

Vehicular Networks

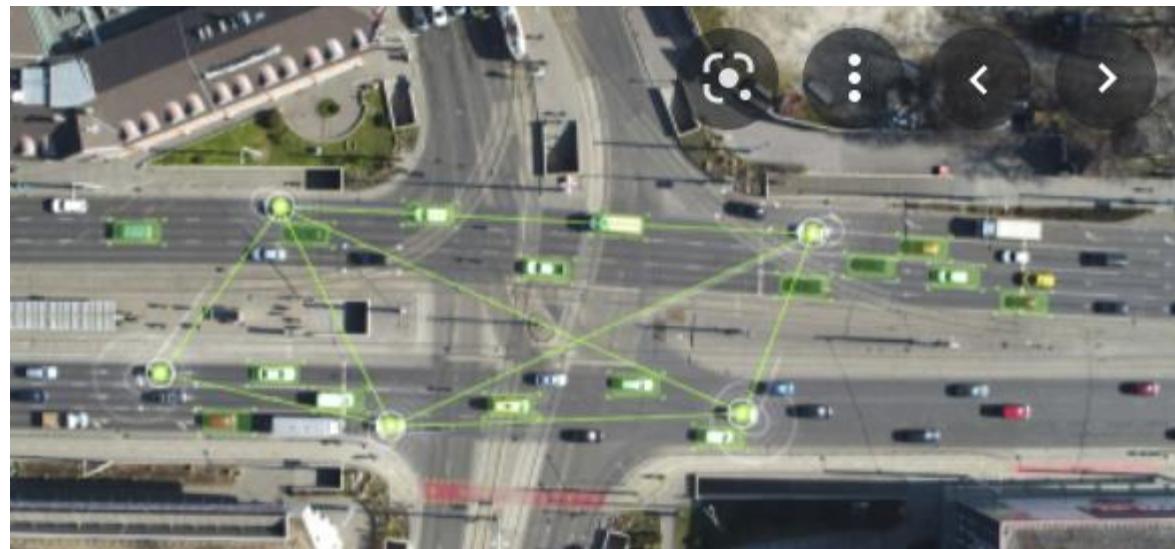
**Mestrado em Engenharia de
Computadores e Telemática**

