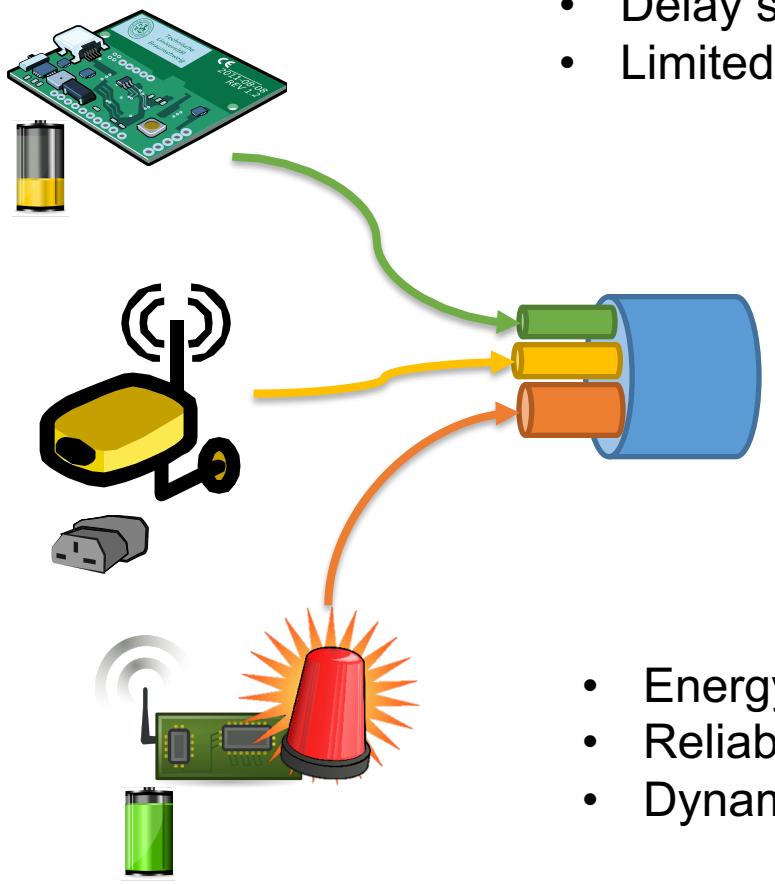
QoS for Things: State of the Art and Challenges

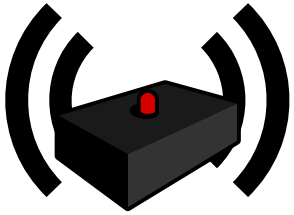
Giacomo Tanganelli
Dip. Ingegneria dell'Informazione
University of Pisa

g.tanganelli@iet.unipi.it

QoS in IoT

- Limited bandwidth
- Delay sensitive flows
- Limited computation capabilities





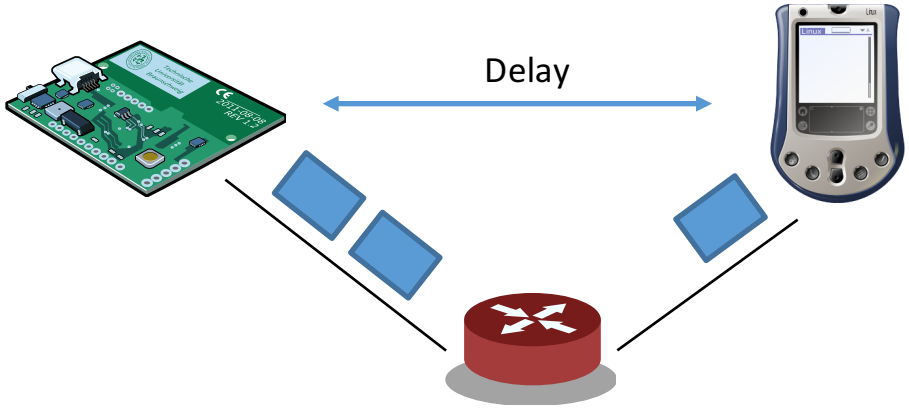
QoS in IoT differs from QoS in traditional networks

