DTU

Networked Embedded Systems

Week 12: Industrial Embedded Networks

Xenofon (Fontas) Fafoutis
Professor
xefa@dtu.dk
www.compute.dtu.dk/~xefa

Requirements of (Industrial) Networked Embedded Systems

- Scalability
  - Supported number of embedded systems
- Throughput
  - Bytes received per second
- Reliability (Packet Delivery Ratio, PDR)
  - Number of packets received over packets sent
- Latency/Delay
  - Time interval from the time a packet is sent to the time packet received
- Jitter
  - Variability of latency/delay
- Time-sensitivity or determinism
  - Predictable/bounded latency and jitter
- Energy Efficiency and Consumption
  - All above need energy

Requirements of (Industrial) Networked Embedded Systems

- Scalability
  - Supported number of embedded systems
- Throughput
  - Bytes received per second
- Reliability (Packet Delivery Ratio, PDR)
  - Number of packets received over packets sent
- Latency/Delay
  - Time interval from the time a packet is sent to the time packet received
- Jitter
  - Variability of latency/delay
- Time-sensitivity or determinism
  - Predictable/bounded latency and jitter
- Energy Efficiency and Consumption
  - All above need energy

Trade-offs: There is no solution that optimises all; we need figure out what we can afford to sacrifice

Industrial Embedded Networks

- Industrial Networks carry vital and often safety critical data
  - Tend to prioritise reliability and determinism over throughput and energy efficiency
  - Best-effort data may co-exist (e.g. vehicle sensor data vs entertainment)

- Common features
  - Schedule-based communication (avoid contention-based protocols)
  - Time synchronisation
  - Traffic prioritisation
  - Redundancy
  - Interference mitigation

- Examples
  - TSCH
  - TSN

Industrial Wireless Networks

- The 6TiSCH protocol stack
- A collection of IEEE/IETF standards
- Aiming at industrial wireless networks that require high reliability and deterministic delay/jitter
- Main components
  - RPL
  - IPv6
  - 6top
  - IEEE 802.15.4 TSCH

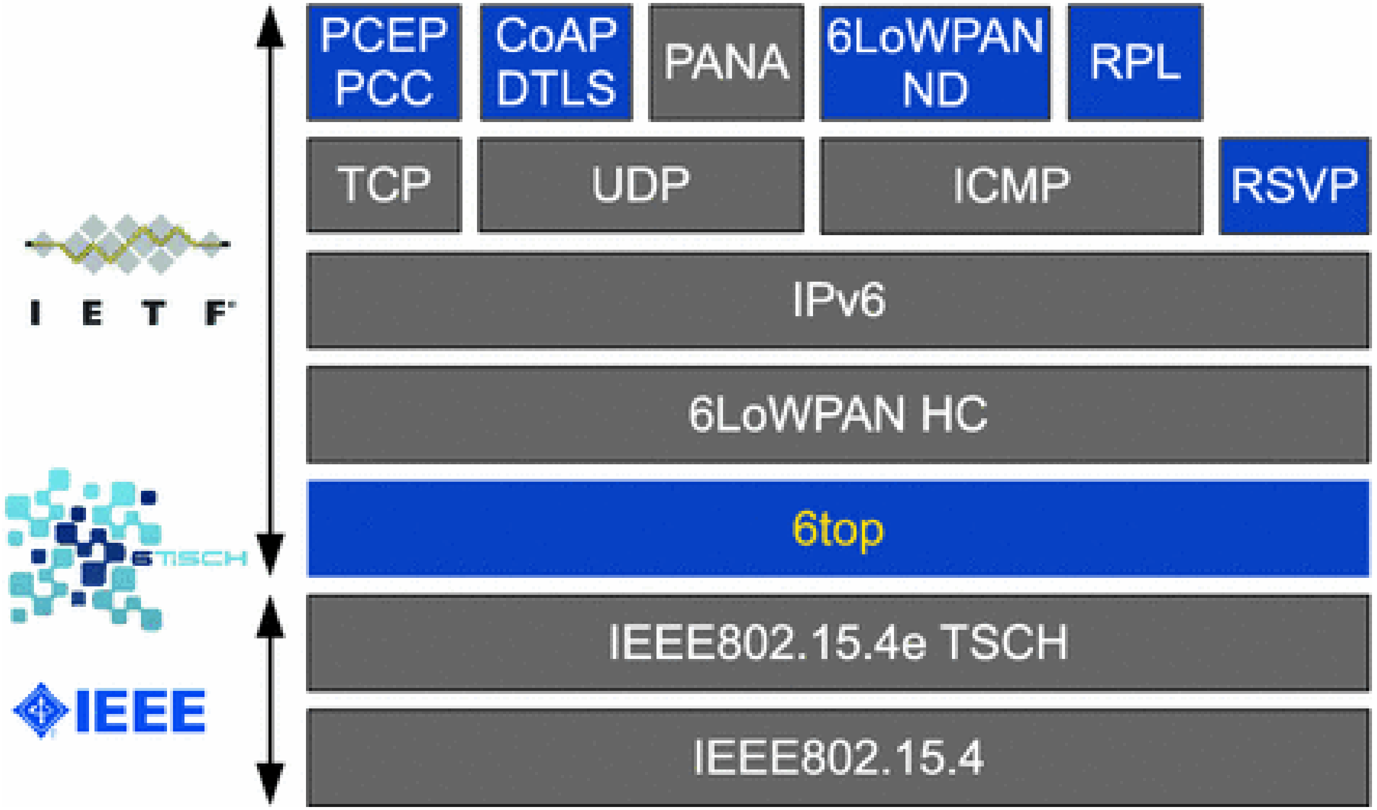


Image source: https://doi.org/10.1007/978-3-319-04223-7_5

IEEE 802.15.4 PHY

- A low-power wireless standard for Low-Rate Wireless Personal Area Networks (LR-WPANs)
  - Supports two ISM Bands (2.4 GHz, sub-GHz)
  - 250 kbps data rate @ 2.4 GHz
  - 127-byte packets
  - 16 channels @ 2.4 GHz
    (+ 1 channel @ 868 MHz in Europe)
  - Supports encryption/authentication

- Used in smart home/city/* wireless networks (Thread, Zigbee)

- Used in industrial wireless networks (6TiSCH)

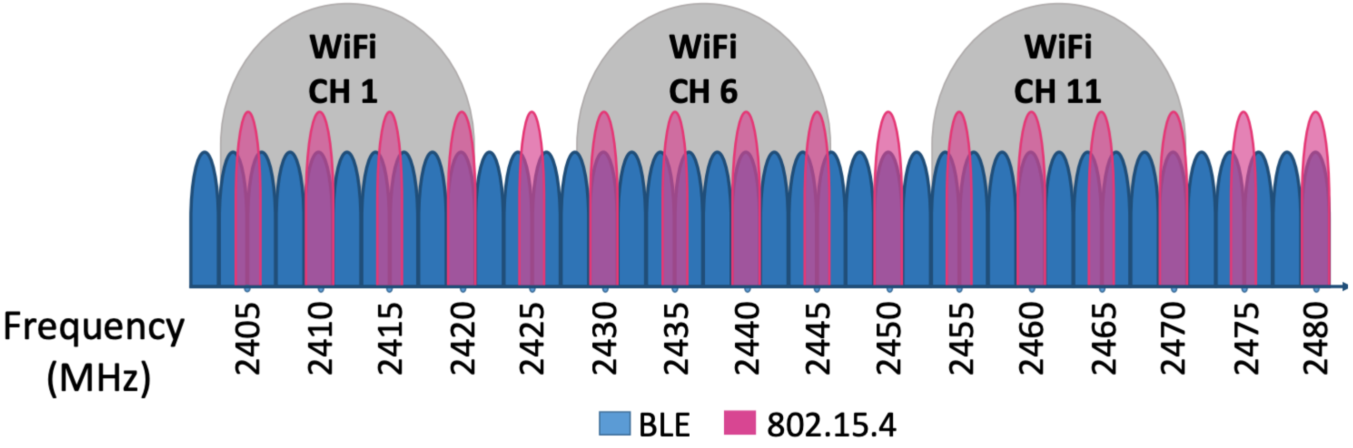


Image source: https://doi.org/10.1109/PIMRC.2017.8292262

IEEE 802.15.4 MAC

- Traditional IEEE 802.15.4-2011 MAC
  - Based on two classes of devices (asymmetric resources)
  - Full-Function Devices (FFD): Coordinator, frame forwarder
  - Reduced-Function Devices (RFD): Simple, energy-preserving
  - Supports contention-based (CSMA/CA) and contention-free (GTS) communication

- IEEE 802.15.4e MAC Enhancements
  - Enhanced MAC protocols for industrial applications
  - DSME (Deterministic and Synchronous Multi-channel Extension)
  - LLDN (Low Latency Deterministic Network)
  - TSCH (Time Slotted Channel Hopping)

IEEE 802.15.4 TSCH

- TSCH (Time-Slotted, Channel Hopping)
  - A MAC protocol based on the IEEE 802.15.4 PHY

- TSCH supports
  - Multi-hop networks (line, tree, mesh) topologies
  - Radio duty cycling in all devices

- TSCH provides
  - Very high reliability (99.99%+ PDR)
  - Interference avoidance
  - Predictable delay/jitter/energy consumption
  - Time synchronisation (10ms)

TSCH Mechanics

- A combination of TDMA and FDMA

- Time-Slotted: Synchronous schedule-based channel access
  - Predictable channel access, possible to eliminate collisions

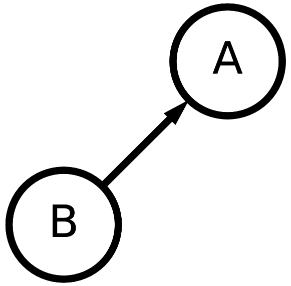
- Channel Hopping: Use of multiple frequency channel in a rotating fashion
  - Resilience to external interference