2022/2023

Vehicular Ad Hoc Networks

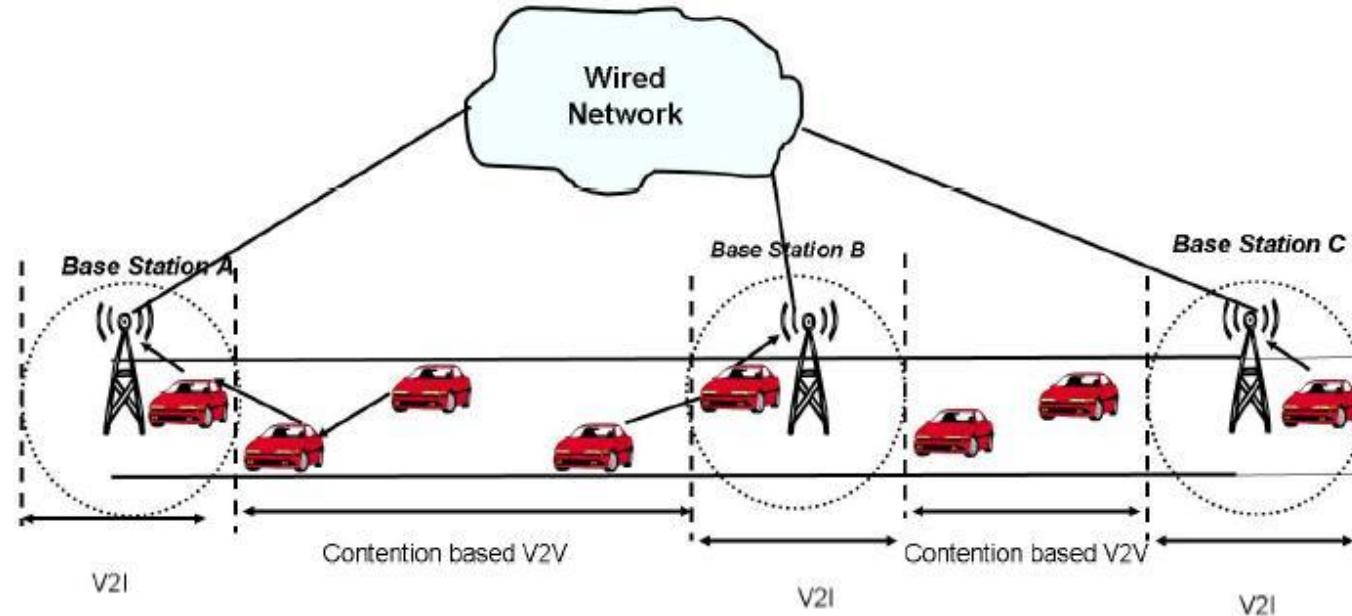
The VANET can provide

- Safety
- Efficiency
- Traffic and road conditions
- Road signal alarm
- Local information



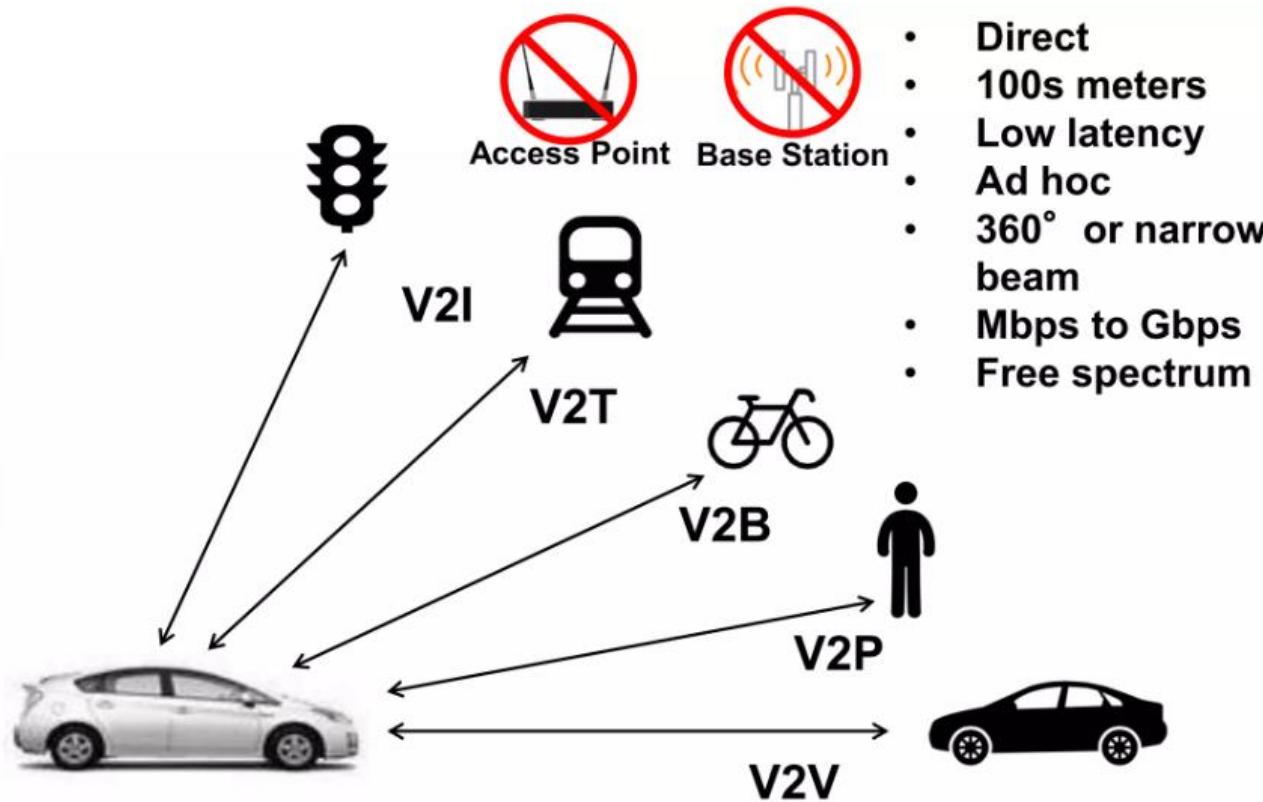
Vehicular networks

- Vehicle-to-vehicle (*V2V*) and vehicle-to-infrastructure (*V2I*) communication will be possible
- On-board units in vehicles to perform communication, routing and application
- Road-side infrastructure units (RSUs), named network nodes, are equipped with on-board processing and wireless communication modules

