer topologies

Can coexist with IEEE 802.11 and IEEE 802.15.1 (Bluetooth),...

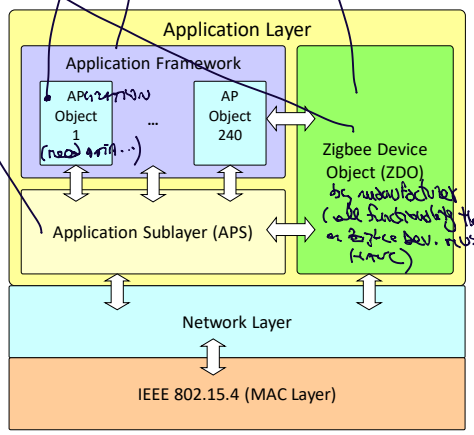
Licence-free frequency bands:

- 868–868.6 MHz (e.g., Europe) with a data rate of 20 kbps;
- 902–928 MHz (e.g., North America) with a data rate of 40 kbps; or
- 2400–2483.5 MHz (worldwide) with a data rate of 250 kbps.

6

THE ZIGBEE STANDARD

- Built upon the IEEE 802.15.4 standard.
- Specifies a network and an application layers



The application layer comprises:

- The Application Framework
  - Contains up to 240 Application Objects (APO)
  - Each APO: a user-defined ZigBee application
- The ZigBee Device Objects (ZDO)
  - Provides services to let the APOs organize into a distributed application.
- The Application Support sublayer (APS).
  - Provides data and management services to the APOs and ZDO.

7

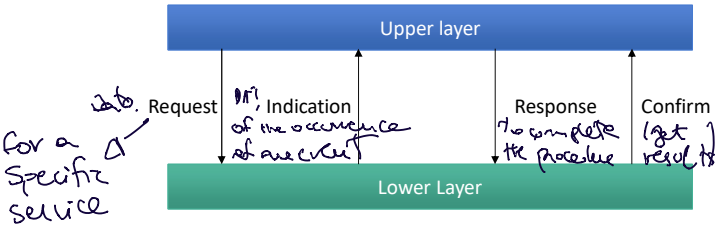
SERVICE PRIMITIVES

Each layer provides its data and management services to the upper layer

Each service is specified by a set of primitives of four generic types:

1) **Request**: It is invoked by the upper layer to request for a specific service;
2) **Indication**: It is generated by the lower layer and is directed to the upper layer to notify the occurrence of an event related to a specific service;
3) **Response**: It is invoked by the upper layer to complete a procedure previously initiated by an indication primitive;
4) **Confirm**: It is generated by the lower layer and is directed to the upper layer to convey the results of one or more associated previous service requests.

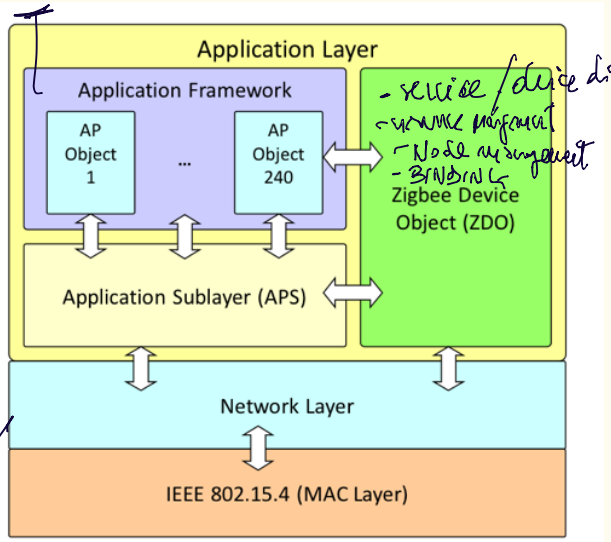
Services may not use all the four primitives



8

• STRUCTURE OF ZIGBEE

KVP endpoint/net. adress



1) Build above 15.4
2) provides APPLICATION and NETWORK LAYER
3) APPLICATION organized in sublayers
   1) APS: sort of "TRANSPORT LAYER" but light
   2) ZDO: - component that defines STANDARD BEHAVIOUR of DEVICE
      - the protocol stack provided by manufacturer that allows you to define your BUSINESS LOGIC, code that provides ALL FUNCTIONALITY for your application (build a network, connect to coord, obtain net adress, communicate with others...)

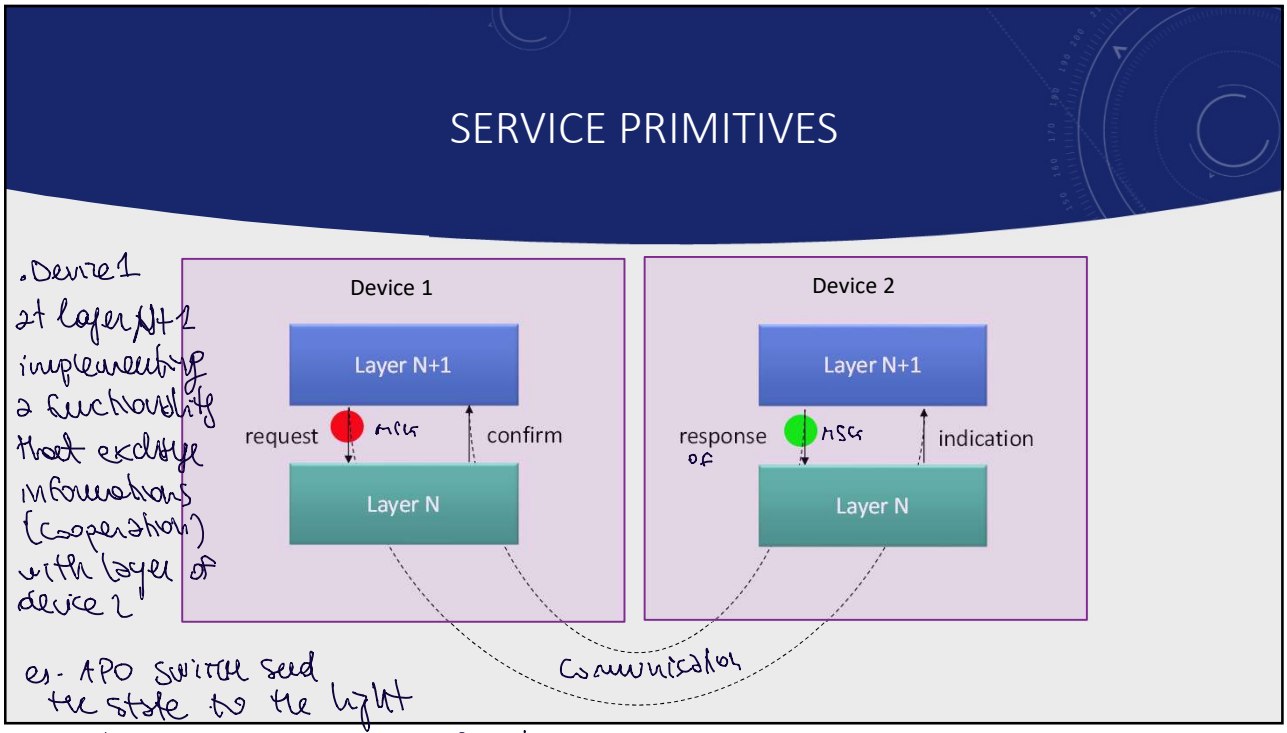
↳ creazione rete
- addressing
- routing
- data transmission
- join/leave

3) APP FRAMEWORK: where you define your business logic with APO
   ↳ pieces of sw that controls sensors, actuator and implement all your functionalities

• All the components interact via INTERFACE
   ⇓
specified by the primitives

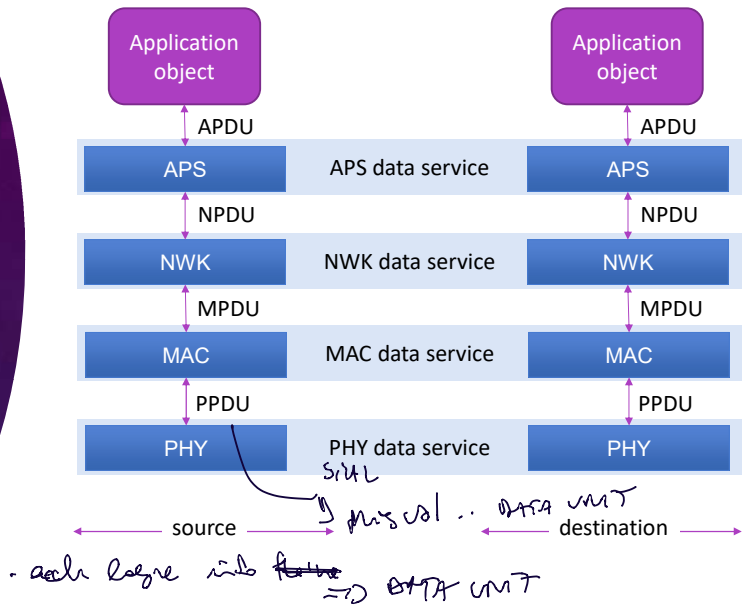
RIRC:
- every single service offered from a layer to another layer, is used by UPPER LAYER using: REQUEST
- then receive INDICATION
- send response
- confirm