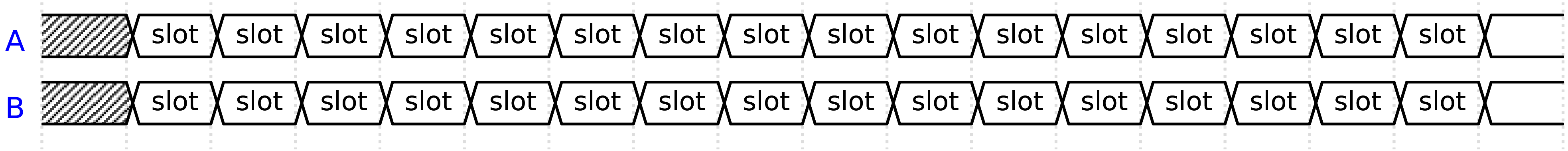
- TSCH traces its roots to older industrial wireless standards
  - Time Synchronized Mesh Protocol (TSMP, 2006) by Dust Networks (proprietary)
  - WirelessHART (2007) by the HART Communication Foundation
  - ISA100.11a (2009) by the International Society of Automation (ISA)

TSCH: Timeslots and Time Synchronisation

- Network-wide Time Synchronisation
  - All devices in a TSCH network are time synchronised with each other
  - Relative time synchronisation based on the implicit synchronisation technique

- Time is split into timeslots
  - Each slot is long enough to support:
    - Transmission of a frame of maximum size (~4 ms)
    - An acknowledgement (~0.35 ms)
    - The guard time (~2 ms)
    - Time required to switch from TX to RX mode, encryption/decryption
  - The duration of the timeslot is not imposed by the standard
    - Typical slot size is 10 ms
    - Some older radios are slow at switching from TX to RX mode and need 15ms

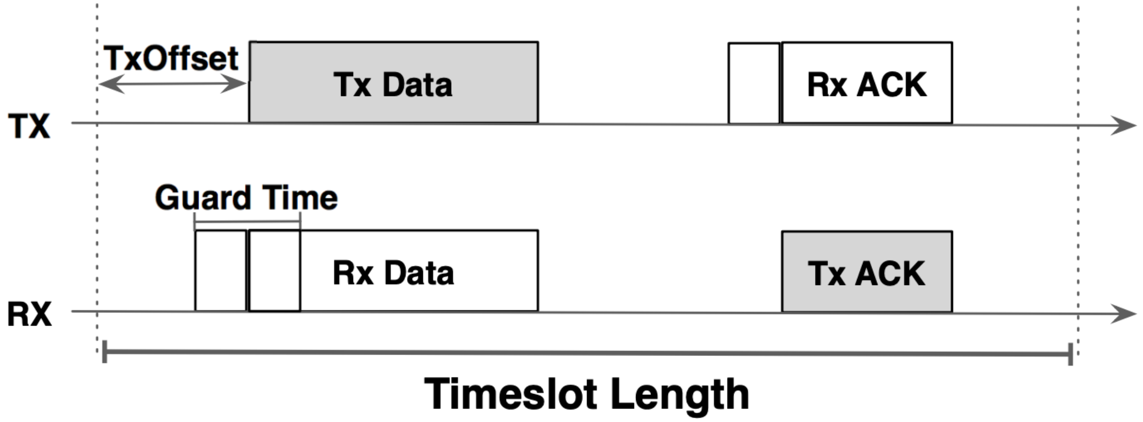



A
B
slot slot slot slot slot slot slot slot slot slot slot slot slot slot slot slot
slot slot slot slot slot slot slot slot slot slot slot slot slot slot slot slot

TSCH: Timeslot

- A timeslot defines a transmission opportunity
  - An actual frame transmission may not happen if there is nothing to transmit!

- The sender initiates the frame transmission at *TxOffset* after the beginning of the timeslot



- The receiver listens to the channel for the time defined by the *Guard Time*
  - The guard time is equally spaced around the expected transmission time to account for both positive and negative drift
  - If a preamble is captured within the guard time, the receiver stays on to capture the full frame
  - Otherwise, the receiver goes to sleep

- After a frame transmission, sender and receiver swap the modes of their radios for the ACK

Image source: https://doi.org/10.1109/WF-IoT.2016.7845475

TSCH: Guard Time

- The guard time allows for a certain degree of synchronisation error
  - Periodic synchronisation is required

- Longer Guard Time
  - More robust to synchronisation errors
  - Less frequent synchronisation required
  - More idle listening, more energy consumption
  - Occupy the channel, bigger slots
  - Fewer slots per second, less data

- Ideally as short as possible, but long enough to maintain robust synchronisation

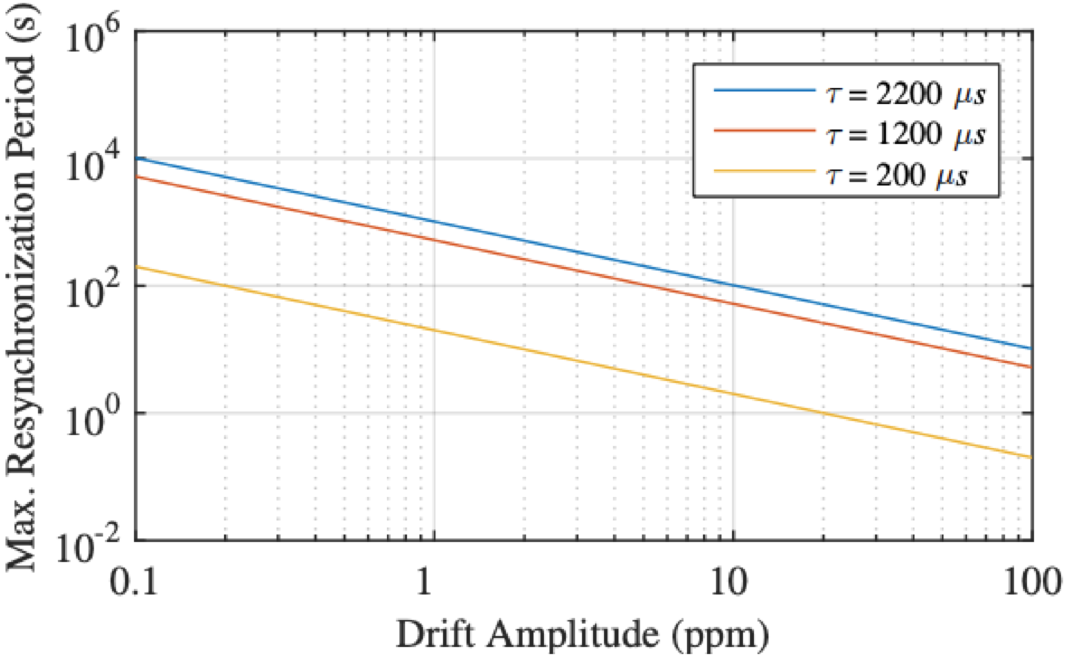
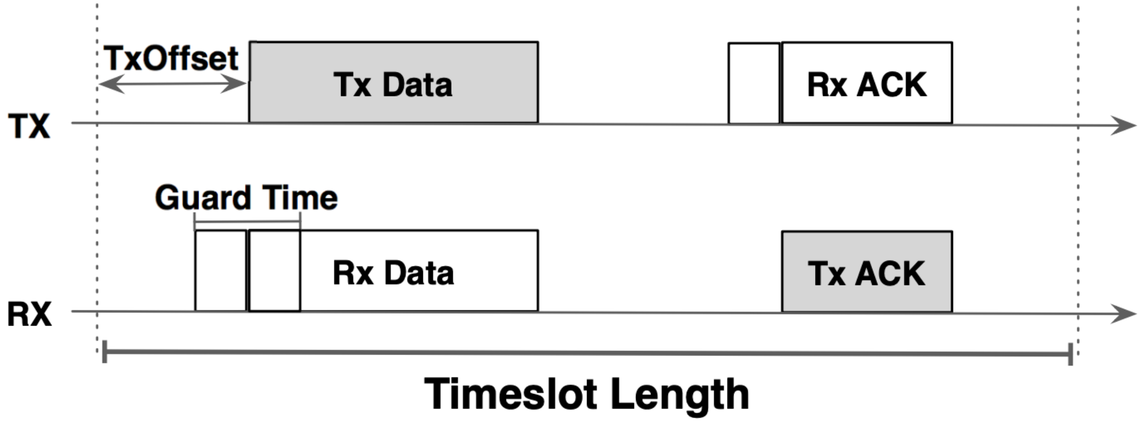


Image source: https://doi.org/10.1109/LCN.2016.042

TSCH: Frame-Based and ACK-Based Synchronisation



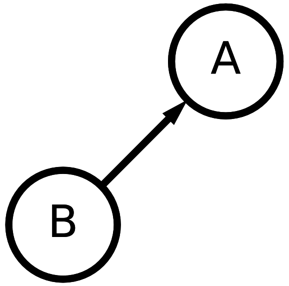
- TSCH uses implicit synchronisation with a neighbour on each frame transmissions
- The receiver calculates the difference between the expected and actual time of frame arrival
- In frame-based synchronisation, the receiver uses this offset to synchronise to the clock of the sender
- In ACK-based synchronisation, the receiver adds this offset in the acknowledgement frame and the sender uses it to synchronise to the clock of the receiver
- In the absence of data, devices need to exchange an empty *Keep-Alive* frame to re-synchronise
  - Significant overhead in applications that transmit very infrequently, e.g. smart metering