- Energy consumption
- Reliability issues
- Dynamic environment

QoS requirements

- Bandwidth
- Peak Rate

- End-to-end delay
- Jitter

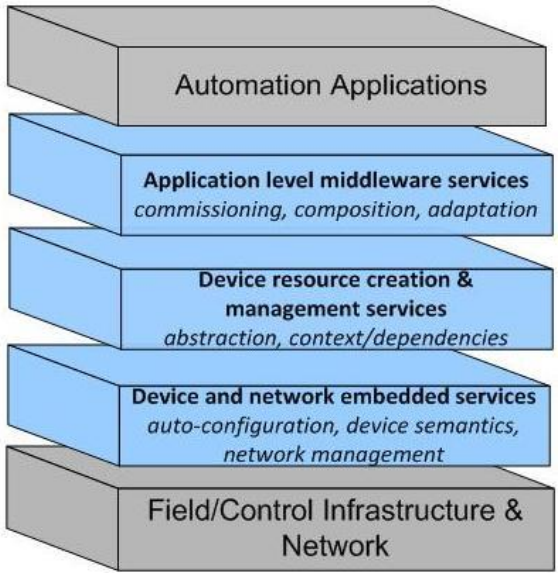


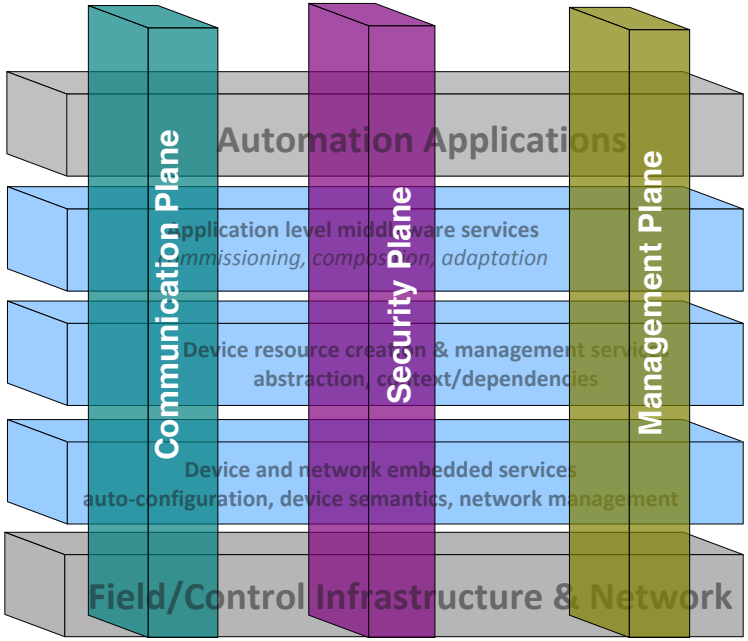
State of the Art

- Few IoT protocols targets QoS
- Many general purpose protocols with QoS by design
  - Use them as examples

IoT@Work

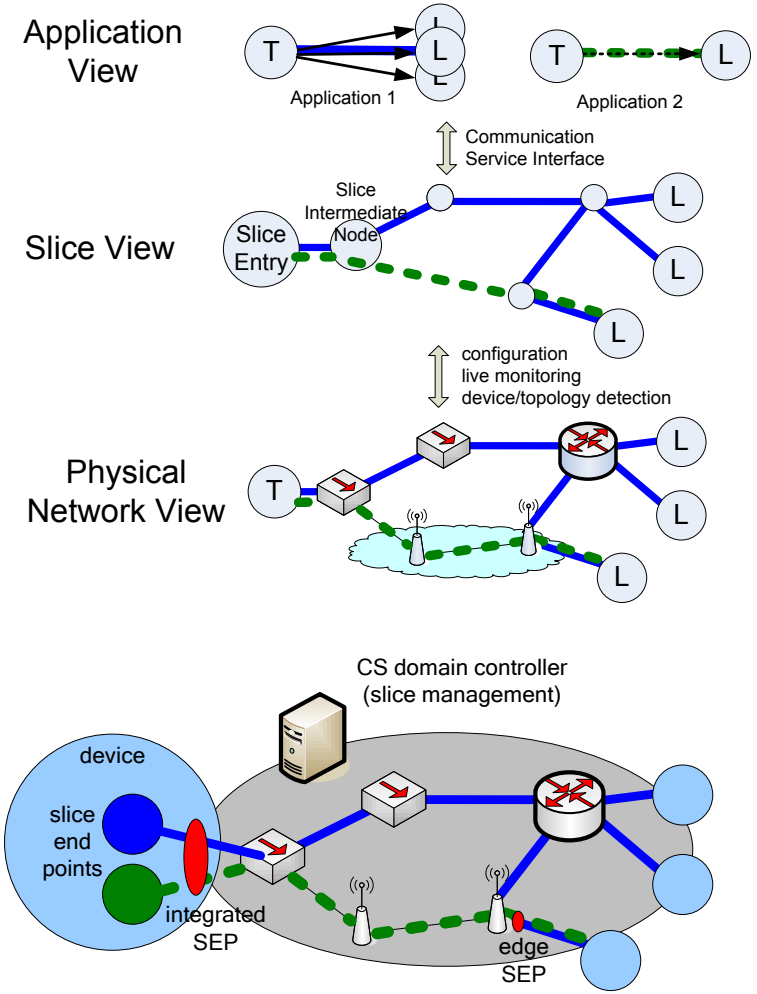
- FP7 Project: https://www.iot-at-work.eu/
- Publish/Subscribe based on AMQP
  - Relies on TCP