SERVICE PRIMITIVES



9

DATA TRANSFER



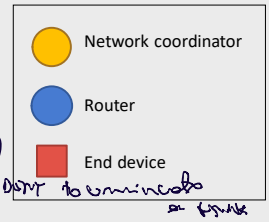
10

NETWORK LAYER

11

NETWORK LAYER

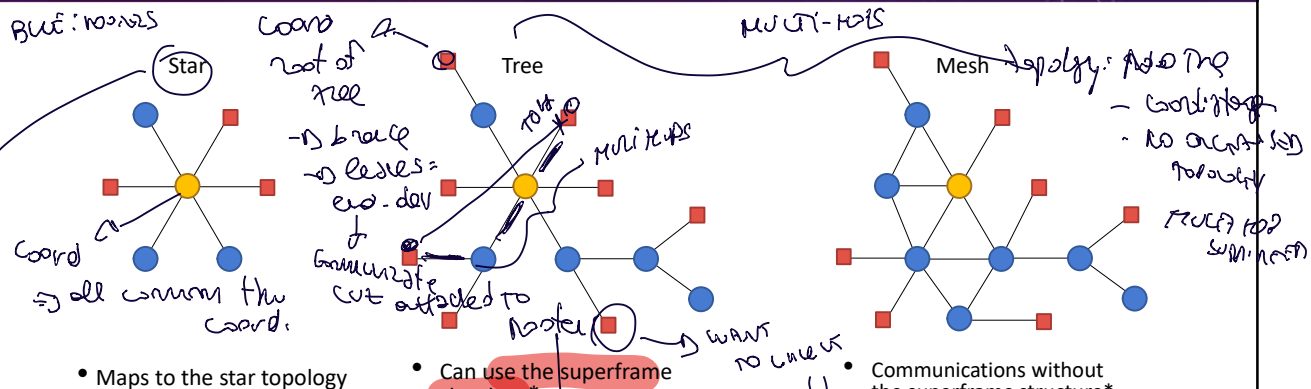
- Three types of devices:
  1) The network coordinator
     - A FFD that creates and manages the entire network.
  2) Routers
     - A FFD with routing capabilities
  3) End-devices
     - Correspond to a RFD or to a FFD acting as simple devices,



FFD: full functional device
RFD: reduced functional device
(See classes on IEEE 802.15.4)

12

NETWORK LAYER TOPOLOGIES



- Maps to the star topology of IEEE 802.15.4
- Uses the superframe*

- Can use the superframe structure*

- Communications without the superframe structure*

* See classes on IEEE 802.15.4

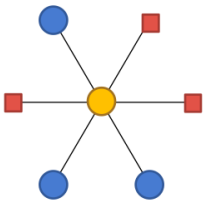
13

NETWORK LAYER

- Services for:
  1) Data transmission (both unicast & multicast)
  2) Network initialization
  3) Devices addressing
  4) Routes management & routing
  5) Management of joins/leaves of devices.

14

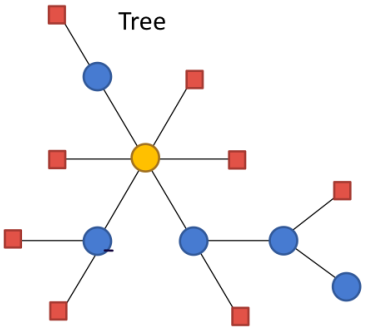
Star



- Maps to the star topology of IEEE 802.15.4
- Uses the superframe*

- coordinator at center: all communications thru it.
- ROUTERS operate as END DEVICE
- (A) MSG 2 HOPS
- (F) BOTTLENECK

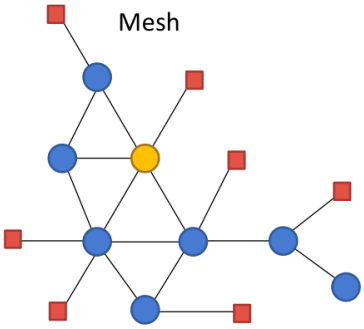
Tree



- Can use the superframe structure*

- END-DEVICES at leaves
- rely on routers/coord to communicate
- ROUTERS may be as leaves

Mesh



- Communications without the superframe structure*

* See classes on IEEE 802

- arbitrary topology, grows in an UNCONTROLLED WAY
- network is P2P
- to communicate, rely on routers

NETWORK LAYER SERVICES

Name	Request	Indication	Confirm	Description
DATA	X	X	X	Data transmission service
NETWORK-DISCOVERY	X		X	Look for existing PANs
NETWORK-FORMATION	X		X	Create a new PAN (invoked by a router or by a coordinator)
PERMIT-JOINING	X		X	Allows associations of new devices to the PAN (invoked by a router or by a coordinator)
START-ROUTER	X		X	(Re-)initializes the superframe of the PAN coordinator or of a router
JOIN	X	X	X	Request to join an existing PAN (invoked by any device
DIRECT-JOIN	X		X	Used by the coordinator or by the routers to force an end device to join their PAN
LEAVE	X	X	X	Leave a PAN
RESET	X		X	Resets the network layer
SYNC	X		X	Allows the application layer to synchronize with the coordinator or a router and/or to extract pending data from it
GET	X		X	Reads the parameters of the network layer
SET	X		X	Set parameters of the network layer

15

NETWORK LAYER

- Before communicating on a network, a ZigBee device must either:
  - Form a new network → ZigBee Coordinator
  - Join an existing network → ZigBee router or end-device
- The role of the device is chosen at compile-time

16

- Coordinator creates a network via: NETWORK-FORMATION.request
- ZDO of coord. (when is turned-on) request to NETWORK layer the ↑

NETWORK FORMATION

create nwk by coordinator
- invoke NWK function of IT's NWK service ↓ by request primitive

- Initiated by the NETWORK-FORMATION.request primitive

NB:

- Invoked by a device that acts as **coordinator** and that is not already in another PAN → & ∉ another network (≠ from bluetooth)

17

③ Once coord find channel => send periodic packet to announce ITS present (BEACON) → ID of network and many mechanism, running on a coordinator

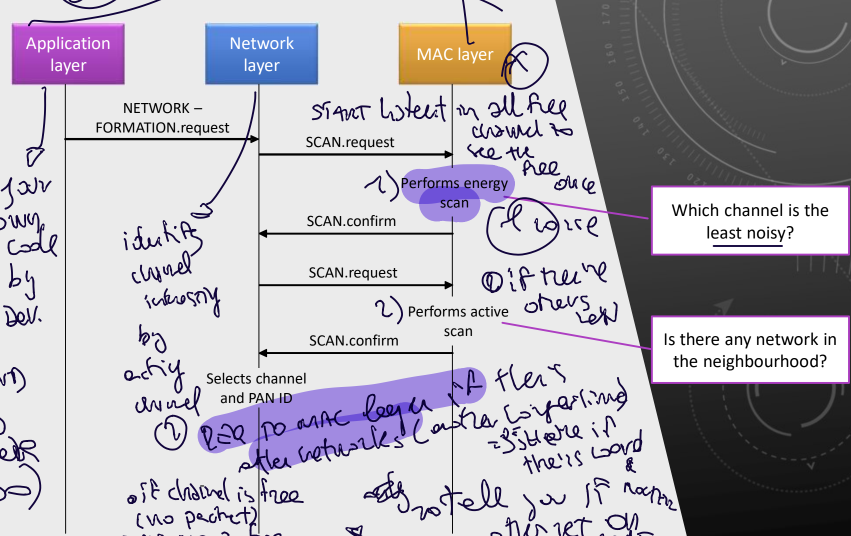
HAS info: channel occupied; net ID (PAN ID); channel free (according to level of noise)

SCANS ON DEVICE running of a single device

NETWORK FORMATION

Flow of command to create nwk

- Uses the MAC layer services to **look for a channel** that does not conflict with other existing networks
- **Selects a PAN identifier** which is not already in use by other PANs