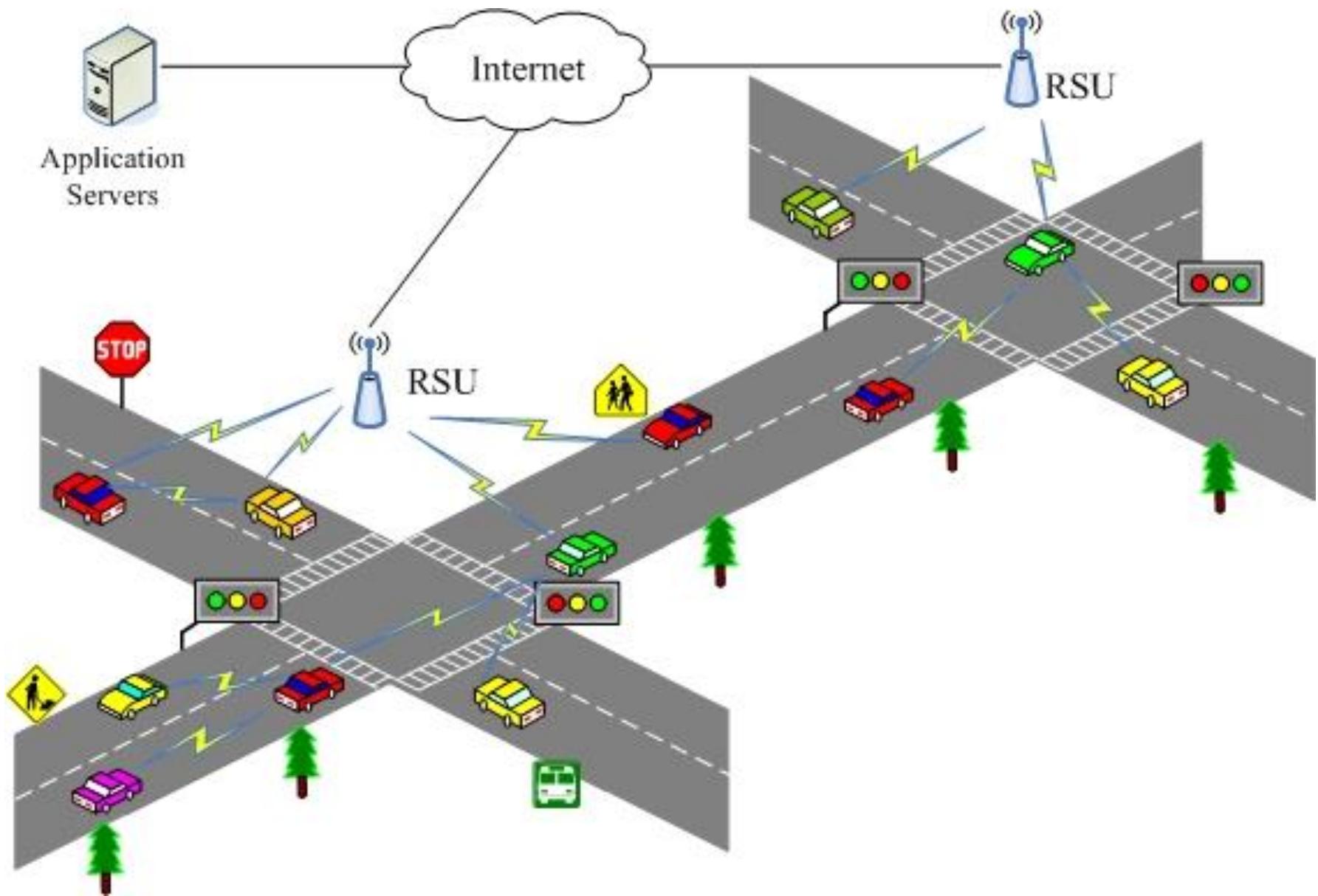


Vehicle to Everything Communication

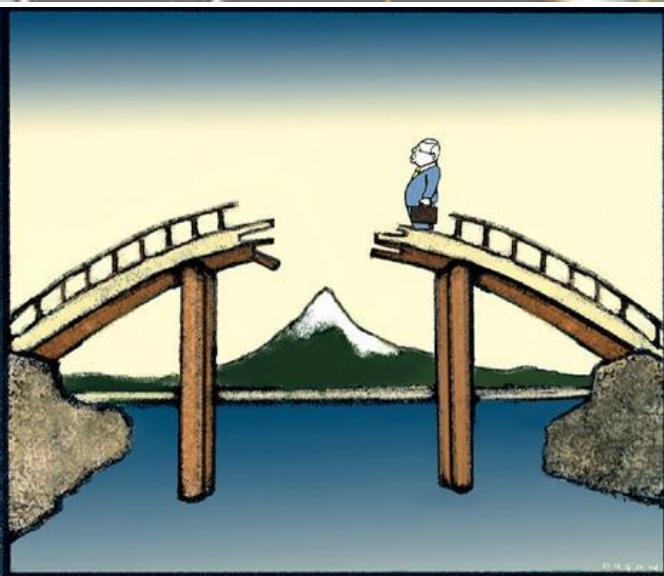