IoT@Work

- QoS implemented through Layer 2 and Layer 3:
  - MPLS/VLAN
  - Centralized approach
- Lack of implementations

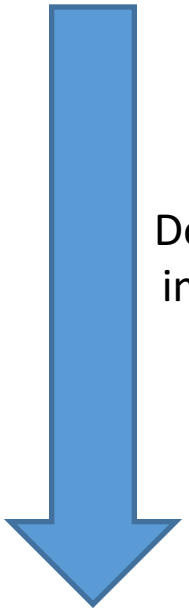


WirelessHART

- Industrial standard, extension of the HART (cabled) standard
- TDMA, strong synchronization
- Centralized coordination
  - Routes forwarded to devices
  - Graph Routing
- Reliable/Unreliable messages

WirelessHART

- QoS Policies:
  - Command: devices must accept message
  - Process-data: devices accept message if above threshold
  - Normal: devices accept message if above threshold
  - Alarm: devices accept message if buffer space
- Route planning for time guarantees



Bluetooth

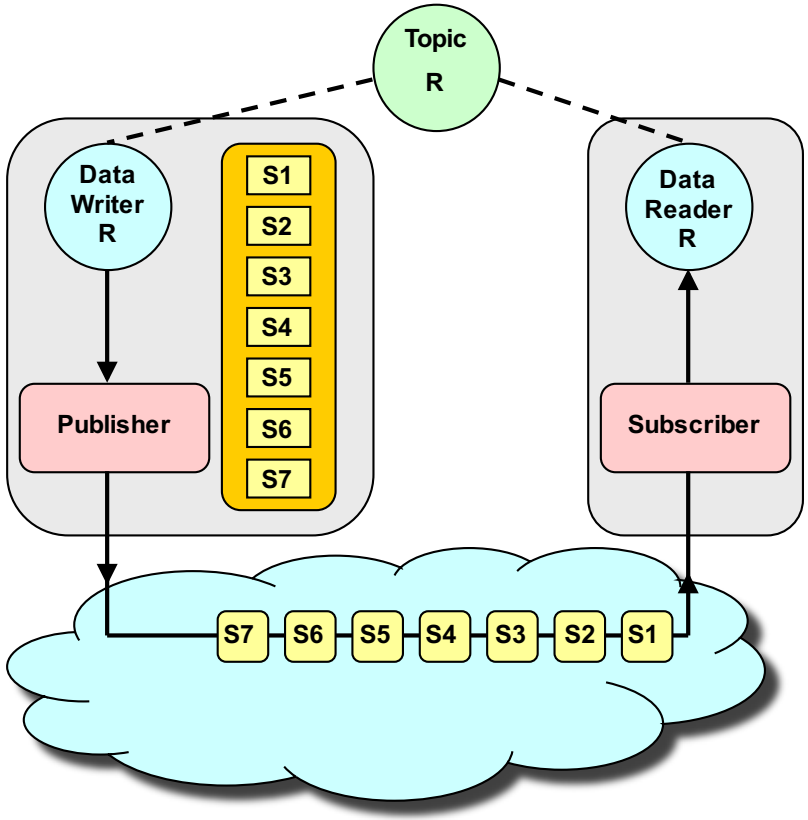
- Master based
  - Small networks (max 7 devices)
  - Polling mechanism
- Time Division Duplex
  - Time is strictly slotted