④ NWK assign itself a PAN (16-bit) ≠ from existing network around

⑤

- part of STACK already implemented (ZDO)

implements Behavior

18

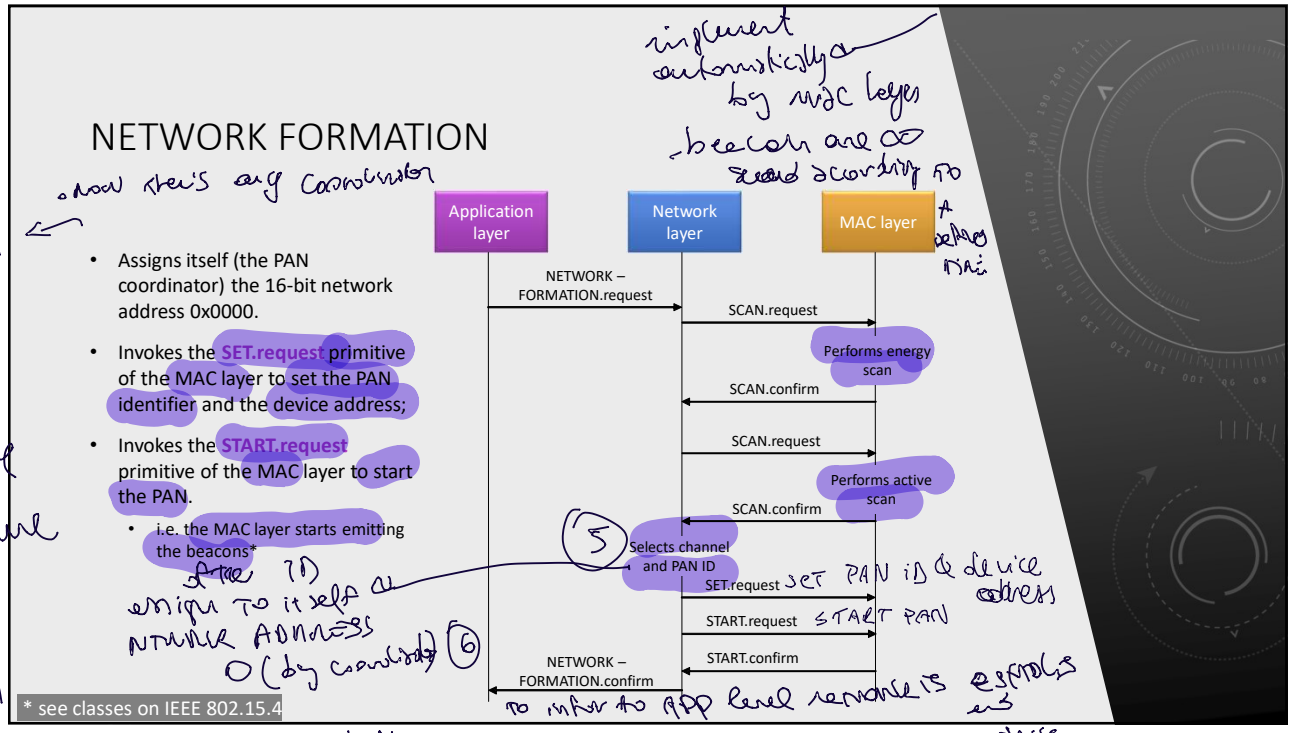
invokes NWK Formation from network layer

① look 4 channel energy scan: report to NWK layer state of all channels; MAC check level of noise

9

NETWORK FORMATION

- Assigns itself (the PAN coordinator) the 16-bit network address 0x0000.
- Invokes the **SET.request** primitive of the MAC layer to set the PAN identifier and the device address;
- Invokes the **START.request** primitive of the MAC layer to start the PAN.
  - i.e. the MAC layer starts emitting the beacons*



* see classes on IEEE 802.15.4

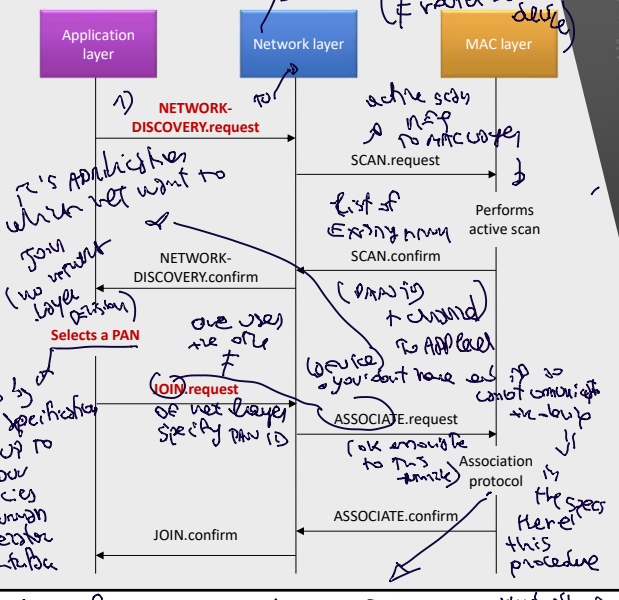
JOINING A NETWORK

- Join through association: initiated by a device wishing to join an existing network
- Direct join: requested by a router or by the coordinator to request a device to join its PAN (direct join).

JOIN THROUGH ASSOCIATION (CHILD-SIDE)

The device wishing to join a PAN:

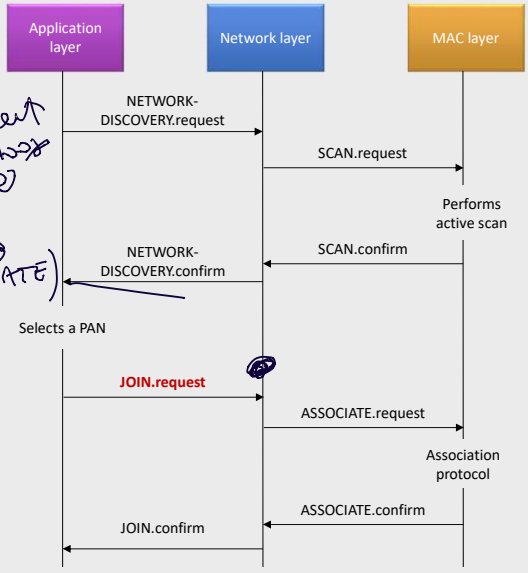
- Performs a NETWORK-DISCOVERY to look for existing PANs.
- Selects a PAN
- Invokes JOIN.request with parameters:
  - The PAN identifier of the selected network
  - A flag indicating whether it joins as a router or as an end device.



21

JOIN THROUGH ASSOCIATION (CHILD-SIDE)

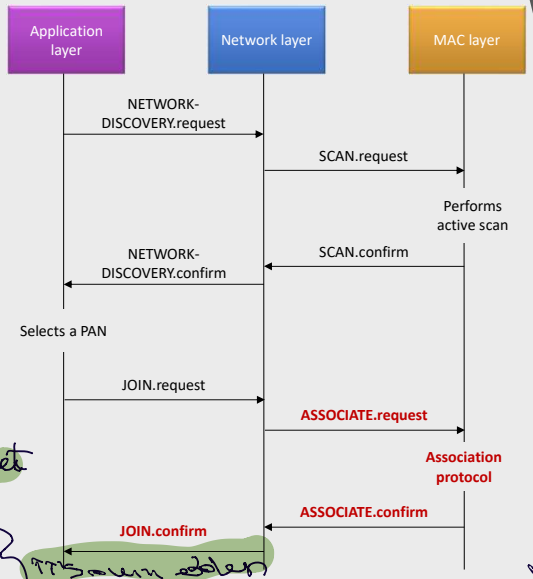
- The JOIN.request primitive in the NWK layer selects a "parent" node *P* (in the PAN) from its neighbourhood.
- In star topologies:
  - *P* is the coordinator
  - the device join as end-device.
- In the tree topology:
  - *P* is the coordinator or a router
  - the device joins as a router or as an end-device



22

JOIN THROUGH ASSOCIATION (CHILD-SIDE)

- The association protocol obtains from *P* a 16-bits short address
- The ASSOCIATE.confirm returns to the NWK layer the short address
- The NWK layer will use the short address in any further communication over the network



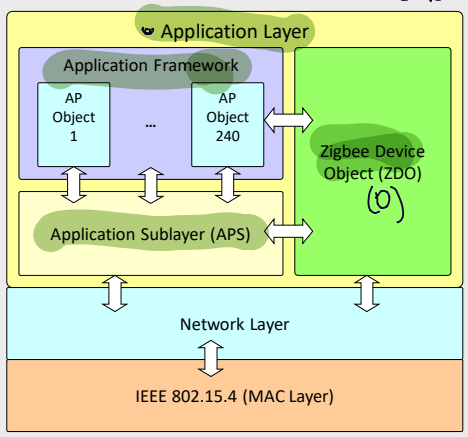
23

ZIGBEE APPLICATION LAYER

40

APPLICATION LAYER

- The Application layer comprises:
  - The Application Framework
    - includes a set of Application Objects (APO)
  - The Zigbee Device Object (ZDO)
    - Manages the application services
  - Application Support Sublayer (APS)
    - Provides data, binding, and discovery services.