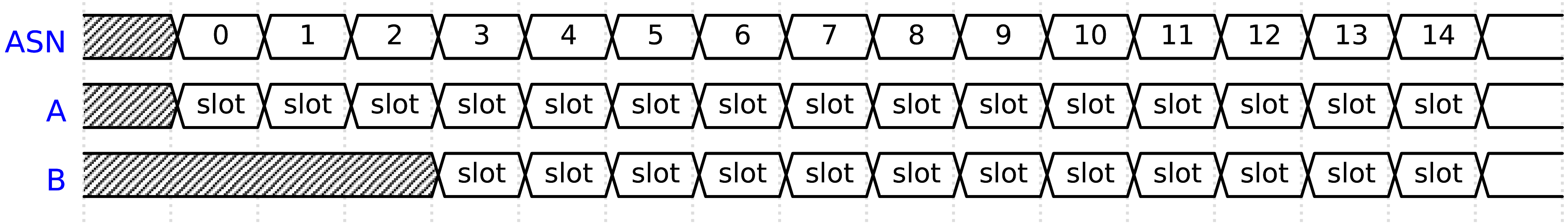
Image source: https://doi.org/10.1109/WF-IoT.2016.7845475

TSCH: Absolute Slot Number

- A number that counts slots since the formation of the network
  - 5 bytes so that it can increment for hundreds of years without wrapping

- ASN Synchronisation
  - ASN is initialised to 0 by the device that forms the network
  - Devices learn the current value of ASN when they join the network
  - Since they are synchronised, all devices know the current value of ASN at any time



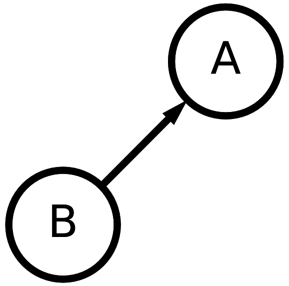


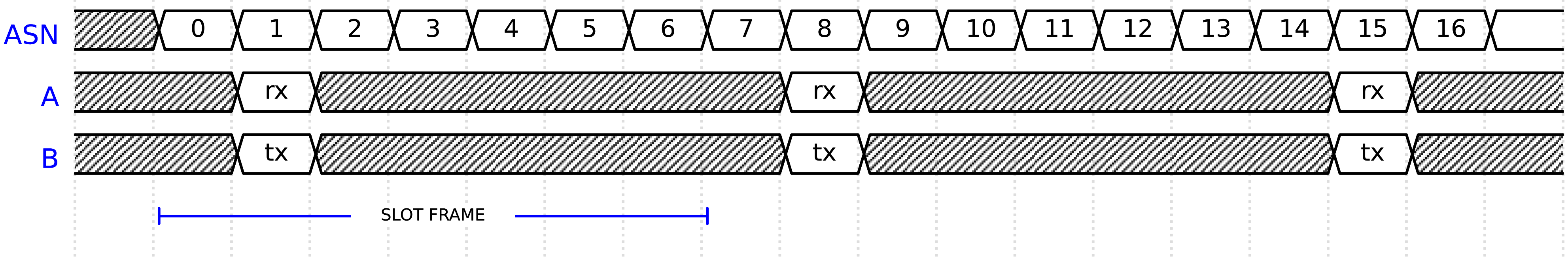
ASN	▨	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A	▨	slot	slot	slot	slot	slot	slot	slot	slot	slot	slot	slot	slot	slot	slot	slot
B	▨	▨	▨	▨	slot	slot	slot	slot	slot	slot	slot	slot	slot	slot	slot	slot

TSCH: Slot frame and Schedule

- Time slots are organised in a slot frame
  - A slot frame is a continuously repeating pattern of slots
  - Slot frame size (not defined by the standard, tailored to the application)
  - [slot offset] = [ASN] % [slot frame size]

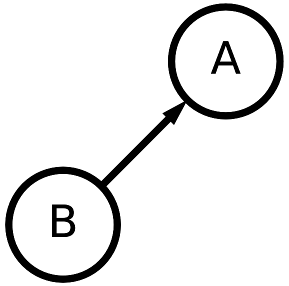
- Communication is based on a schedule, for example:
  - Slot frame size = 7
  - B transits to A in slot offset = 1

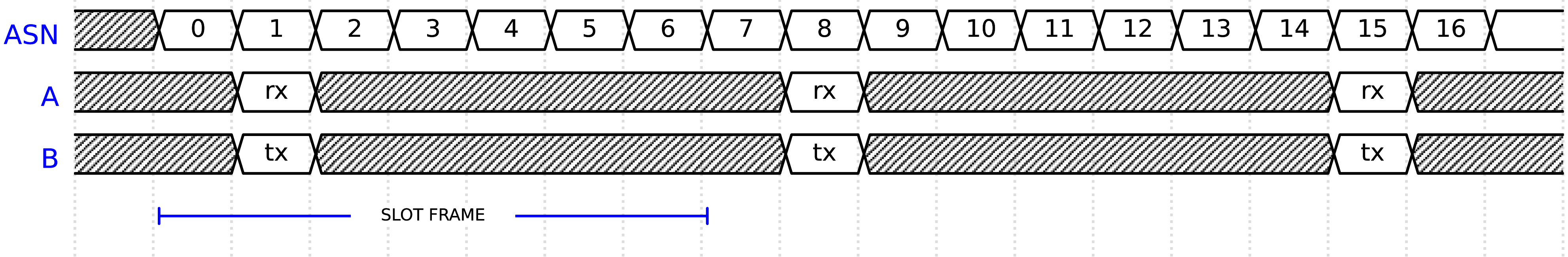




TSCH: The Schedule Controls Several Performance Metrics

- Assuming timeslot size 10 ms and frame size 7…
- Link Capacity: 100 / 7 = 14.28 packets per second
- Energy Consumption: 1 / 7 = 14.28% radio duty cycle
- Latency: up to 7 x 10 ms = 70 ms

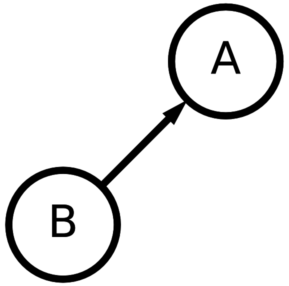


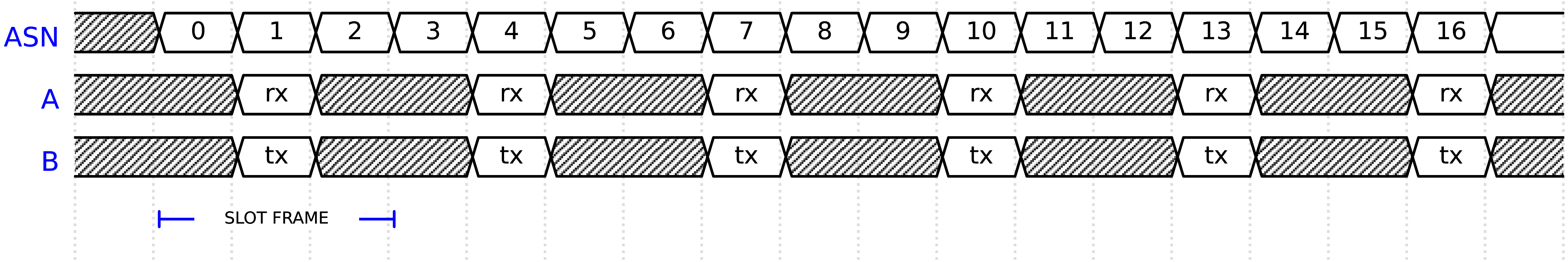


TSCH: Slot Frame Size

- Defines the number of idle slots in a frame

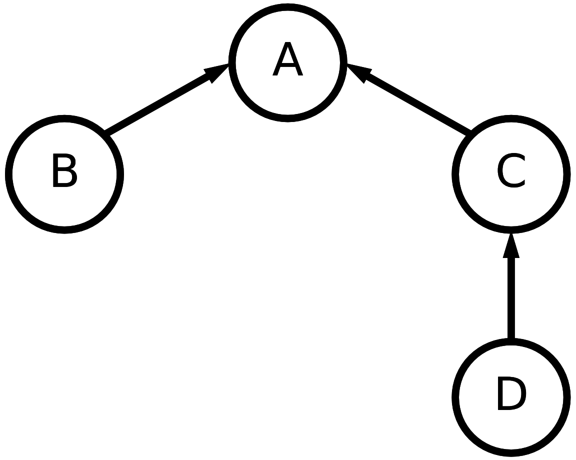
- Shorter slot frame size
  - More transmission opportunities -> higher capacity for data
  - More re-transmission opportunities -> higher reliability
  - Shorter average and worst-case latency
  - More unused slots -> worse energy-efficiency
  - Less room for other devices -> worse scalability