Bluetooth

- Negotiate ACL to achieve QoS
  - Token Rate
  - Token Bucket Size
  - Peak Bandwidth
  - Latency
  - Delay Variation (jitter)
  - Flush Time Out (timeout for retransmit)
  - Tpoll (time interval for slave transmissions)

Data Distribution Service (DDS)

- Publish/Subscribe based on topics
  - Relies on UDP
  - Reliable/Unreliable
- Publisher announces topic enriched with QoS capabilities
- Subscriber matches topics with required QoS guarantees

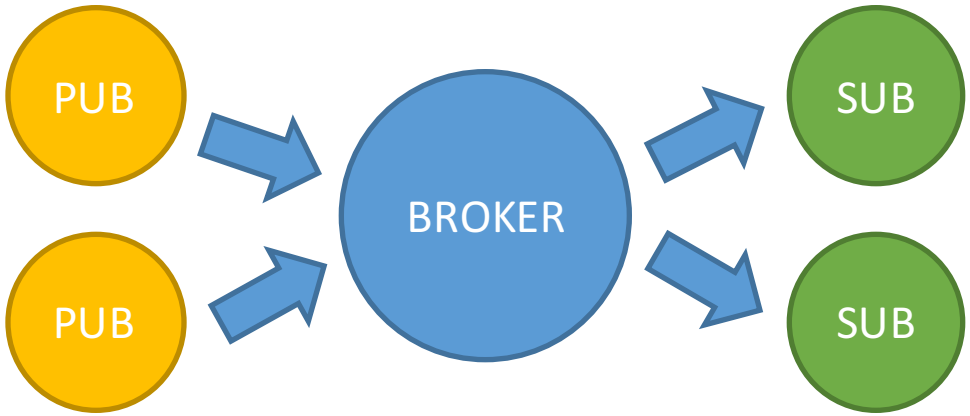


Data Distribution Service (DDS)

- QoS policies:
  - Deadline: maximum inter arrival time (periodic stream)
  - Time based filter: minimum inter arrival time (no flood of data)
  - Latency budget: delay between write on outgoing and receive on incoming (hint)
  - Transport Priority: set priority of underlying transport layer (hint)
- Does not consider lower layer issues

MQTT

- Messaging transport protocol following Publish/Subscribe paradigm
  - Relies on TCP
  - Agnostic to message payload
  - Broker based

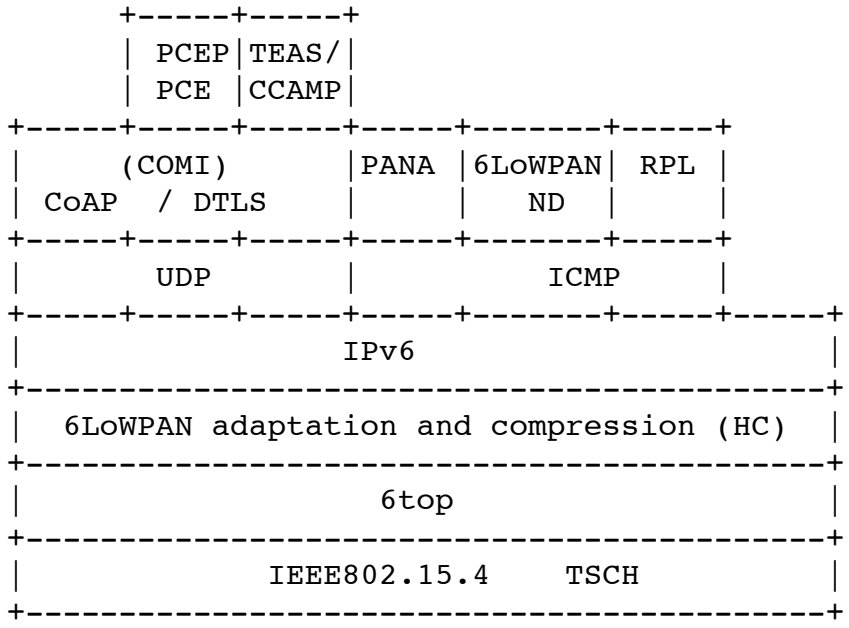


MQTT

- QoS policies (delivery assurances):
  - At most once: best-effort
  - At least once: Application ACK
  - Exactly once: 2-way Application ACK
- Lack of QoS time guarantees