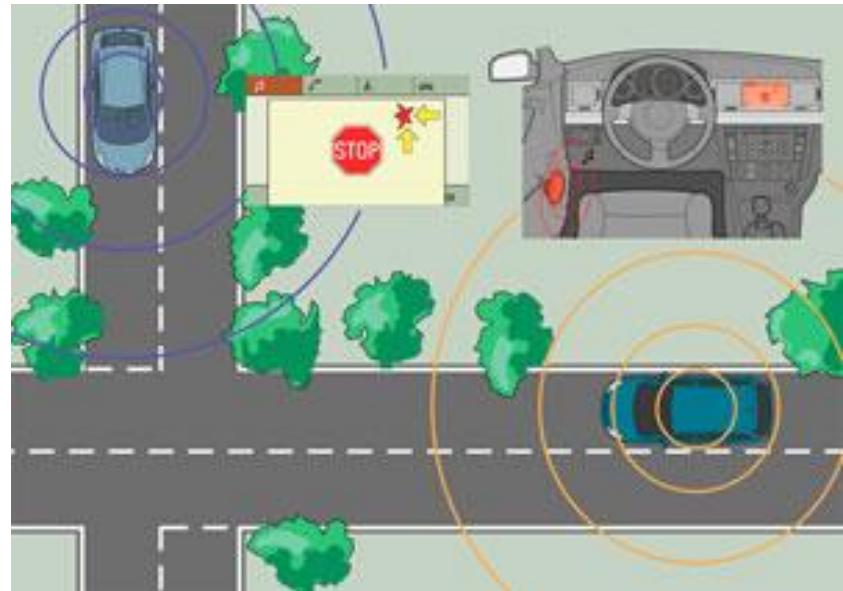
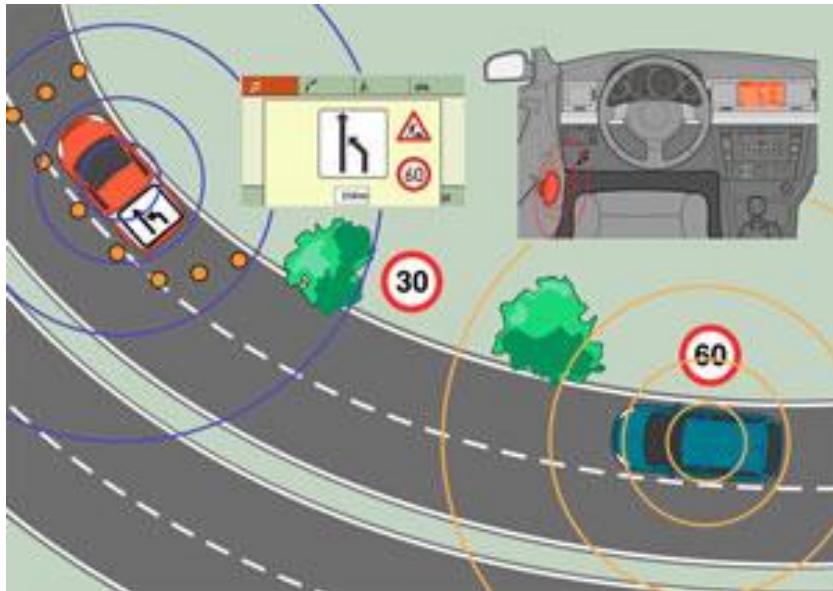
- Initially, before 5G