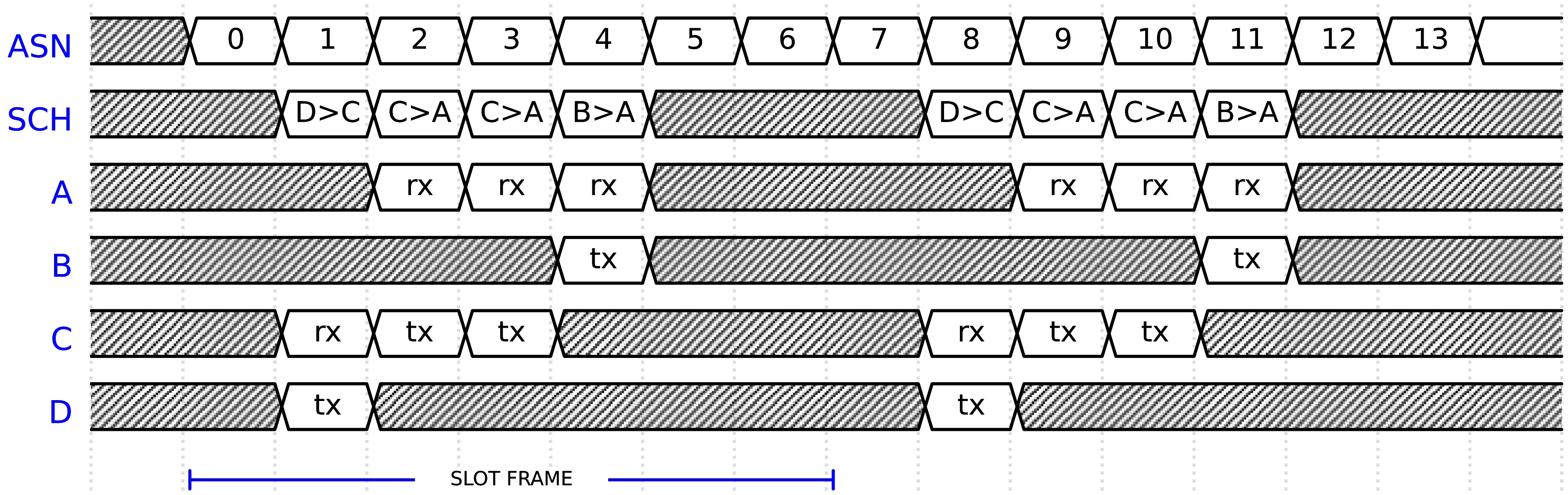




TSCH Schedule: Dedicated Slots

- In each slot, a device can transmit (Tx), receive (Rx), or sleep (idle)
- Dedicated slots (Contention-free)
  - Slot dedicated to a specific link
  - One assigned transmitter -> no collisions
  - Example: B>A, means B transmits (Tx) and A receives (Rx)



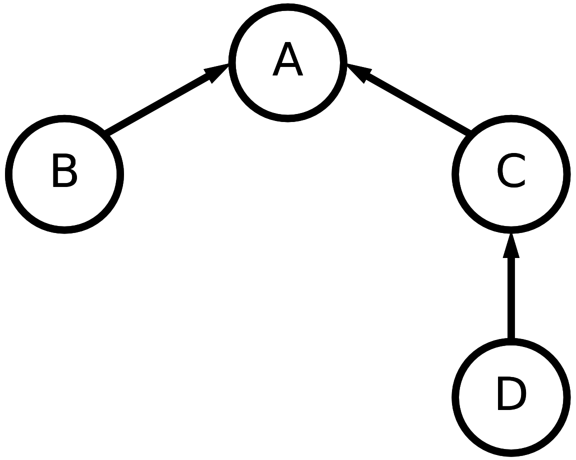


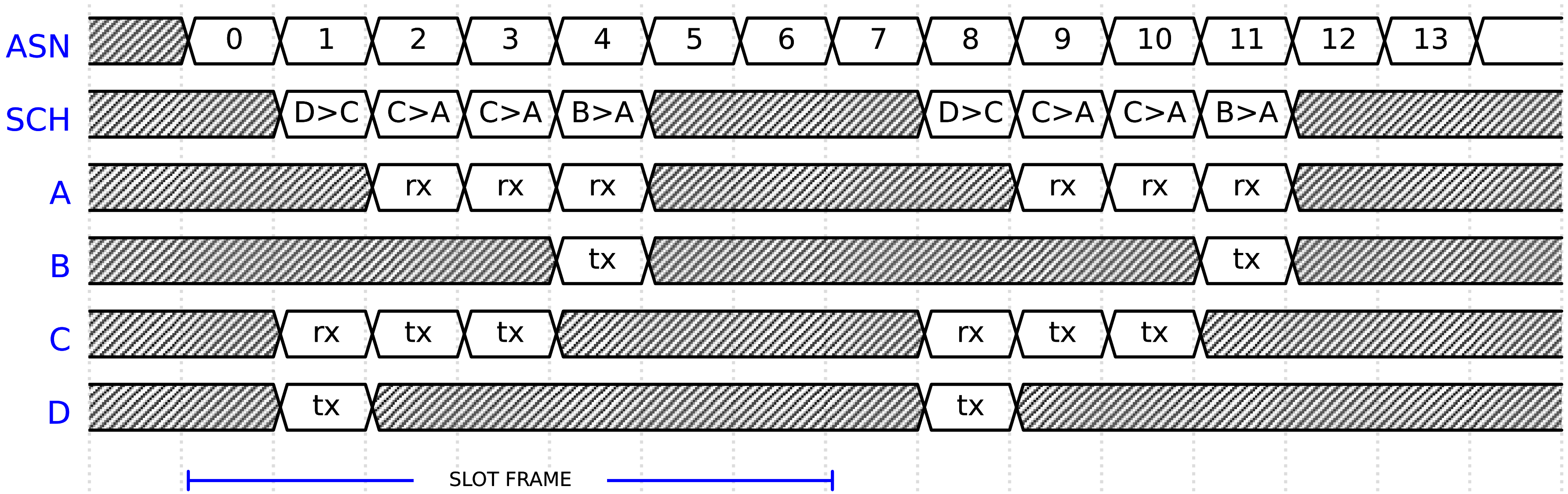
ASN	▨	0	1	2	3	4	5	6	7	8	9	10	11	12	13	
SCH	▨	▨	D>C	C>A	C>A	B>A	▨	▨	▨	D>C	C>A	C>A	B>A	▨	▨	▨
A	▨	▨	▨	rx	rx	rx	▨	▨	▨	▨	rx	rx	rx	▨	▨	▨
B	▨	▨	▨	▨	▨	tx	▨	▨	▨	▨	▨	▨	tx	▨	▨	▨
C	▨	▨	rx	tx	tx	▨	▨	▨	▨	rx	tx	tx	▨	▨	▨	▨
D	▨	▨	tx	▨	▨	▨	▨	▨	▨	tx	▨	▨	▨	▨	▨	▨

SLOT FRAME

TSCH Schedule: How many slots?

- Why did I allocate 2 slots per frame for link C>A?
- How many slots per frame I need generally?

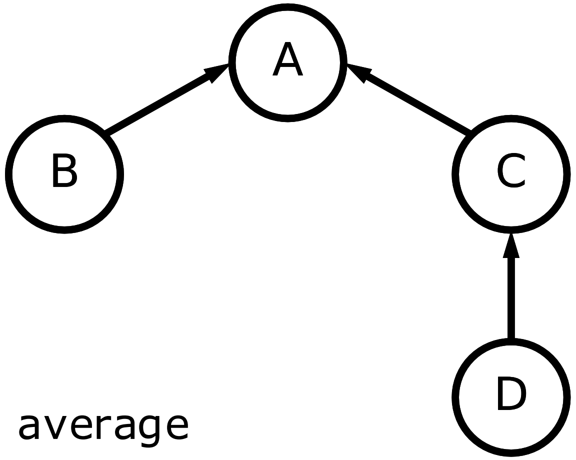


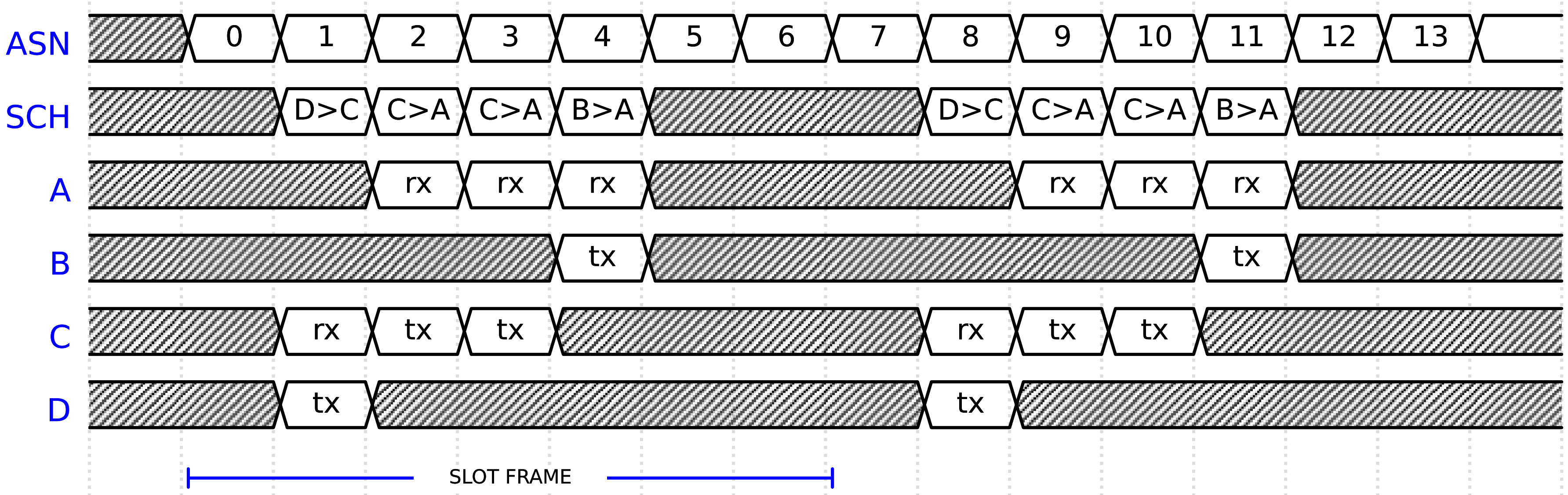


ASN	0	1	2	3	4	5	6	7	8	9	10	11	12	13
SCH		D>C	C>A	C>A	B>A				D>C	C>A	C>A	B>A		
A			rx	rx	rx					rx	rx	rx		
B					tx							tx		
C		rx	tx	tx					rx	tx	tx			
D		tx							tx					

TSCH Schedule: How many slots?