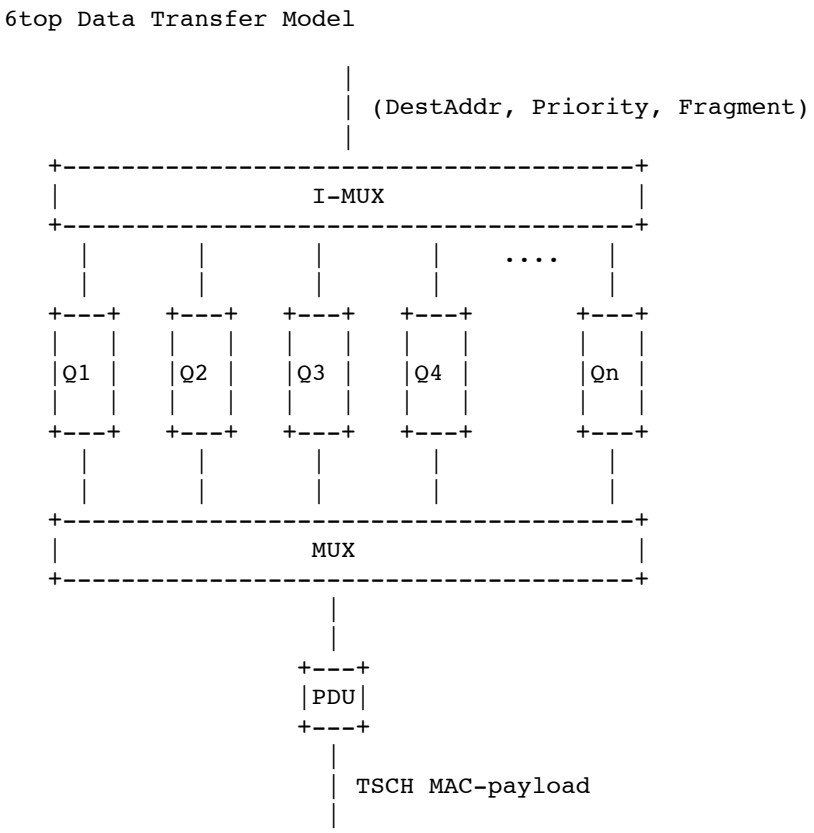
6TiSCH

- 6top
  - LLC between 6LoWPAN and MAC
- Slotted
  - Hard cell
  - Soft cell
- Routing:
  - Centralized (PCE)
  - Distributed (RPL)



6TiSCH

- Negotiate Soft Cells
- QoS policies
  - Multiple outgoing queues
  - DiffServ from upper layer
    - Deterministic forwarding
    - DiffServ for LLN



QoS Modeling

- Deterministic — Static networks
  - Deterministic end-to-end assurances – IntServ like
  - Not possible over contention-based
- Proportional — Dynamic networks
  - Service differentiation - DiffServ like
  - Qualities of flows with respect to each other stay constant
- Statistical
  - Probability to meet QoS requirements