Vehicular networks



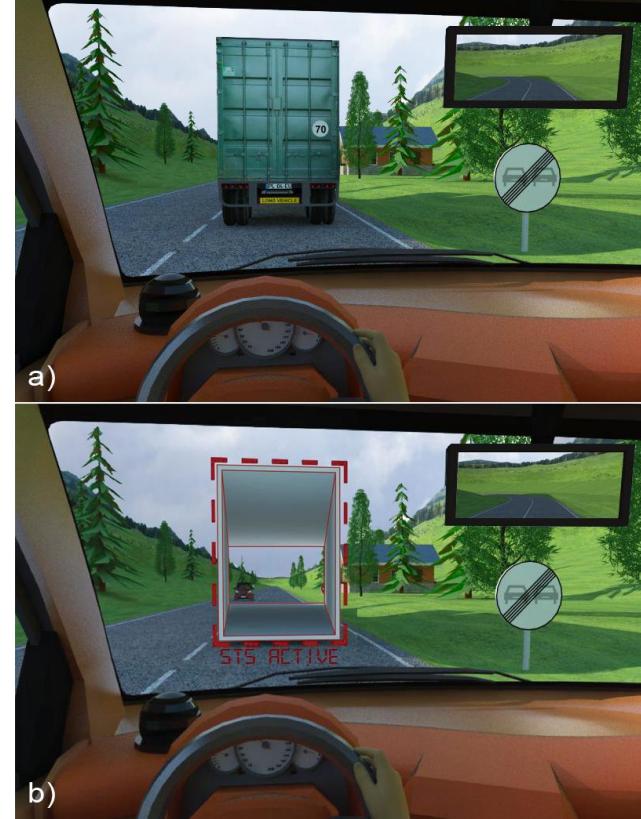
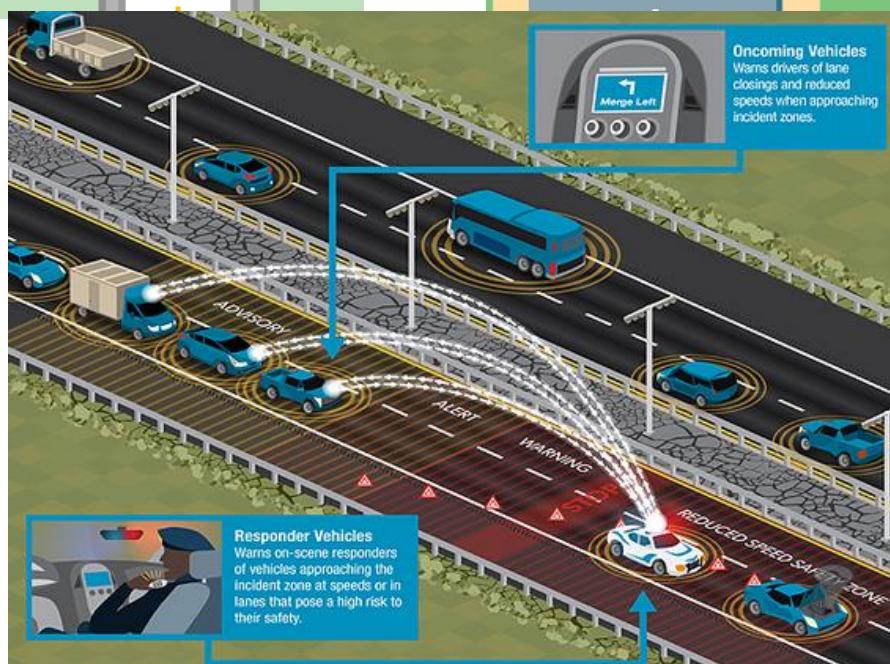
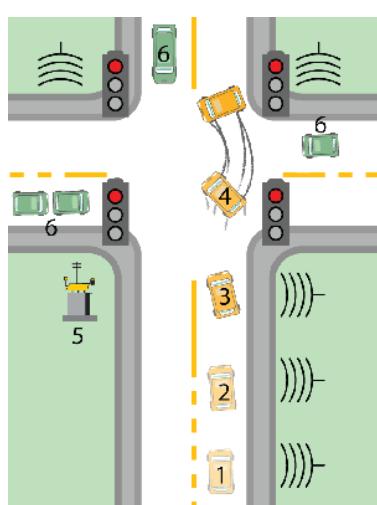
Warnings