- Slots in the schedule should be able to have capacity for:
  - Packets generation rate by the transmitter
  - Packets forwarding rate by the transmitter
  - Retransmissions (ETX, Expected Transmission Count)
    - ETX = 1/p where p is packet reception probability
    - Example: if p=20%, I have to retransmit each packet 5 times on average

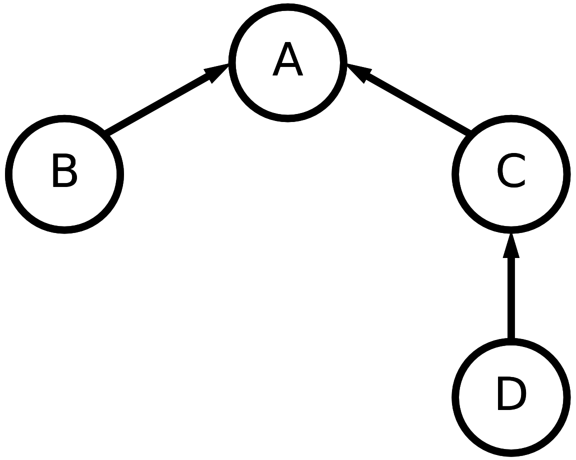


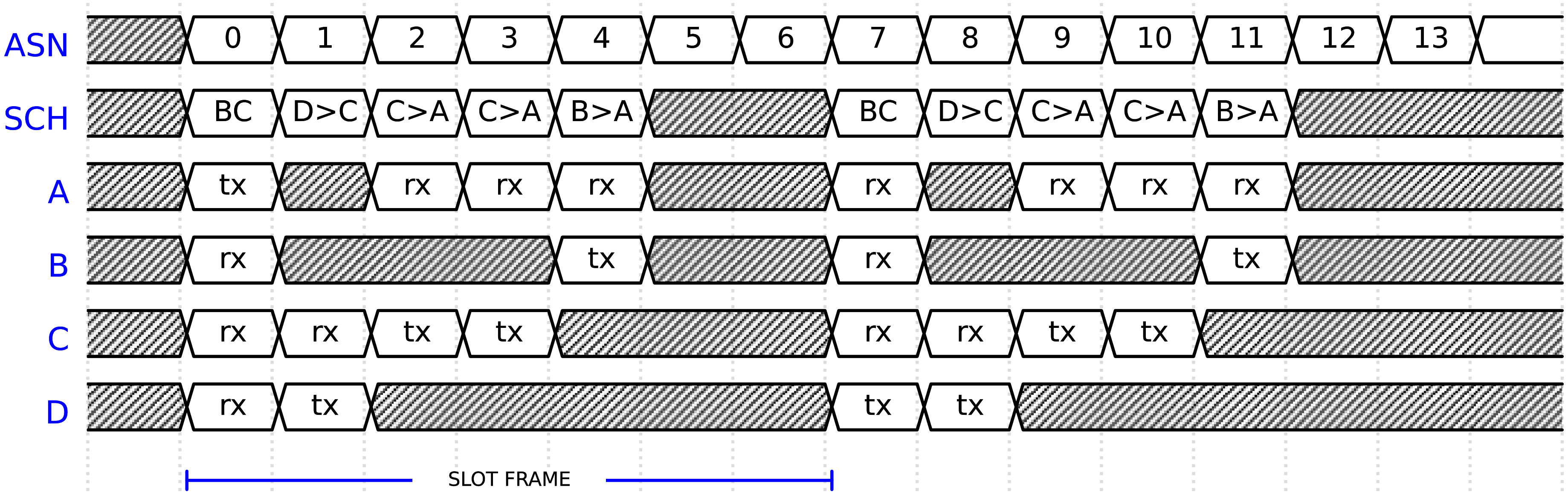


ASN	0	1	2	3	4	5	6	7	8	9	10	11	12	13
SCH		D>C	C>A	C>A	B>A				D>C	C>A	C>A	B>A		
A			rx	rx	rx					rx	rx	rx		
B					tx							tx		
C		rx	tx	tx					rx	tx	tx			
D		tx							tx					

TSCH Schedule: Broadcast Slots

- Broadcast slots (every node active)
  - If I have something to broadcast, I transmit
  - If I have nothing to broadcast, I listen/receive
  - Typically, the first slot of the frame
  - Collisions may happen
  - Transmitters cannot receive broadcasts of others





ASN	▨	0	1	2	3	4	5	6	7	8	9	10	11	12	13	
SCH	▨	BC	D>C	C>A	C>A	B>A	▨	▨	BC	D>C	C>A	C>A	B>A	▨	▨	▨
A	▨	tx	▨	rx	rx	rx	▨	▨	rx	▨	rx	rx	rx	▨	▨	▨
B	▨	rx	▨	▨	▨	tx	▨	▨	rx	▨	▨	▨	tx	▨	▨	▨
C	▨	rx	rx	tx	tx	▨	▨	▨	rx	rx	tx	tx	▨	▨	▨	▨
D	▨	rx	tx	▨	▨	▨	▨	▨	tx	tx	▨	▨	▨	▨	▨	▨

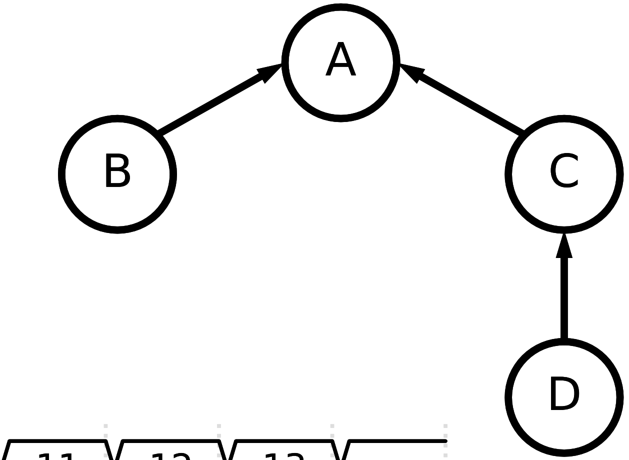
TSCH: Slots Are Limited

- Assuming a time slot of 10 ms, there are only 100 slots per second
  - Permit for maximum 100 packets per second

- What can do if we run out of slots, and we want to support more devices, traffic or retransmissions?

TSCH Schedule: Shared Slots

- Shared slots (Contention-based)
  - Slot assigned to multiple transmitters
  - Collisions are possible, need CSMA/CA

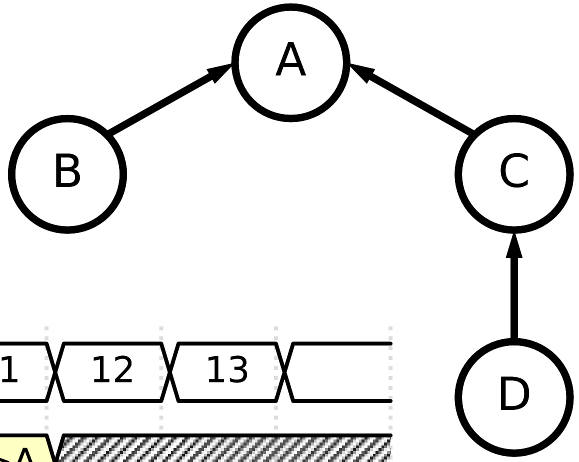


ASN	▨	0	1	2	3	4	5	6	7	8	9	10	11	12	13	
SCH	▨	BC	D>C	▨	▨	*>A	▨	▨	BC	D>C	▨	▨	*>A	▨	▨	▨
A	▨	rx	▨	▨	▨	rx	▨	▨	rx	▨	▨	▨	rx	▨	▨	▨
B	▨	rx	▨	▨	▨	tx	▨	▨	rx	▨	▨	▨	tx	▨	▨	▨
C	▨	rx	rx	▨	▨	tx	▨	▨	rx	rx	▨	▨	tx	▨	▨	▨
D	▨	rx	tx	▨	▨	▨	▨	▨	rx	tx	▨	▨	▨	▨	▨	▨

SLOT FRAME

TSCH Schedule: Multiple Channels

- Parallel transmissions are possible in different channels
  - A transmitter and receiver can only tune to one specific channel



ASN	▨	0	1	2	3	4	5	6	7	8	9	10	11	12	13
SCH CH0	▨	BC	▨	C>A	C>A	B>A	▨	▨	BC	▨	C>A	C>A	B>A	▨	▨
SCH CH1	▨	▨	▨	▨	▨	D>C	▨	▨	▨	▨	▨	▨	D>C	▨	▨
A	▨	rx	▨	rx	rx	rx	▨	▨	rx	▨	rx	rx	rx	▨	▨
B	▨	rx	▨	▨	▨	tx	▨	▨	rx	▨	▨	▨	tx	▨	▨
C	▨	rx	▨	tx	tx	rx	▨	▨	rx	▨	tx	tx	rx	▨	▨
D	▨	rx	▨	▨	▨	tx	▨	▨	rx	▨	▨	▨	tx	▨	▨

SLOT FRAME

Some Channels are better than others

- IEEE 802.15.4 use retransmissions to combat channel errors

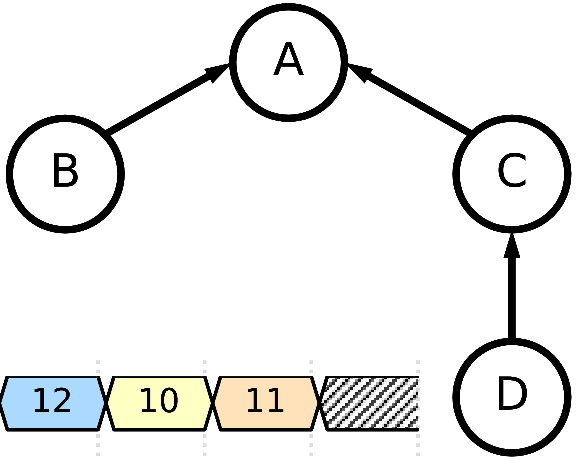
- Interference and fading errors are correlated in time
  - If a channel experiences a lot of interference now, it is very likely to experience a lot of interference in the next frame