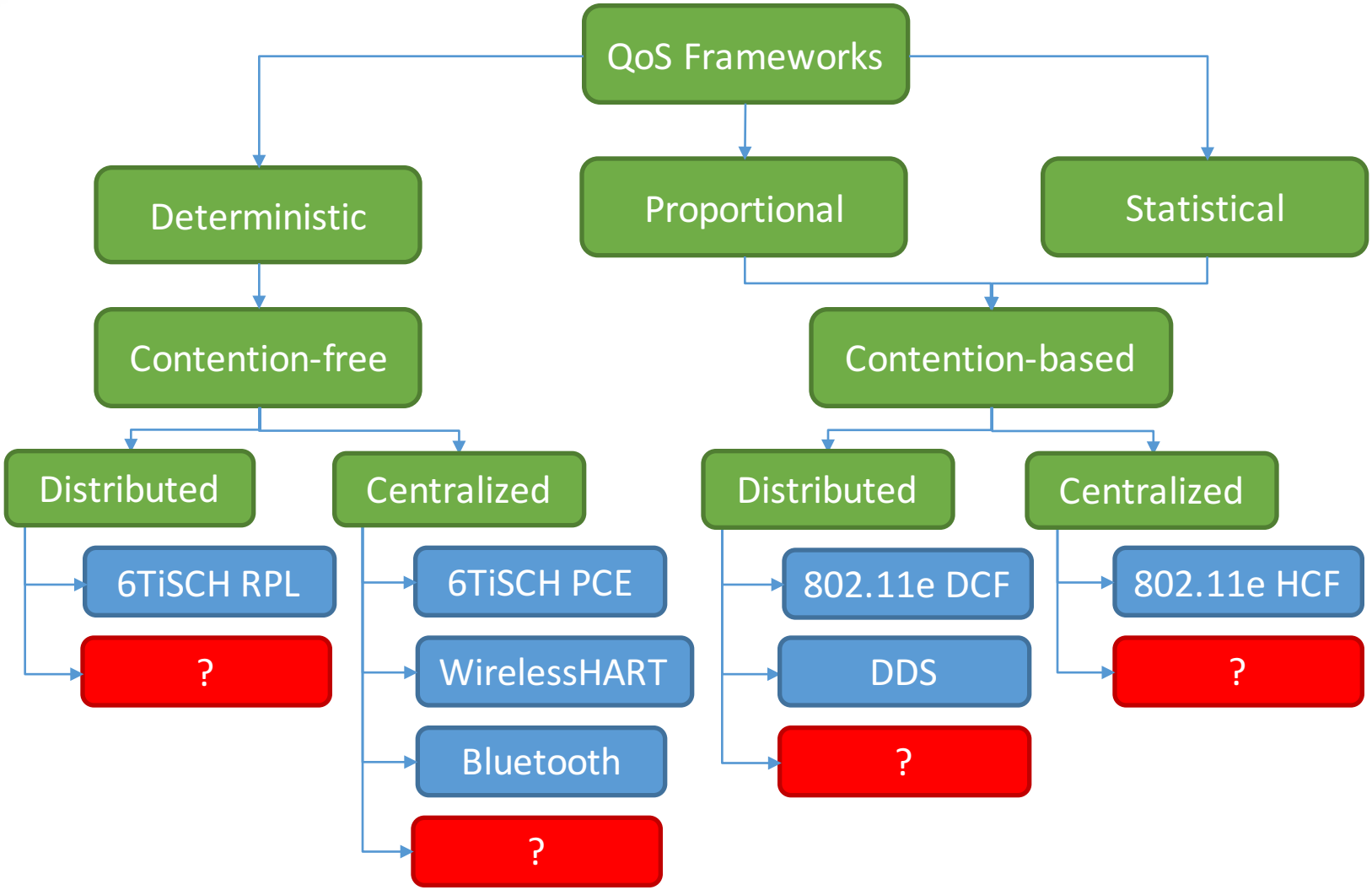
Resource Allocation/Bandwidth Manager

- A Must for deterministic QoS
- Hard in multi-hop distributed networks
  - Exchange of needed information through routing packets
- Centralized approach
  - Scalability issues
  - Involved during flow establishment, teardown, change traffic patter etc.

End-to-End Delay

- May rely on bandwidth management
- Adaptation of Application's requirements
- Cross-layer design
- Applications can be:
  - Aware of QoS
    - Directly negotiate requirements
  - Unaware of QoS
    - Framework infers Application's requirements