41

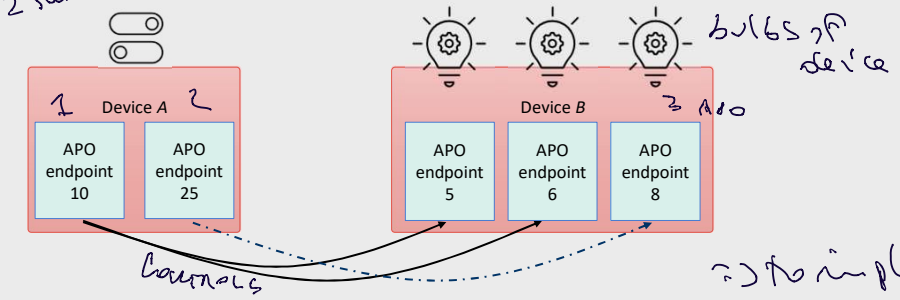
THE APPLICATION FRAMEWORK

- Up to 240 APOs, each corresponding to an application Endpoint
- Endpoint 0 is reserved for the ZDO.
- Each APO in the network is uniquely identified by its endpoint address & the network address of the hosting device.
- Simplest APO can be queried with Key Value Pair data service (KVP)
  - Set/Get/Event transactions on the APOs attributes
  - … native KVP disappeared in the second version of ZigBee
  - but it's now present in the cluster library
- More complex APOs can have complex states and communicate with the Message data service.

42

A SAMPLE APPLICATION

- APOs 5*B*, 6*B*, and 8*B* have a single attribute containing the status of the bulbs (on/off)
  - actuators, control bulbs
  - configured as servers at the application layer
- APO 10 and 25 are switches, and they are configured as clients at the application layer
- The attributes of APOs 5*B*, 6*B*, and 8*B* can be set remotely from the APOs 10*A* and 25*A*



43

APPLICATION SUPPORT SUBLAYER

- The APS frame uses the concepts of endpoints, cluster IDs, profile IDs and device IDs.
- APS provides: services:
  1) Data service (a light transport layer)
     - Filtering out packets (non registered endpoints, profiles that do not match)
     - Generating end-to-end acknowledgments
  2) Management:
     - the local binding table
     - The local groups table
     - The local address map

44

APS: ENDPOINTS

- in each device, endpoints are identified by a number between 1 and 240
- a ZigBee device can run multiple applications
  - one for each APO
- endpoints are seen as virtual wires connecting applications (equivalent to Unix sockets)
- they allow for separate profiles, devices and control points to co-exist within a single node

45

APS: CLUSTERS

- Application meaning:
  - Informally, a cluster provides access to a service (a functionality) of an application object
    - defines an interaction protocol… in the simplest cases just a single message
  - Defines both commands and attributes:
    - Commands cause actions on a device
    - Attributes show the state of a device in a given cluster
- Defined by a 16 bit identifier
  - Example : ID 0x0006 is a cluster that knows how to turn something on/off
- Cluster ID have a meaning within a given profile

46

ZIGBEE CLUSTERS (GENERAL DOMAIN)

Cluster Name	Cluster ID
Basic Cluster	0x0000
Power Configuration Cluster	0x0001
Temperature Configuration Cluster	0x0002
Identify Cluster	0x0003
Group Cluster	0x0004
Scenes Cluster	0x0005
OnOff Cluster	0x0006
OnOff Configuration Cluster	0x0007
Level Control Cluster	0x0008
Time Cluster	0x000a
Location Cluster	0x000b

47

APS: APPLICATION PROFILES

- An application profile is the specification of the behaviour of an application possibly operating on several ZigBee devices.
  - an application profile specifies a set of devices and clusters.
  - each application profile has a unique identification number assigned by the ZigBee alliance.

48

APS: APPLICATION PROFILES

- Profiles can be seen as domain spaces of related applications and devices.
- Profile IDs are 16-bit numbers
  - public profiles range from 0x0000 to 0x7fff
  - manufacturer profiles range from 0xbf00 to 0xffff
- Every message sent (or received) is tagged with a profile ID
- Different application profiles may co-exist in a single ZigBee network

49

APS: APPLICATION PROFILE IDS

Profile ID	Profile name
0101	Industrial Plant Monitoring
0104	Home Automation
0105	Commercial Building Automation
0107	Telecom Applications
0108	Personal Home & Hospital Care
0109	Advanced Metering Initiative

50

• DEVICE IDs: #s that map a textual description to a device

APS: DEVICE IDS

- ZigBee device IDs range from 0x0000 to 0xFFFF.
- They have two purposes:
  - To allow human-readable displays (e.g., an icon is related to a device)
  - Allows ZigBee tools to be effective also for humans:
    - a device may implement the on/off cluster, but you don't know whether it is a bulb or a oven…
    - … you only know you can turn it on or off.
    - The device ID tells you what it is, but it does not tell you how you can communicate with it…
    - … that is given by the IDs of the clusters it implements!
- ZigBee discovers services in a network based on profile IDs and cluster IDs, but not on device IDs

51

APS: DEVICE IDS: in the sense of a given app profile

Name	Identifier	Name	Identifier
Range Extender	0x0008	Light Sensor	0x0106
Main Power Outlet	0x0009	Shade	0x0200
On/Off Light	0x0100	Shade Controller	0x0201
Dimmable Light	0x0101	Heating/Cooling Unit	0x0300
On/Off Light Switch	0x0103	Thermostat	0x0301
Dimmer Switch	0x0104	Temperature Sensor	0x0302

example from the Home Automation Profile

52

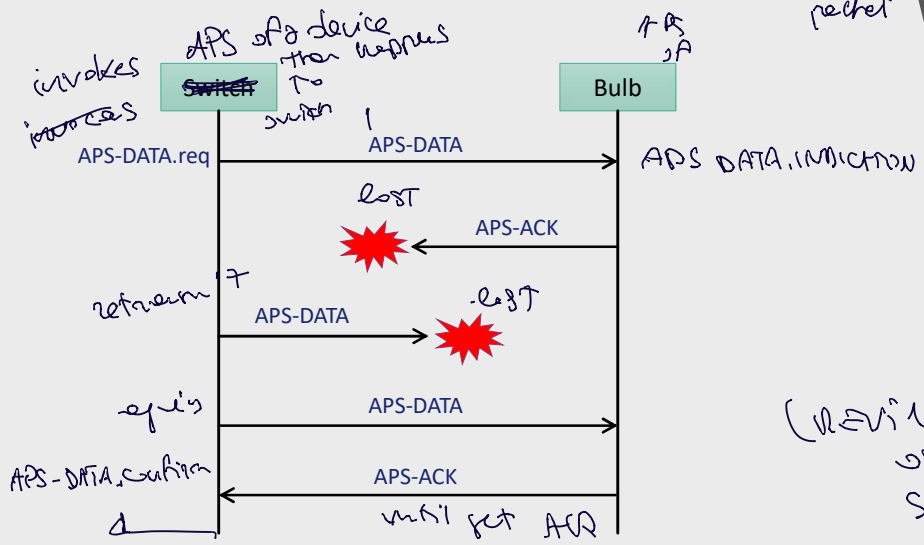
APPLICATION SUPPOR SUBLAYER SERVICES

APS SERVICES

- Provides:
  - data service to both the APOs and the ZDO.
  - binding service to the ZDO
  - group management services
- APS data services enables the exchange of messages between two or more devices within the network.
- The data service is defined in terms of the primitives:
  - request (send),
  - confirm (returns status of transmission) and
  - indication (receive).