- Yet, interference and fading of different channels are often uncorrelated
  - If a channel experiences a lot of interference now, it is very likely that another channel will not be affected by the same source of in the next frame

- Idea: Change channels for each retransmission to enhance reliability

TSCH: Channel Hopping

- Channels change following a pre-agreed pseudo-random order
- All nodes agree on a Channel Hopping Sequence (CHS), e.g. 10-11-12
- Channel = CHS[index], where index = ASN mod len(CHS)



CHS	▨	10	11	12	10	11	12	10	11	12	10	11	12	10	11	12	10	11	▨
ASN	▨	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
SCH	▨	BC	D>C	C>A	C>A	B>A	▨	▨	BC	D>C	C>A	C>A	B>A	▨	▨	BC	D>C	C>A	▨
A	▨	rx	▨	rx	rx	rx	▨	▨	rx	▨	rx	rx	rx	▨	▨	rx	▨	rx	▨
B	▨	rx	▨	▨	▨	tx	▨	▨	rx	▨	▨	▨	tx	▨	▨	rx	▨	▨	▨
C	▨	rx	rx	tx	tx	▨	▨	▨	rx	rx	tx	tx	▨	▨	▨	rx	rx	tx	▨
D	▨	rx	tx	▨	▨	▨	▨	▨	rx	tx	▨	▨	▨	▨	▨	rx	tx	▨	▨

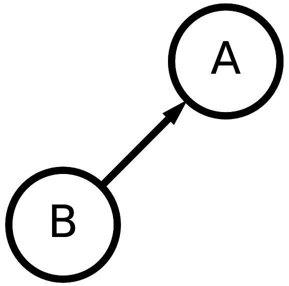
SLOT FRAME

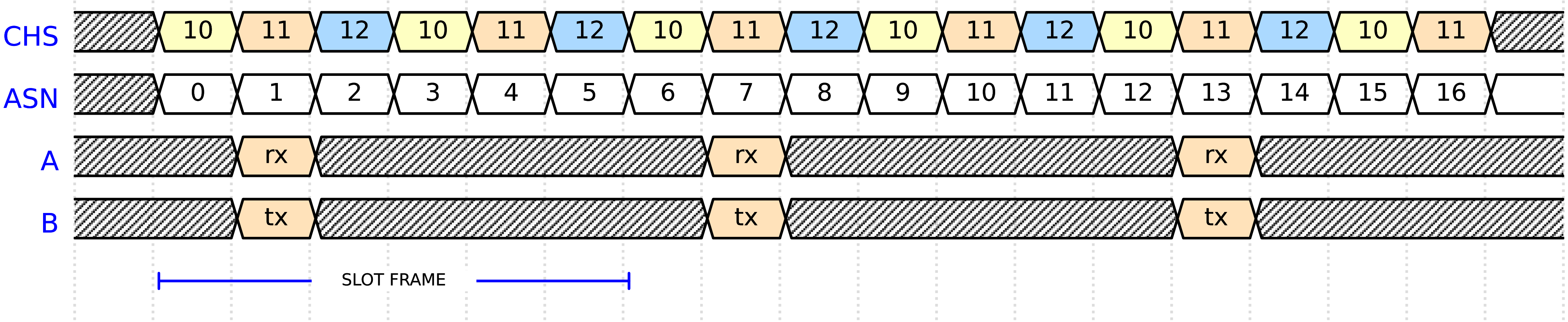
Slot Frame Size vs Channel Hopping Sequence Size

- The Slot Frame Size and CHS size must be relatively prime
  - Their highest common factor should be 1

- Otherwise, there is no channel hopping!

- Example: frame size is 6 and CHS size is 3

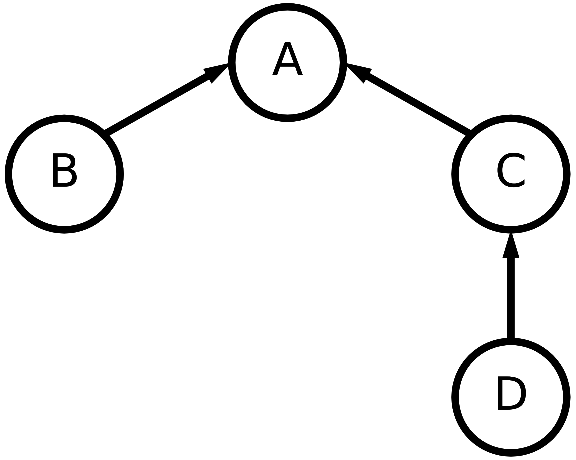




CHS	10	11	12	10	11	12	10	11	12	10	11	12	10	11	12	10	11
ASN	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A		rx						rx						rx			
B		tx						tx						tx			

The TSCH Schedule

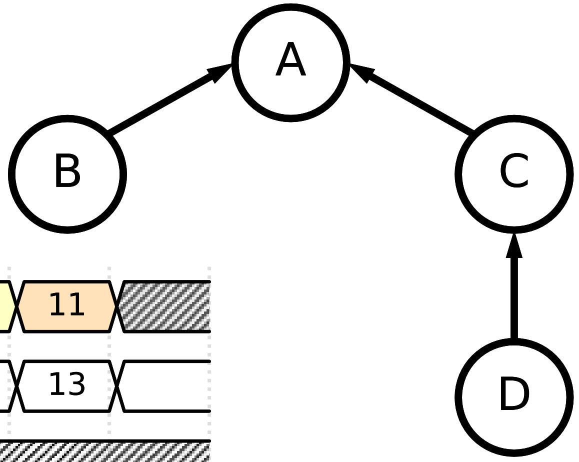
- A N-by-M table where N is the slot frame size and M is the CHS size
  - Each column represents a slot offset (SO)
  - Each row represents a channel offset (CHO)
- Channel = CHS[index], where index = (ASN + CHO) mod len(CHS)

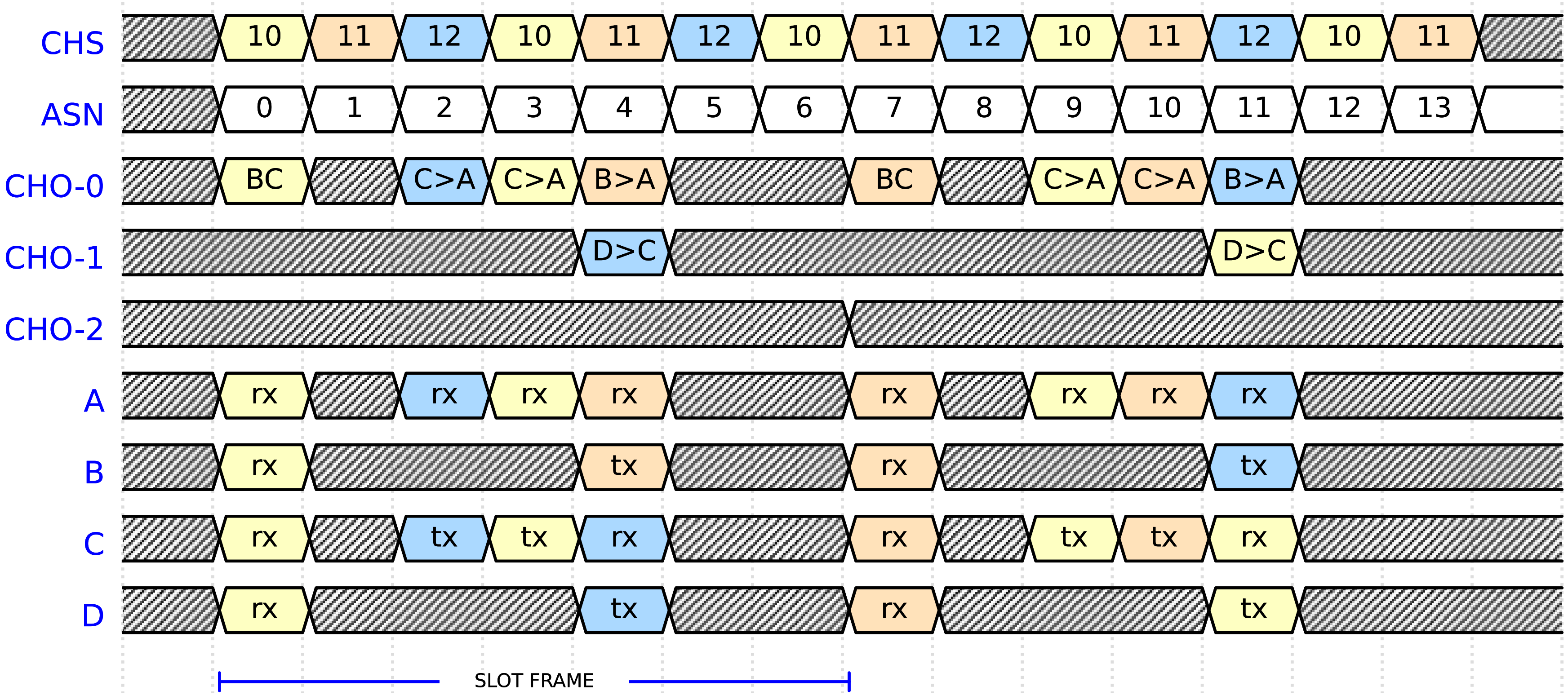


CHO \ SO	0	1	2	3	4	5	6
0	BC		C > A	C > A	B > A		
1					D > C		
2							

TSCH Schedule

- Channel = CHS[index], where index = (ASN + CHO) mod len(CHS)