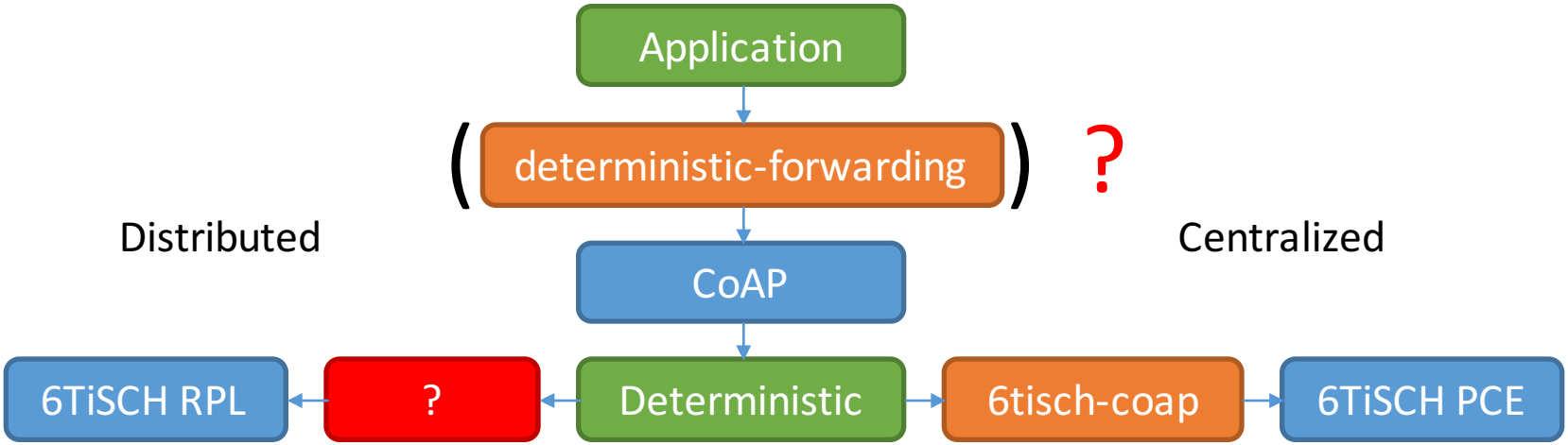
QoS Frameworks



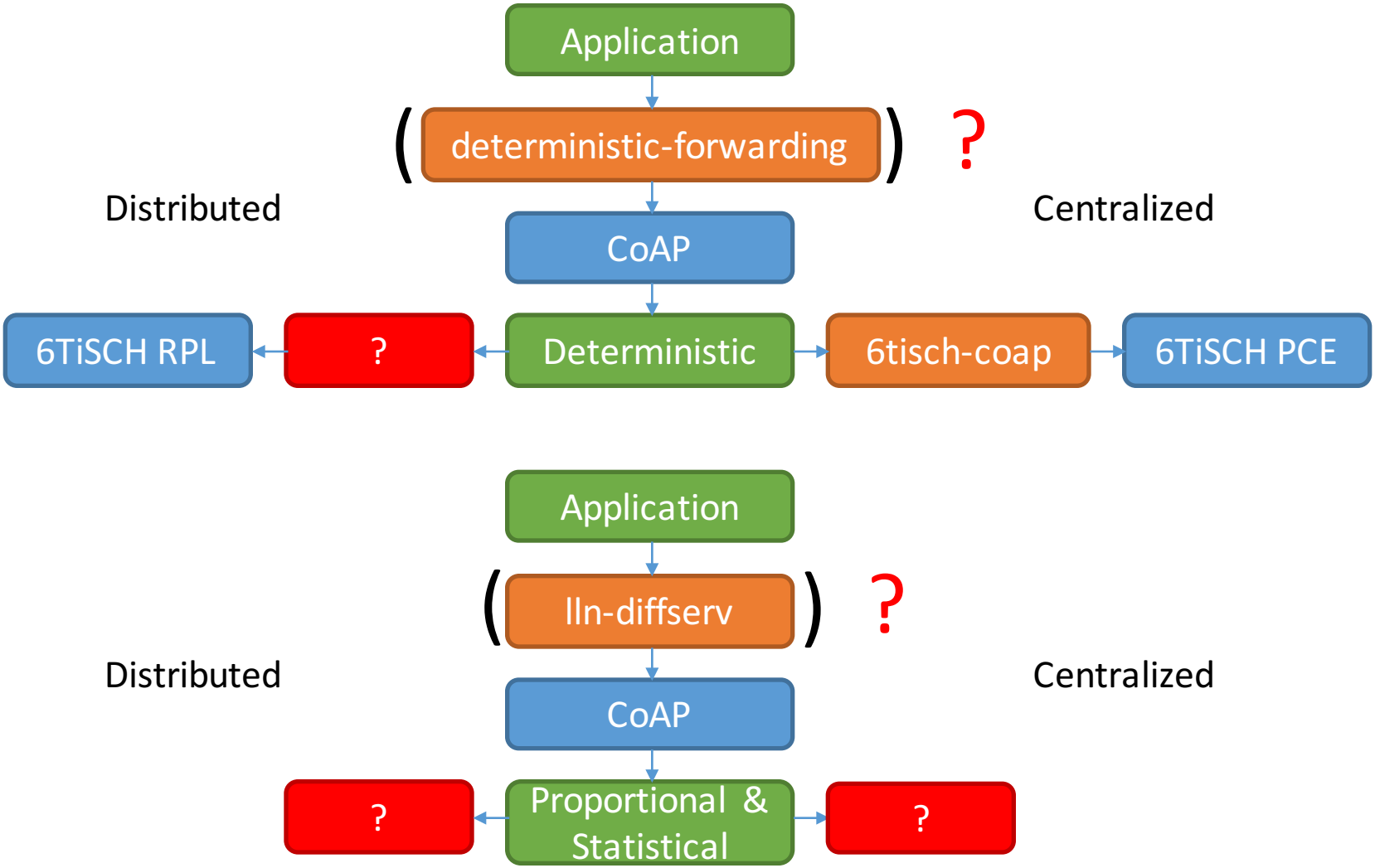
Cross-Layer QoS

- MAC
  - Differentiation scheduler
- Transport
  - Classifier
    - Maps application's priorities to network's priorities
  - Delay monitor
    - Measures delays, e.g. through timestamps
  - Feedback adapter
    - Dynamically adjust priorities to meet requirements
- Application
  - QoS requirements adapter
    - Generates delay requirements

Cross-Layer implementations



Cross-Layer implementations



Questions?

Questions?

Questions?

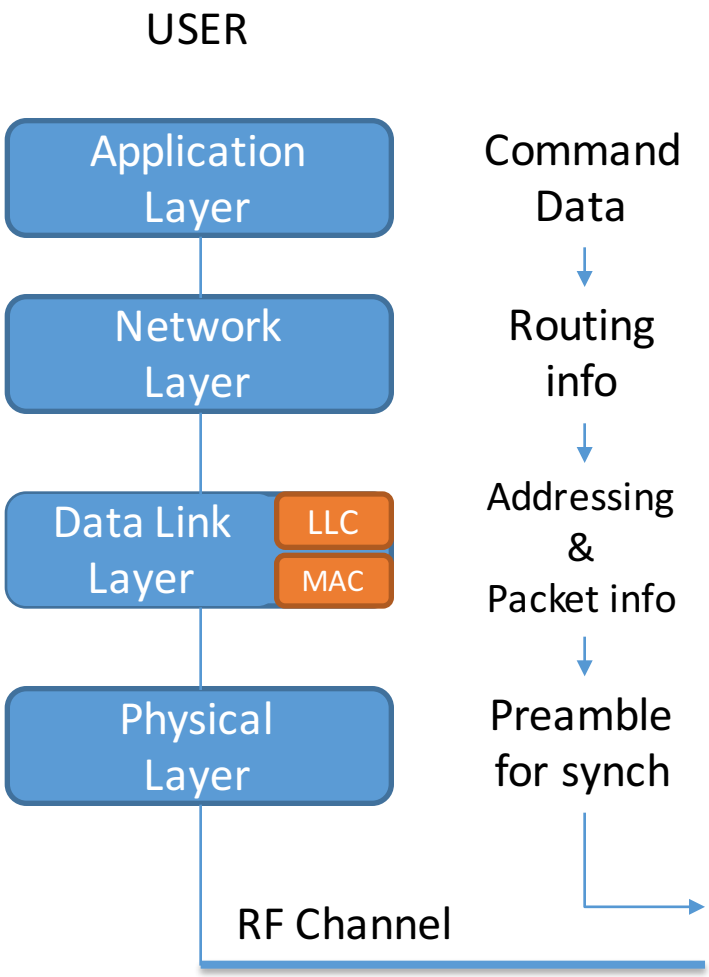
Questions?

ZeroMQ

- Asynchronous message exchange framework
- Many communication patterns:
  - Publisher/Subscriber
  - Request/Response
  - Pipeline
  - …
- Many underlying protocols
  - Inter process communication
  - Multicast (pgm)
  - TCP
  - …
- No explicit QoS support
  - Relies on underlying transport protocol and network devices configurations

WirelessHART

- Industrial standard, extension of the HART (cabled) standard
- TDMA, strong synchronization
- Centralized coordination
  - Routes forwarded to devices
  - Graph Routing
- Reliable and Unreliable messages



(Needed) components

- Admission control
- Enforcement
- Monitoring/Adaptation
- Routing

- All components not always needed
  - Enforcement and Monitoring/Adaptation can be enough for statistical/proportional QoS

Admission

- Stores admitted flows
- Keeps track of resource usage
- Decision based on application QoS requirements and available resources

- Central point of coordination

Enforcement

- Regulates sending rates
- Manages negotiated flows and best-effort
- Must be cross-layer

Monitoring/Adaptation

- Reactive approach
- Re-allocate resource in case of issues
- Must be cross-layer

Routing

- QoS based best route
- Difficult in distributed environment
- Centralized approach suffer from scalability