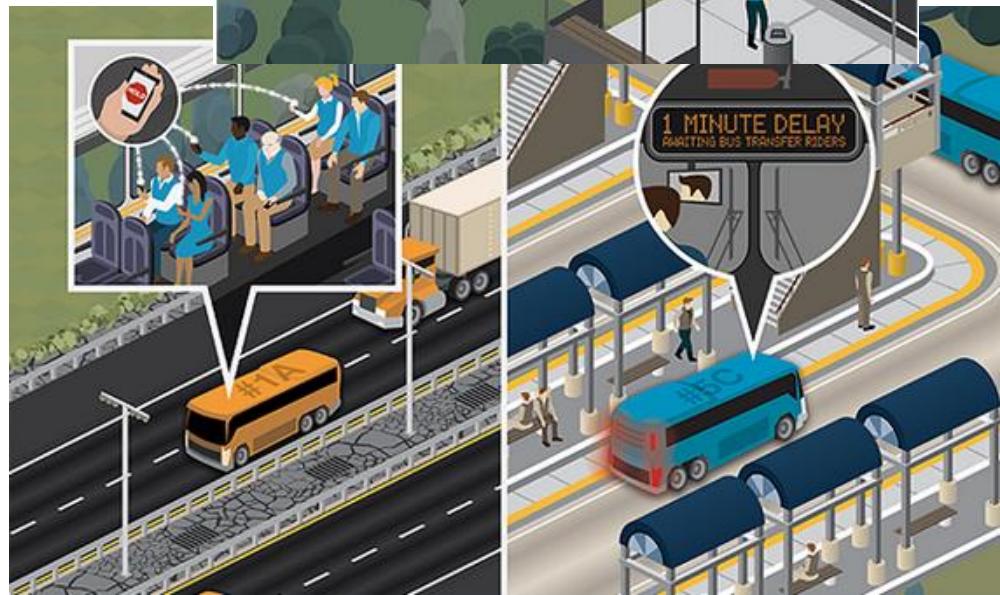
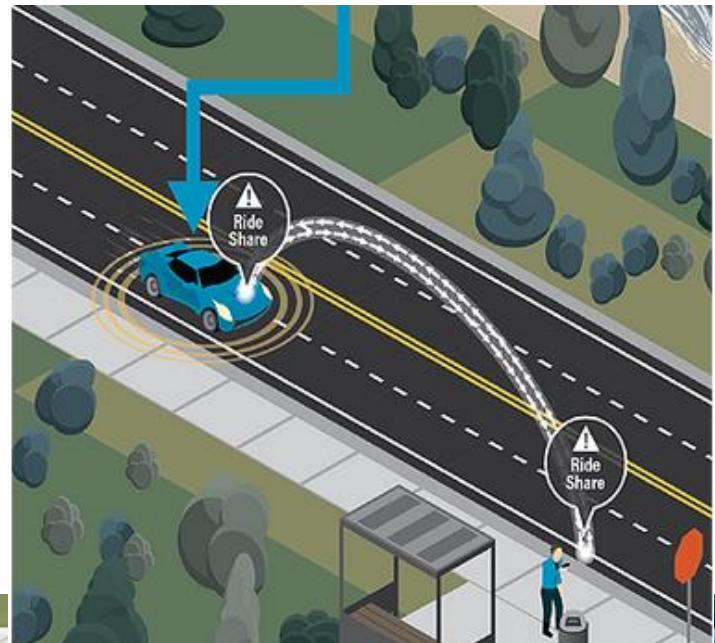
Traffic and road conditions