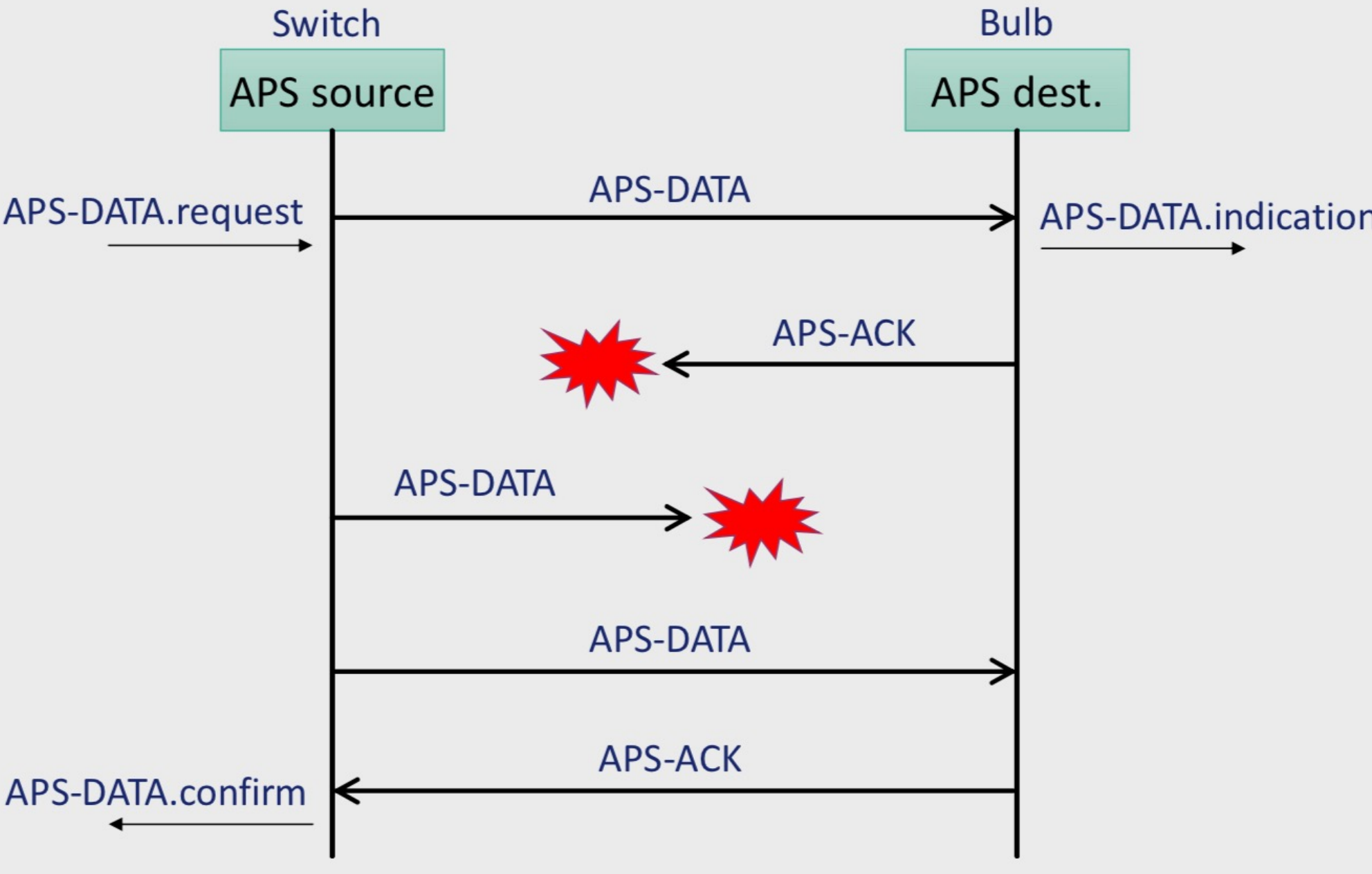
53

APS DATA SERVICE RELIABILITY



54

APS DATA SERVICE RELIABILITY



- Group MNGTMNT on Zigbee support MULTICAST: if a switch controls 10 lights in 10 devices
  => easy way to do is multicast: change state of a switch
  - the APO request to APS to send in multicast 1 msg to 10 devices

=> to support MULTICAST at NETWORK LAYER, have to support creation of groups

APS GROUP MANAGEMENT

(to implement multicast have to create groups to APOs in the route table)

The group management provides services to build and maintain groups of APOs

- A group identified by a 16-bits address
- Each APO in the group identified by the pair network address/endpoint
  - ZDO: tells to add/remove a device from a Group
- ADD-GROUP primitive to add an APO to a group and REMOVE-GROUP primitive to remove an APO from a group
  - takes group number and endpoint number
  - If the group does not exists it is created
- Information about groups in a group table in the APSs

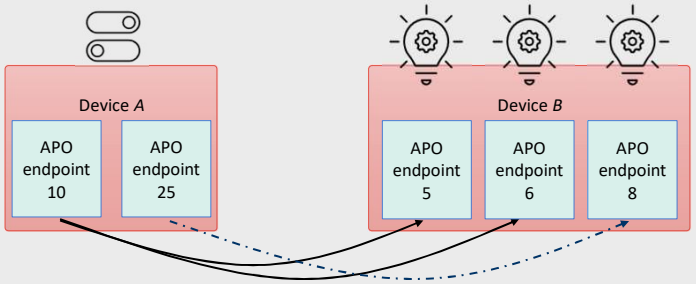
55

- BINDING: - when join to a net, you get a NETWORK ADDRESS. if 2 device stop working and turn on (100)
  => have to RECONNECTING AGAIN and it will have a ≠ NETWORK adress (200) => SOLVE the PROBLEM by BINDING

=> BINDING: 1) Create Links (binds) among APO
2) Some device will be to low-powered to be able to form a complete MSG

APS BINDING

- MSG (to be send read): - source's address - DEST ADDRESS - source's APO - DEST APO
- Allows an endpoint on a node to be connected (bound) to one or more endpoints on other nodes.
- Binding is unidirectional
- Can be configured only by the ZDO of the coordinator or of a router.



- uses mac adresses that don't change (mapping MAC to NETWORK ADRESSES)
- MAC for this only

56

- BINDING: creates a link in the network so that, when an APO generates a MSG, without specify the destination, with binding will be forwarded to destination APO
- So keeps in a TABLE the corrispondence between SOURCE - DESTINATION (association) (association in term of MAC address)

APS BINDING

The binding provides a way to implicitly specify the destination of messages (**Indirect addressing**)

- A message is normally routed to the destination APO based on its address pair <destination endpoint, destination network address> (**direct addressing**)
- Direct addressing might be unsuitable for extremely simple devices
- Indirect addressing exploits binding tables

57

Cluster: access to a source of an APO
- Commands
- Attributes
- Interface protocol

APS BINDING:

BIND and UNBIND primitives:

- BIND.request creates a new entry in the local binding table
  - Takes in input the tuple <source address, source endpoint, cluster identifier, destination address, destination endpoint>
- UNBIND.request deletes an entry from the local binding table.

58

APS BINDING

Indirect addressing

- By the binding table:
  - matches **source address** <network addr, endpoint addr> and the **cluster identifier** into the pair: <destination endpoint, destination network addr>
- The binding table:
  - is stored in the APS of the ZigBee coordinator and/or of the routers
  - it is updated on explicit request of the ZDO in the routers or in the coordinator.
  - it is usually initialised at the network deployment

BINDING TABLE

- Example: a APS-DATA.req of cluster 6 from EP 5 on node 0x3232… (indirect) will generate 3 data requests:
  - node 0x1234… endpoint 12
  - multicastcast to group 0x9999
  - node 0x5678… endpoint 44

Src Addr (64 bits)	Src EP	Cluster ID	Dest Addr (16/64 bits)	Addr/Grp	Dest EP
0x3232…	5	0x0006	0x1234…	A	12
0x3232…	6	0x0006	0x796F…	A	240
0x3232…	5	0x0006	0x9999	G	–
0x3232…	5	0x0006	0x5678…	A	44

APS ADDRESS MAP

- The APS layer contains the address map table.
- Associates the 16 bit NWK address with the 64 bit IEEE MAC address
- Zigbee end devices (ZED) may change their 16 bit NWK address (e.g. they leave and join again). In that case an announcement is sent on the network and every node updates its internal tables to preserve the binding.

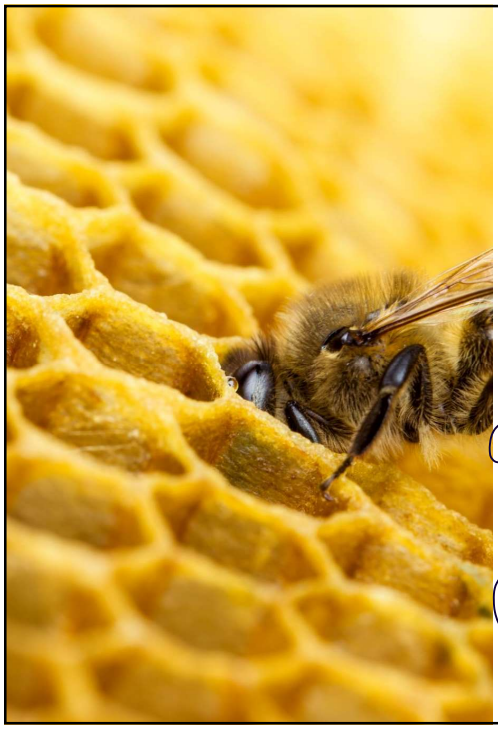
61

APS ADDRESS MAP

IEEE Addr	NWK Addr
0x0030D237B0230102	0x0000
0x0030B237B0235CA3	0x0001
0x0031C237b023A291	0x895B

Example of APS address map

62



Question

Consider the binding table and the address maps shown below that represent the state of the ZigBee network at time *t*. Assume device 0X0022 disconnects from the network at time *t'*>*t* and it reconnects again at time *t"*>*t'*, obtaining network address 0X0003. Discuss to what network address and endpoint are delivered the following messages:

1. Time *h*<*t'* : message of cluster 0X0006 generated by device 0X0001 from endpoint 5
2. Time *h*∈[*t'*,*t"*]: message of cluster 0X0006 generated by device 0X0000 from endpoint 5
3. Time *h*>*t"* : message of cluster 0X0006 generated by device 0X0022 from endpoint 4
4. Time *h*>*t"* : message of cluster 0X0006 generated by device 0X0001 from endpoint 6

Src Addr (64 bits)	Src EP	Cluster ID	Dest Addr (64 bits)	Addr/Grp	Dest EP
0x00...02	5	0x0006	0x22...91	A	12
0x10...A3	6	0x0006	0x22...91	A	240
0x10...A3	5	0x0006	0x00...02	A	34
0x22...91	4	0x0006	0x10...A3	A	44

IEEE Addr	NWK Addr
0x0030D237B0230102	0x0000
0x1030B237B0235CA3	0x0001
0x2231C237b023A291	0x0022

63

ZIGBEE DEVICE OBJECT (ZDO)

- ZDO is a special application attached to endpoint 0
- Implements ZigBee End Devices, ZigBee Routers and ZigBee Coordinators.
- It is specified by a special profile, the **ZigBee Device Profile**:
  - It describes the clusters that must be supported by any ZigBee device.
  - It defines how the ZDO implements the services of discovery and binding and how it manages the network and the security.