CHS	▨	10	11	12	10	11	12	10	11	12	10	11	12	10	11	▨
ASN	▨	0	1	2	3	4	5	6	7	8	9	10	11	12	13	
CHO-0	▨	BC	▨	C>A	C>A	B>A	▨	▨	BC	▨	C>A	C>A	B>A	▨	▨	▨
CHO-1	▨	▨	▨	▨	▨	D>C	▨	▨	▨	▨	▨	▨	D>C	▨	▨	▨
CHO-2	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
A	▨	rx	▨	rx	rx	rx	▨	▨	rx	▨	rx	rx	rx	▨	▨	▨
B	▨	rx	▨	▨	▨	tx	▨	▨	rx	▨	▨	▨	tx	▨	▨	▨
C	▨	rx	▨	tx	tx	rx	▨	▨	rx	▨	tx	tx	rx	▨	▨	▨
D	▨	rx	▨	▨	▨	tx	▨	▨	rx	▨	▨	▨	tx	▨	▨	▨

SLOT FRAME

Channel Selection

- Consider a channel with a lot of interference
  - Using as the only single channel is very bad
  - Using it as part of the CHS is better, but inefficient

- How can we select which channels to use?
  - Some channels are more busy than others
  - This quality of the channels changes over time

- Adaptive Channel Selection
  - Measure the quality of channels
  - Update metrics with statistics
  - Select the best channels (blacklist/whitelist)
  - Distribute changes to the channel hopping sequence

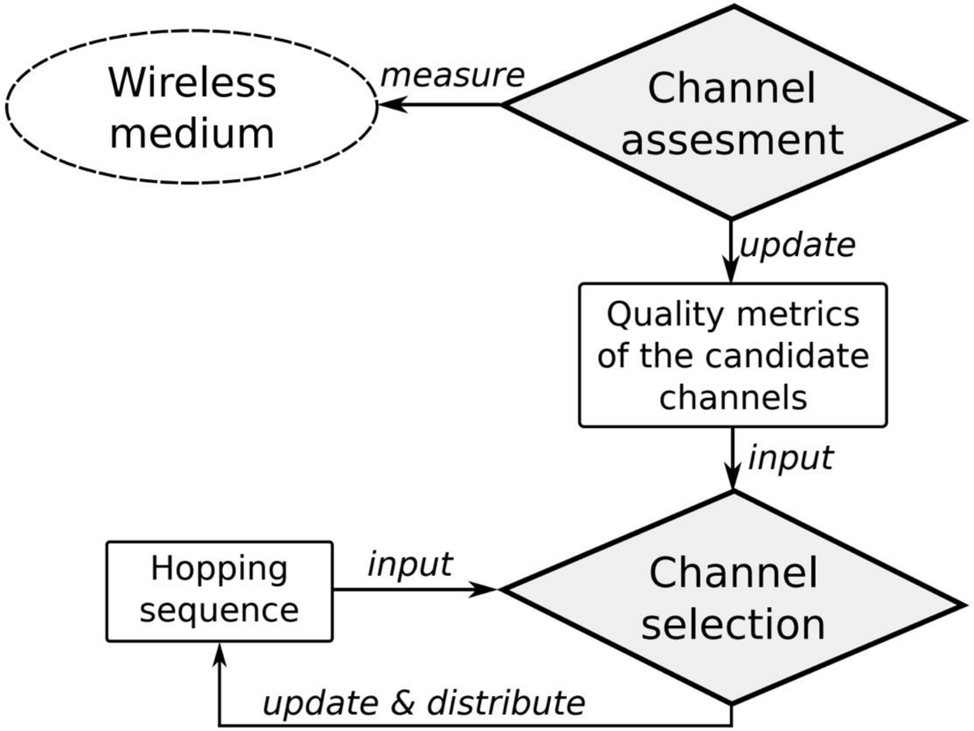
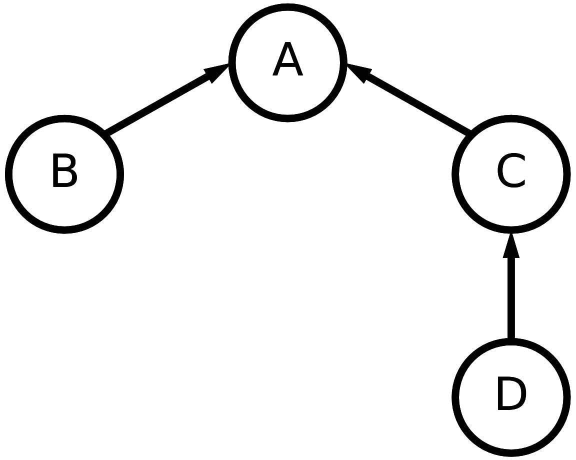


Image source: https://doi.org/10.1109/GIOTS.2017.8016246

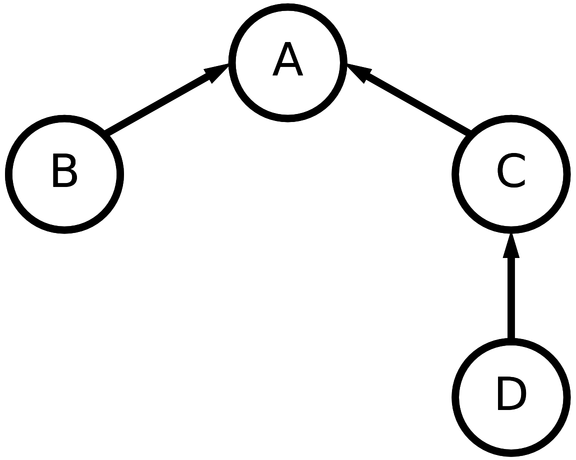
TSCH Scheduler

- A TSCH Scheduler is an algorithm that determines the TSCH schedule
- The Scheduler is not defined by the standard (flexible)
- Schedules are tailored to the use case
- Types of TSCH Scheduling
  - Static Scheduling
  - Dynamic Scheduling
    - Centralised Scheduling
    - Distributed Scheduling
    - Autonomous Scheduling



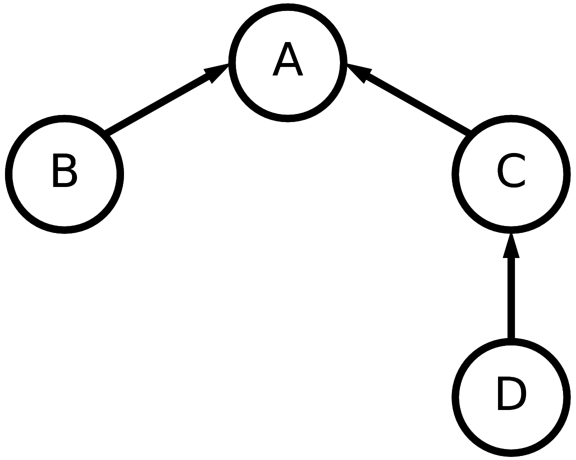
Static Scheduling

- Schedule is hardcoded on the firmware of the devices
- A static schedule can be handcrafted or algorithmically derived
- Makes sense only in relatively static TSCH networks
  - No. of devices, topology, traffic patterns, expected retransmissions



CHO \ SO	0	1	2	3	4	5	6
0	BC		C > A	C > A	B > A		
1					D > C		
2							

Dynamic Scheduling



- A minimal schedule with shared slots is defined
  - Hardcoded or installed when joining the network

- Dedicated slots are allocated dynamically

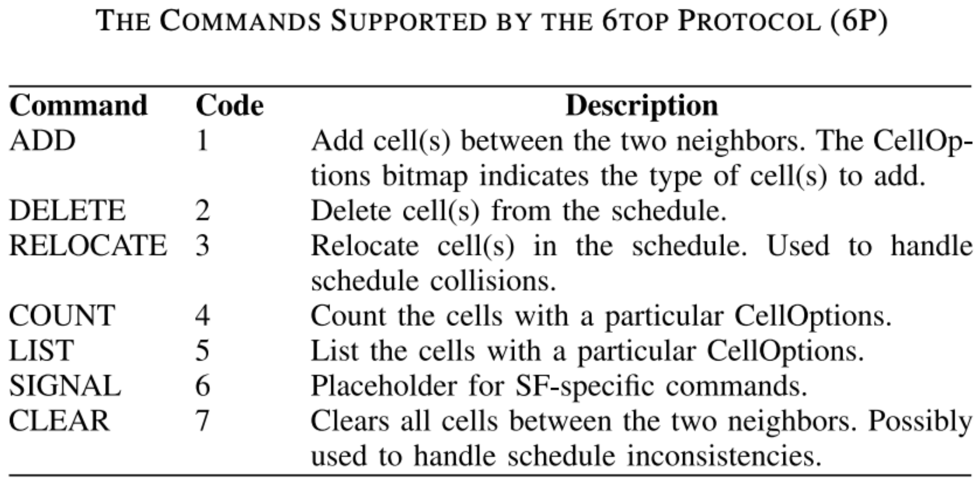
- Centralised Scheduling
  - The root node collects information about the nodes via these shared slots
  - Having the global picture of the network, it allocates dedicated slots to specific links
  - Such decisions can be determined using machine learning or reinforcement learning

- Distributed Scheduling
  - Neighbouring nodes exchange information and agree on which slots to use between them

IETF Minimal Schedule and 6top

- Minimal Schedule
  - One broadcast slot in offset (0,0)
  - The slot frame size and channel sequence size is undetermined
  - Hardcoded or installed when joining the network