64

ZIGBEE DEVICE OBJECT (ZDO)

ZDO services:

1) • Device and service discovery
2) • Binding management
3) • Network management
4) • Node management

65

ZDO: DEVICE AND SERVICE DISCOVERY

- The ZigBee Device Profile (ZDP) specifies the device and service discovery mechanisms
  - To obtain any information about devices and services in the network.
- Device discovery
  - Allows a device to obtain the (network or MAC) address of other devices in the network.
  - Hierarchical implementation:
    - a router returns to its parent its address and the address of all the end devices associated to itself
    - the coordinator returns the address of its associated devices

66

- SERVICE DISCOVERY

1) Query based on: PROFILE ID; CLUSTER ID; addresses, device ID

- Can be:
  1) BROADCAST:
     - query goes to coordinator
     - propagates along the tree
     - coord. collects everything
     - coord. provides report
       (like: active APSs over each network address, cluster the APDs responds...)
  2) UNICAST: directed to a single device

- AFTER DEVICE/SERVICE DISCOVERY HAVE A MAP of the network

- NB.: YOU DON'T KNOW WHERE DEVICES ARE PHISICALLY DEPLOYED
  (have to act on them physically)

67

=> still don't know where a device is
(I know it's a sensor, not where it is).

=> with ZDO: network topology and cluster the devices implement

ZDO: BINDING MANAGEMENT

identify cluster invocated to a single dev and make an operation (like flash)

implement service/device discovery

- The ZDO processes the binding requests received from local or remote EP
  - To adding entries in the APS binding table.
  - To delete entries from the APS binding table.
- Requires an IEEE address

to ZDO of coord: which forward query to ZDO of routers

coordinator ZDO invoke primitive with APS to build an binding table

68

ZDO: MANAGING NETWORK AND NODES

Join/leave req. by coord

- Network management
  - Implements the protocols of the coordinator, a router or an end device according to the configuration settings established either via a programmed application or at installation.
- Node management
  - The ZDO serves incoming requests aimed at performing network discovery, retrieving the routing and binding tables of the device and managing joins/leaves of nodes to the network.

ZDO has to manage the incoming requests like secure/device discovery or join/leave

69

Also if Join requires AUTH, provided by ZDO of router/coordinator

ZIGBEE CLUSTER LIBRARY (ZCL)

70

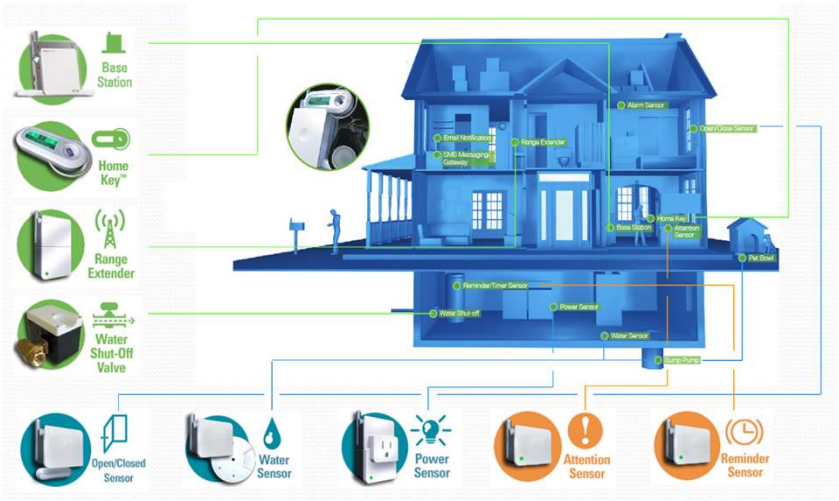
CLUSTER LIBRARY: specifies behaviour of applications

Cluster libraries: describe end behaviour of device in that smart environment Domain (eg. home automation)

ZCL PURPOSES

- ZCL is a repository for cluster functionalities
- a "working library" with regular updates and new functionalities
- ZigBee developers are expected to use the ZCL to find relevant cluster functionalities to use for their applications
  - Avoid re-inventing the wheel
  - Supports interoperability : if 2 ED ≠ implementation has just device/service discovery and can't interact each other at application level
    → solution Cluster library
  - Facilitates maintainability

71



AN APPLICATION EXAMPLE: HOME AUTOMATION

72

read

1 dev. c → specifying msg to require ~~to~~ from c to s

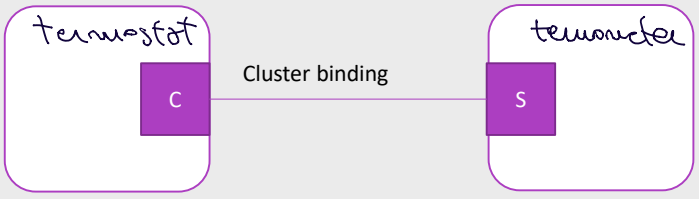
1 dev. s → sensor/actuators that keep internal state

model of cluster lib c/s

THE ZCL CLIENT-SERVER MODEL

SERVER is the device that keep the information that is queried

- A cluster is a collection of commands and attributes, which define an interface to a specific functionality
- The device that stores the attributes is the *server* of the cluster
- The device that manipulates the attributes is the *client* of the cluster



73

- every device can be C/S at same time

FUNCTIONAL DOMAINS

domain

space for all device

- **General**: to access and control attributes of any device, irrespective of their functional domain
- **Closures**: for shade controllers, door locks etc
- **HVAC**: for pumps (fan, heating, dehumidification etc.)
- **Lighting**: to control lights
- **Measurement and sensing**: illuminance, presence, flow, humidity etc.
- **Security and safety**: security zone devices etc.
- **Protocol interfaces**: to interconnect with other protocols

specific functionality

es. alarm infrared

designed to interoperability with other protocols

74

- Profile, functional domain, cluster that fits functional domain
  => cluster defines a protocol

Libraries define: COMMANDS and ATTRIBUTES
cluster

• REQUESTS ARE in terms of commands

ZCL defines: command & attributes

COMMANDS

~ Commands are messages in a format defined by the ZCL	- Commands used to manipulate attributes:	Commands used for dynamic attribute reporting
• Include a header and a payload	• are typically sent from the client to the server • the server responses are received by the client	• i.e., the report attribute command • are typically sent from the server device (where the attribute data itself is stored)… • to the client (that is usually bound to the server device)

75

TYPES OF COMMANDS

1) Read/write an attribute

2) Configure a report and read a reporting response
- Request periodic attribute/config. readings
- Request for report whenever an attribute/configuration changes
- Defines parameters of the reporting (duration, period, minimum change etc.)

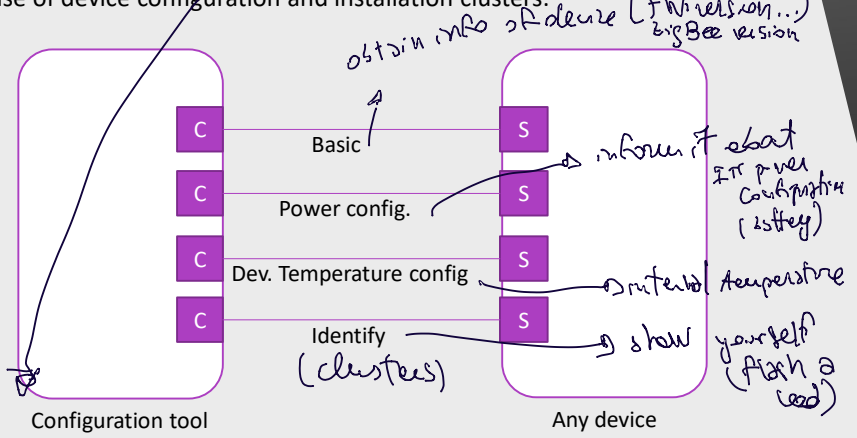
3) Discover attributes
- To discover the ids and types of the attributes of the cluster that are supported by a server

76

clusters to configure devices: configurator tool (operated to a PC)

SCHEMES OF USE

Typical use of device configuration and installation clusters:

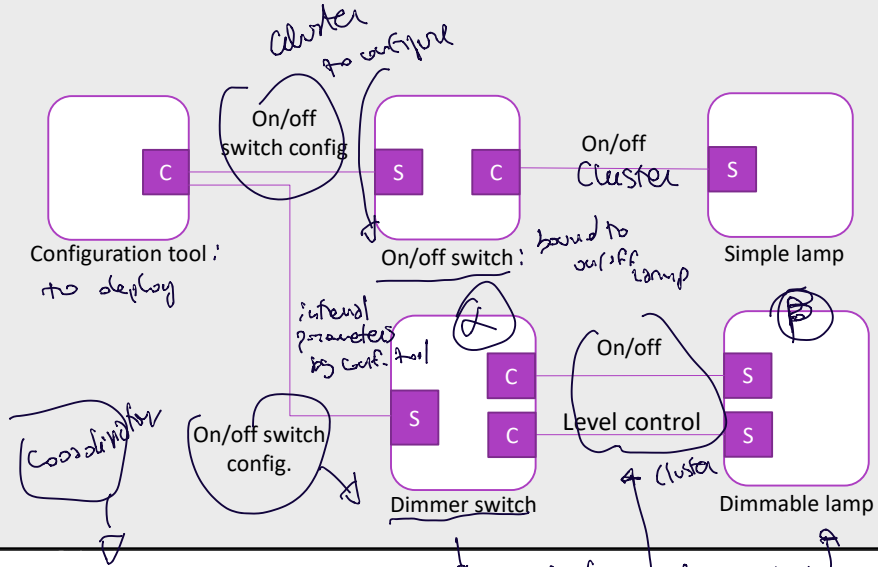


client server

77

application configuration to control some lamps

SCHEMES OF ON/OFF CLUSTERS



78

- set up binding table to bind (A), (B) => need epo to on/off - level control and other others all info to => set up Binding table on coord

=> implement: - device - service - identify } discovery: on all device

- in this case
- 1 dimmable lamp & level control cluster (no need identify)

ATTRIBUTES – BASIC DEVICE INFO.

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	*ZCLVersion*	Unsigned 8-bit integer	0x00 – 0xff	Read only	0x01	M
0x0001	*ApplicationVersion*	Unsigned 8-bit integer	0x00 – 0xff	Read only	0x00	O
0x0002	*StackVersion*	Unsigned 8-bit integer	0x00 – 0xff	Read only	0x00	O
0x0003	*HWVersion*	Unsigned 8-bit integer	0x00 – 0xff	Read only	0x00	O
0x0004	*ManufacturerName*	Character string	0 – 32 bytes	Read only	Empty string	O
0x0005	*ModelIdentifier*	Character string	0 – 32 bytes	Read only	Empty string	O
0x0006	*DateCode*	Character string	0 – 16 bytes	Read only	Empty string	O
0x0007	*PowerSource*	8-bit enumeration	0x00 – 0xff	Read only	0x00	M

79

EXAMPLES OF POWER-SOURCE ATTRIBUTE VALUES

Attribute Value	Description
0x00	Unknown
0x01	Mains (single phase)
0x02	Mains (3 phase)
0x03	Battery
0x04	DC source
0x05	Emergency mains constantly powered
0x06	Emergency mains and transfer switch
0x07 – 0x7f	Reserved

80

TEMPERATURE MEASUREMENT CLUSTER

Attributes:

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	*MeasuredValue*	Signed 16-bit integer	*MinMeasuredValue* to *MaxMeasuredValue*	Read only	0	M
0x0001	*MinMeasuredValue*	Signed 16-bit integer	0x954d – 0x7ffe	Read only	-	M
0x0002	*MaxMeasuredValue*	Signed 16-bit integer	0x954e – 0x7fff	Read only	-	M
0x0003	*Tolerance*	Unsigned 16-bit integer	0x0000 – 0x0800	Read only	-	O

81

TEMPERATURE MEASUREMENT CLUSTER

- Uses the generic commands to:
  - read/write attributes
  - discover attributes
  - configure and report
- There are no specific commands or reporting for this cluster

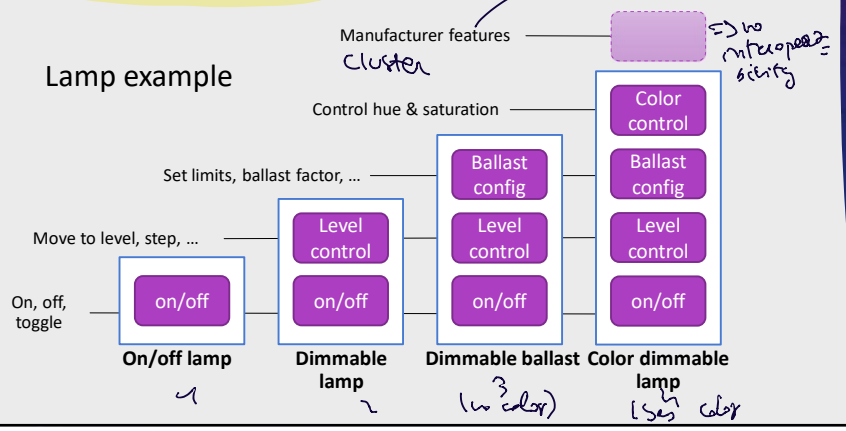
82

COMBINING CLUSTERS

ZCL designed to enforce a hierarchical approach to device functionality

- Support backward compatibility
- Supports interoperability

Lamp example



83

AN EXAMPLE: INDOOR LOCALIZATION IN ZIGBEE

84

INDOOR LOCALIZATION

Can be achieved in many ways

A common method localizes a mobile device by means of the exchange of radio signals

- three components:
  - radio signals transmitter and receiver (HW);
  - measuring unit (HW);
  - localization algorithm (SW)

85

INDOOR LOCALIZATION

Two architectures:

- Remote positioning
  - a mobile device sends radio signals (beacons)
  - a network of devices (anchors) receive the signals and estimate the position
- Self positioning
  - a network of fixed devices (anchors) send radio signals
  - a mobile device receives the signals and estimate its own position

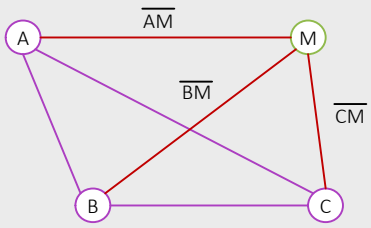
86

INDOOR LOCALIZATION WITH SIGNAL STRENGTH

Example…

The measuring unit may compute the distance of the mobile device from an anchor by measuring the strength of the radio signal

Localization of the mobile device by means of triangulation



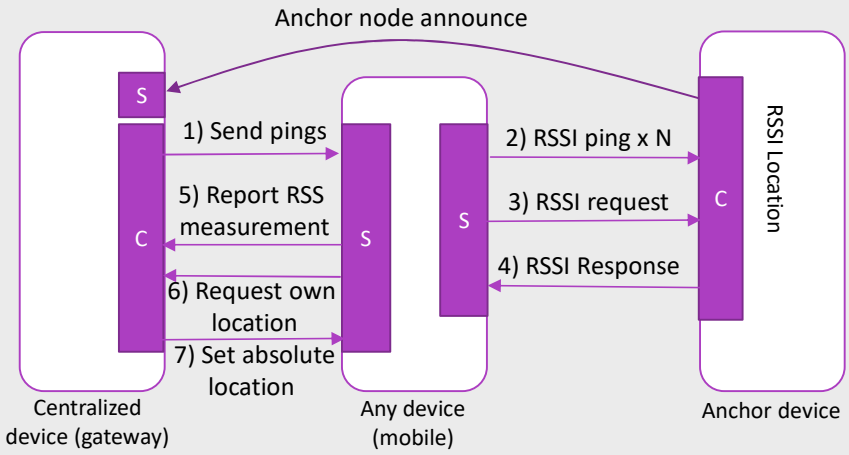
87

RSSI LOCATION CLUSTER

- Exchanges RSSI-based location information
- Exchanges channel parameters among devices
- Reports RSSI data to a centralized device that computes the actual localization information
  - This is optional
- The mobile node (to be localized) is the server
- The anchors are clients

88

USE OF LOCATION CLUSTER WITH CENTRALIZED DEVICE



89

RSSI LOCATION CLUSTER – ATTRIBUTES

Location information attributes:

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	*LocationType*	8-bit Data	0000xxxx	Read / Write	-	M
0x0001	*LocationMethod*	8-bit enumeration	0x00 – 0xff	Read / Write	-	M
0x0002	*LocationAge*	Unsigned 16-bit integer	0x0000 – 0xffff	Read only	-	O
0x0003	*QualityMeasure*	Unsigned 8-bit integer	0x00 – 0x64	Read only	-	O
0x0004	*NumberOfDevices*	Unsigned 8-bit integer	0x00 – 0xff	Read only	-	O

90

RSSI LOCATION CLUSTER – ATTRIBUTES

- Location type can be:
  - Absolute/relative positioning
  - 2D/3D
  - Rectangular coordinate system (or other, reserved)
- Location method can be:
  - Lateration
  - Signposting
    - proximity: coordinates of mobile set to the coordinates of the closest anchor
  - RF fingerprinting
  - Out of band
    - Localization by means of devices out of the ZigBee network
  - Centralized
    - Localization executed by a centralized devices that collects all RSSI data

91

RSSI LOCATION CLUSTER – ATTRIBUTES

Location settings attributes (I):