CHO \ SO	0	1	2	3	4	5	…
0	BC						
1							
…							



The Commands Supported by the 6top Protocol (6P)

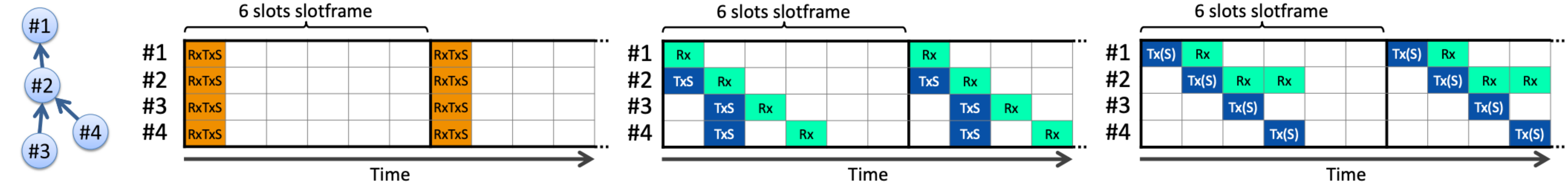
Command	Code	Description
ADD	1	Add cell(s) between the two neighbors. The CellOptions bitmap indicates the type of cell(s) to add.
DELETE	2	Delete cell(s) from the schedule.
RELOCATE	3	Relocate cell(s) in the schedule. Used to handle schedule collisions.
COUNT	4	Count the cells with a particular CellOptions.
LIST	5	List the cells with a particular CellOptions.
SIGNAL	6	Placeholder for SF-specific commands.
CLEAR	7	Clears all cells between the two neighbors. Possibly used to handle schedule inconsistencies.

- 6top (6TiSCH Operation Sublayer Protocol)
  - A protocol to exchange messages for distributed scheduling

Image source: https://doi.org/10.1109/COMST.2019.2939407

Autonomous Scheduling

- Each node derives and changes its schedule autonomously
  - No need to exchange messages
- Shared slots are used for the routing protocol to determine parent nodes
  - Nodes autonomously determine when to transmit/receive using routing information
  - Devices hash the Node ID or MAC address (of sender or receiver) modulo frame size to determine which slot to use
- Orchestra is a widely used autonomous scheduler (albeit not a standard)



(a) Topology

(b) Common Shared Slot

#1	RxTxS						RxTxS			
#2	RxTxS						RxTxS			
#3	RxTxS						RxTxS			
#4	RxTxS						RxTxS			

(c) Receiver-based Shared Slot

#1	Rx						Rx			
#2	TxS	Rx					TxS	Rx		
#3		TxS	Rx					TxS	Rx	
#4		TxS		Rx				TxS		Rx

(d) Sender-based (Shared) Slot

#1	Tx(S)	Rx					Tx(S)	Rx		
#2		Tx(S)	Rx	Rx				Tx(S)	Rx	Rx
#3			Tx(S)						Tx(S)	
#4				Tx(S)						Tx(S)

Image source: https://doi.org/10.1145/2809695.2809714

Joining a TSCH Network

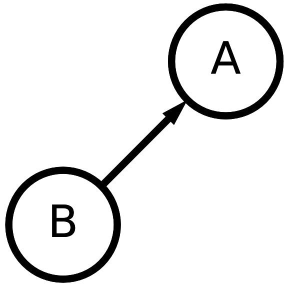
- How does a new device joins a TSCH network?

- A new device that wants to join a TSCH network:
  - Does not know the schedule (unless static scheduling is used)
  - Does not know the current ASN
  - Does not know when the beginning of a frame/slot is (not in sync)

- Enhanced Beacon (EB)
  - Devices of the TSCH network periodically broadcast an EB to advertise the network
  - EB includes the current ASN and (part of) the schedule

- The joining device scans the wireless medium for an EB but without knowing the time and channel of the transmission

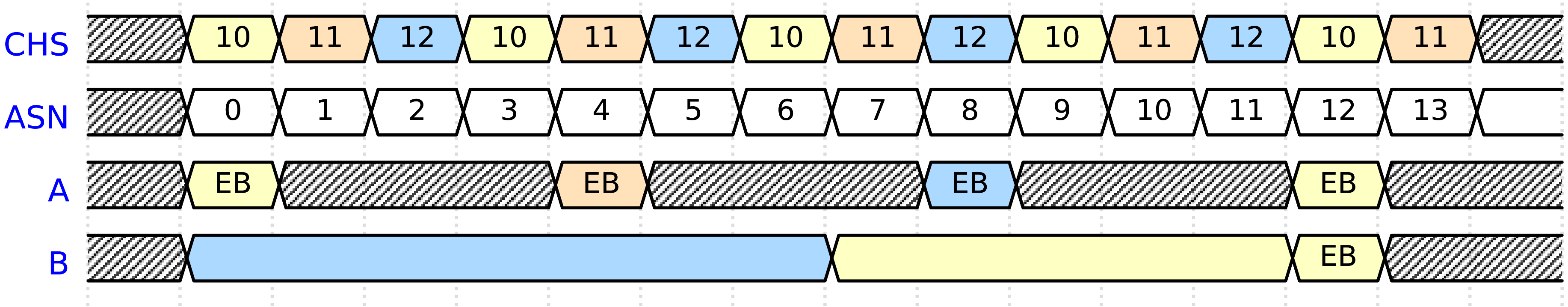
Searching for an EB packet



- The joining node scans sequentially all the channels until it receives an EB
  - If the CHS is fixed, the joining node needs to search only those channels
  - Otherwise, it needs to search all 16 channels

- Upon the reception of the first EB, the joining node keeps scanning for a potentially better parent until a timeout or a maximum number of candidate parents is reached

- The joining process is notoriously slow and inefficient
  - But often considered insignificant in static TSCH networks as it ideally happens just once



CHS	▨	10	11	12	10	11	12	10	11	12	10	11	12	10	11	▨
ASN	▨	0	1	2	3	4	5	6	7	8	9	10	11	12	13	
A	▨	EB	▨	▨	▨	EB	▨	▨	▨	EB	▨	▨	▨	EB	▨	▨
B	▨													EB	▨	▨

Industrial Wired Networks

- IEEE 802.1Q
  - Supports Virtual LANs (VLAN) over Ethernet (IEEE 802.3)
  - VLAN Tagging with 8 priority classes

- Time-Sensitive Networking (TSN)
  - A set of extensions on top of IEEE 802.1Q

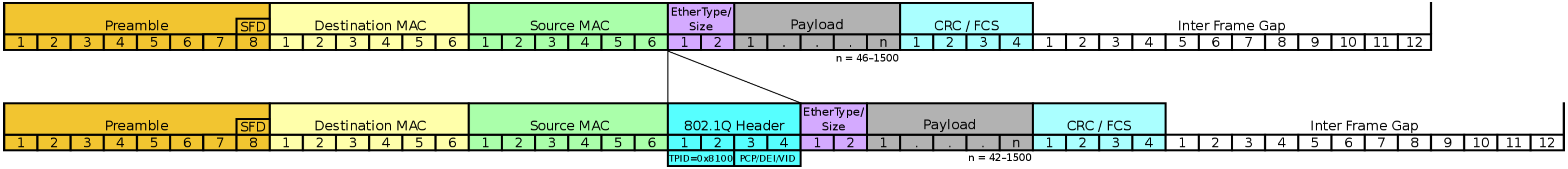


Image source: Wikipedia

Time Sensitive Networking (TSN)

- IEEE 802.1AS Timing and Synchronization for Time-Sensitive Applications
  - Precise time synchronisation based on PTP

- IEEE 802.1Qbv Enhancements to Traffic Scheduling: Time-Aware Shaper (TAS)
  - Synchronised TDMA-based medium access
  - Time is organised in repeating cycles of time slices
  - Time slices can be allocated to specific VLAN priorities for exclusive access
  - Time-critical traffic can be separated from non-critical background traffic

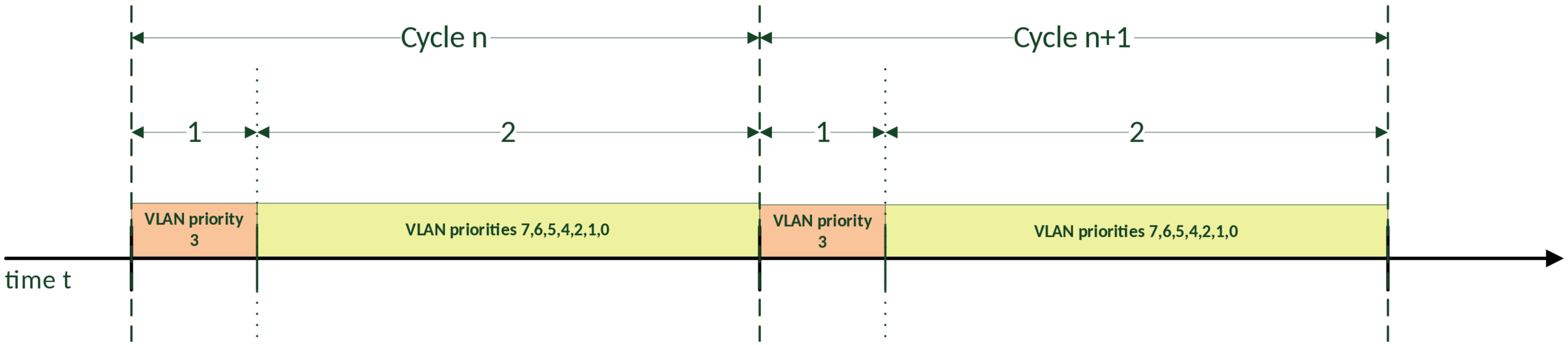


Image source: Wikipedia