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0010	*Coordinate1*	Signed 16-bit integer	0x8000 – 0x7fff	Read / Write	-	M
0x0011	*Coordinate2*	Signed 16-bit integer	0x8000 – 0x7fff	Read / Write	-	M
0x0012	*Coordinate3*	Signed 16-bit integer	0x8000 – 0x7fff	Read / Write	-	O
0x0013	*Power*	Signed 16-bit integer	0x8000 – 0x7fff	Read / Write	-	M

92

RSSI LOCATION CLUSTER – ATTRIBUTES

Location settings attributes (II):

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0014	*PathLossExponent*	Unsigned 16-bit integer	0x0000 – 0xffff	Read / Write	-	M
0x0015	*ReportingPeriod*	Unsigned 16-bit integer	0x0000 – 0xffff	Read / Write	-	O
0x0016	*CalculationPeriod*	Unsigned 16-bit integer	0x0000 – 0xffff	Read / Write	-	O
0x0017	*NumberRSSIMeasurements*	Unsigned 8-bit integer	0x01 – 0xff	Read / Write	-	M

93

RSSI LOCATION CLUSTER – COMMANDS

Command Identifier Field Value	Description	Mandatory / Optional
0x00	Set Absolute Location	M
0x01	Set Device Configuration	M
0x02	Get Device Configuration	M
0x03	Get Location Data	M
0x04	RSSI Response	O
0x05	Send Pings	O
0x06	Anchor Node Announce	O
0x07 – 0xff	Reserved	-

94

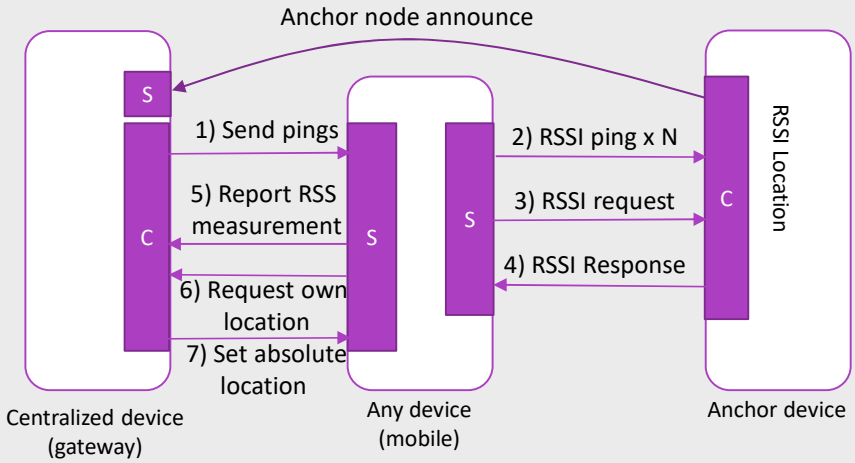
RSSI LOCATION CLUSTER – COMMANDS

Commands generated by the server

Command Identifier Field Value	Description	Mandatory / Optional
0x00	Device configuration response	M
0x01	Location data response	M
0x02	Location data notification	M
0x03	Compact location data notification	M
0x04	RSSI Ping	M
0x05	RSSI Request	O
0x06	Report RSSI Measurements	O
0x07	Request Own Location	O
0x08 – 0xff	Reserved	-

95

USE OF LOCATION CLUSTER WITH CENTRALIZED DEVICE



96

WRAP-UP: BUILDING A ZIGBEE SOLUTION

- Manufacturers of ZigBee-compliant HW platform usually provide the Zigbee libraries and ZDO
- Starting from this, the manufacturer of the ZigBee solution builds its devices, possibly adding the necessary hardware (e.g. transducers), and implementing the rest of the SW & config
  - For each device decide its role in the network (coordinator, router, end-device) and configure it accordingly (either by SW or by tools)
  - For each device: implement the APO.
    - An APO implements the «businness logic» of the device
    - If it is compliant with the ZCL it will also be interoperable with devices of other manufacturers

97

ZIGBEE SECURITY SPECIFICATION

98

ABOUT SECURITY IN ZIGBEE

Security services provided for ZigBee include methods for:

1) key establishment,
2) key transport,
3) frame protection,
4) device management.

These services form the building blocks for implementing security policies within a ZigBee device.

99

ASSUMPTIONS

Level of security depends on:

- the safekeeping of the symmetric keys,
- the protection mechanisms employed,
- the proper implementation of the cryptographic mechanisms and associated security policies involved.

Trust in the security architecture comes from:

- trust in the secure initialization and installation of keying material and
- trust in the secure processing and storage of keying material.

Further assumptions:

- correct implementation of all security protocols
- correct random number generators
- secret keys do not become available outside the device in an unsecured way.
  - The only exception to this is when a device joins the network

100

protection operates at network/device level
=> no individual APO
=> entire device

CAVEATS

due to the low-cost nature of ZigBee devices:

1. one cannot generally assume the availability of tamper resistant hardware.
   Hence, physical access to a device may yield access to secret keying material and other privileged information, as well as access to the security software and hardware.
2. different APO in the same device are not logically separated and lower layers are entirely accessible to the application layer.
   Hence different APO in the same device must trust each other

=> not secure mechanism to physical tamperify of device
(not cover by specification)

101

SECURITY DESIGN CHOICES

- Aims at the protection of individual devices **but not individual applications** in the same device.
  - This allows the re-use of the same keying material among the different layers on the same device
- Keys: for # purposes entire
  - Single key per network (network level security)
  - Single key per link (device-device level security)

=> individual keys to protect dev-dev communications

102

FVFFA

SECURITY DESIGN CHOICES

- decision to use security mechanism
  C, T → implemented to which layer

1. the layer that originates a frame is responsible for initially securing it
   ex. if a NWK command needs protection, NWK security must be used
2. if protection from theft of service is required, NWK layer security to be used for all frames
   (the only exception is when a new device joins)
3. reuse of key materials among different layers in a device, and link-keys to additional security from source to destination
4. an application may use additional security mechanisms, but it's entirely upon itself

APS
ZDO
Keys

103

SECURITY DESIGN CHOICES

APP profile gives you: sec. specification mechanism to convert to a network end pt
keys

Application profiles should include policies to:

- Handle error conditions arising from securing and unsecuring packets. Error conditions may indicate loss of synchronization of security material or may indicate ongoing attacks.
- Detect and handle loss of counter synchronization and counter overflow.
- Detect and handle loss of key synchronization.
- Expire and periodically update keys, if desired.

send.
- detect something was wrong

104

THE KEYS

128-bit **Network** key (shared by all devices):
- acquired either by key-transport or pre-installation
- two types of network keys: standard, and high-security

128-bit **Link** keys (shared by two devices):
- acquired either by key-transport, key-establishment, or pre-installation
- key establishment based on a pre-existing master key
- two types of link keys: global or unique
  - A default global key serves to connect to the network trust center

128-bit **Master** key:
- acquired either by key-transport or pre-installation

Other keys to implement the secure transport of key material
- derived from the link key with one-way functions

105

SECURITY ARCHITECTURE

- The network layer is responsible for the secure transport of its frames
  - Uses the keys provided by the APS layer
  - Provides message encryption, integrity and freshness (to avoid message duplicates).
  - However, some command messages (such as association messages) cannot be encrypted
- The APS is responsible for:
  - assuring secure transport of APS-level frames
  - providing mechanisms for key establishment and transport
  - providing device management services
- the ZDO manages the security policies and the security configuration of a device
  - In particular, it controls the management of cryptographic keys by issuing primitives to the APS layer

106

APS SECURITY SERVICES: KEY ESTABILISHMENT

Key establishment involves two entities, an initiator device and a responder device

To this purpose the APS implements the Symmetric-Key Key Establishment protocol (SKKE, a challenge response protocol):

1. trust provisioning based on trust information (typically the master key)
   - the master key can be either pre-installed during manufacturing,
   - It may be installed by a Trust Center (for example, from the initiator, the responder, or a third-party device acting as a Trust Center),
   - or it may be based on user-entered data (for example: PIN, password, or key).
2. exchange of ephemeral data (a random number)
3. the use of this ephemeral data to derive the link key.
4. confirmation that the link key was computed correctly

107

THE TRUST CENTER

- The Trust Center is the device trusted by devices within a network
  - All members of the network shall recognize exactly one Trust Center
  - There shall be exactly one Trust Center in each secure network.
- In high security applications the Trust Center can be a device can be pre-loaded with the Trust Center address and initial master key
- In alternative the coordinator acts as Trust center
  - … or it delegates another device to this purpose

108

THE TRUST CENTER

- The Trust Center distributes keys for the purpose of network and end-to-end application configuration management.
  - in low-security applications the Trust Center may install the master key in a device by using unsecured transport
- The devices that join the network:
  - obtain the keys from the trust center according to different protocols, depending on which keys they already have (or no key)
  - If they already have the master key, they establish a secure communication link with the Trust Center
    - … and use this secure link to obtain the keys they need

109

SUMMARY

- Structure of ZigBee and relationship with IEEE 802.15.4
- Network layer of ZigBee: association of devices and network formation
- Application sublayer, Application framework and ZigBee Device object
- Endpoints, clusters, profiles and device IDs
- Cluster library
- Security

110