Safety



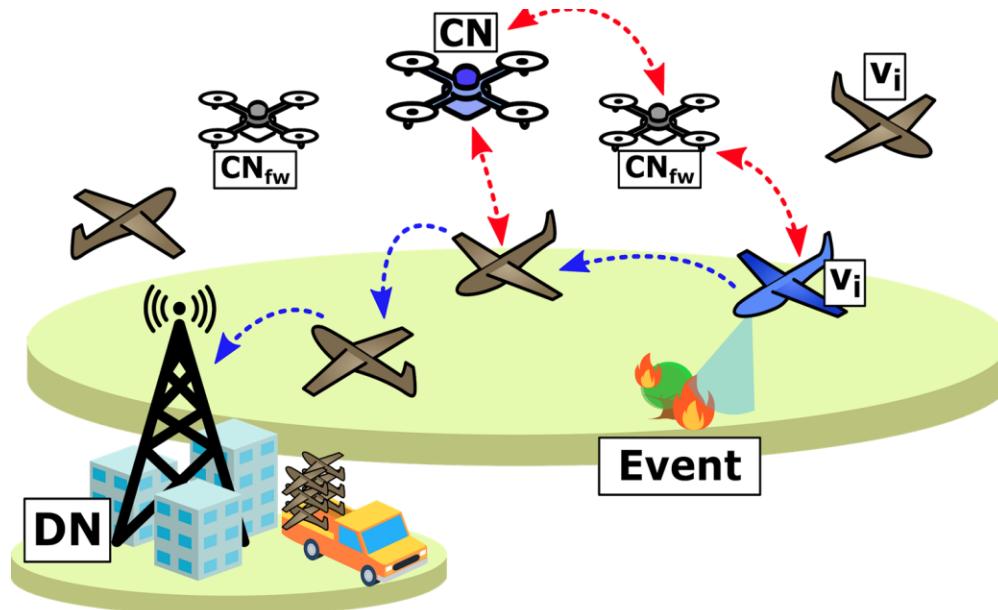
Efficiency



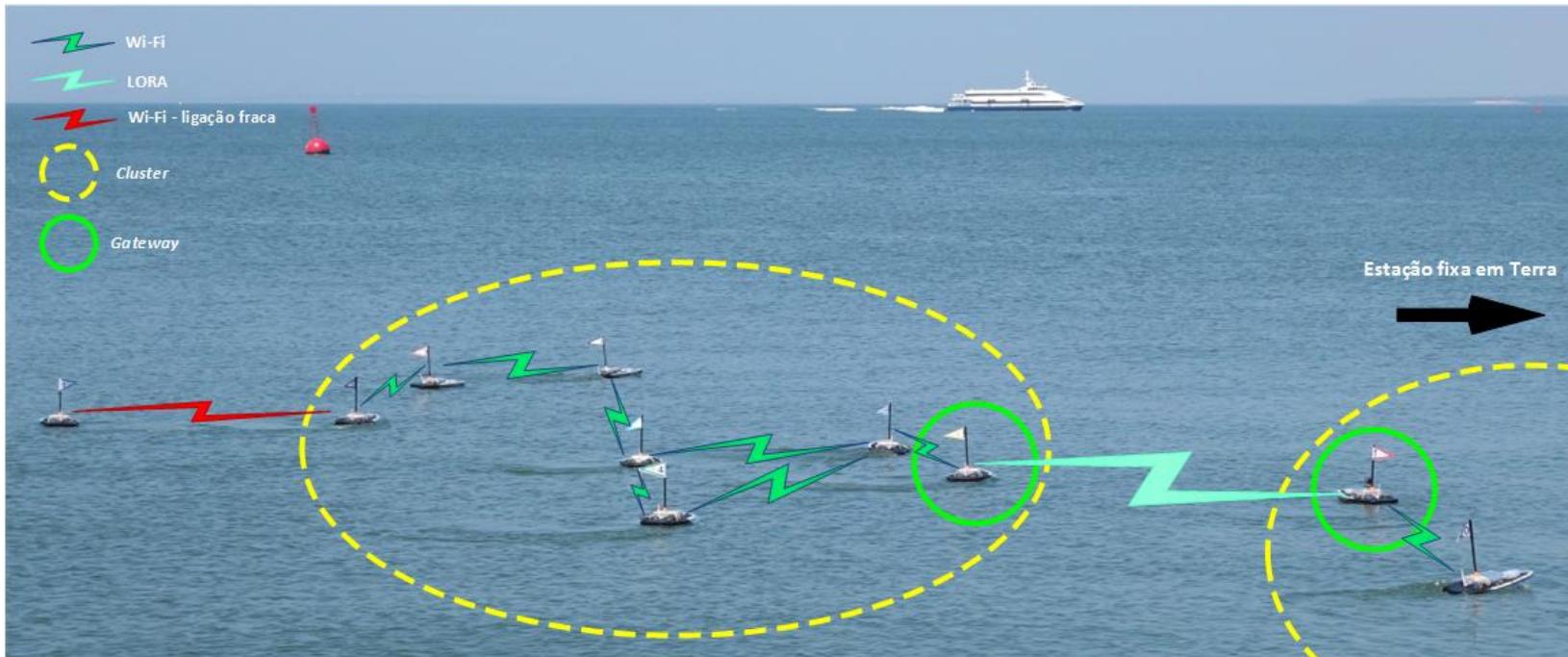
SelfDriving

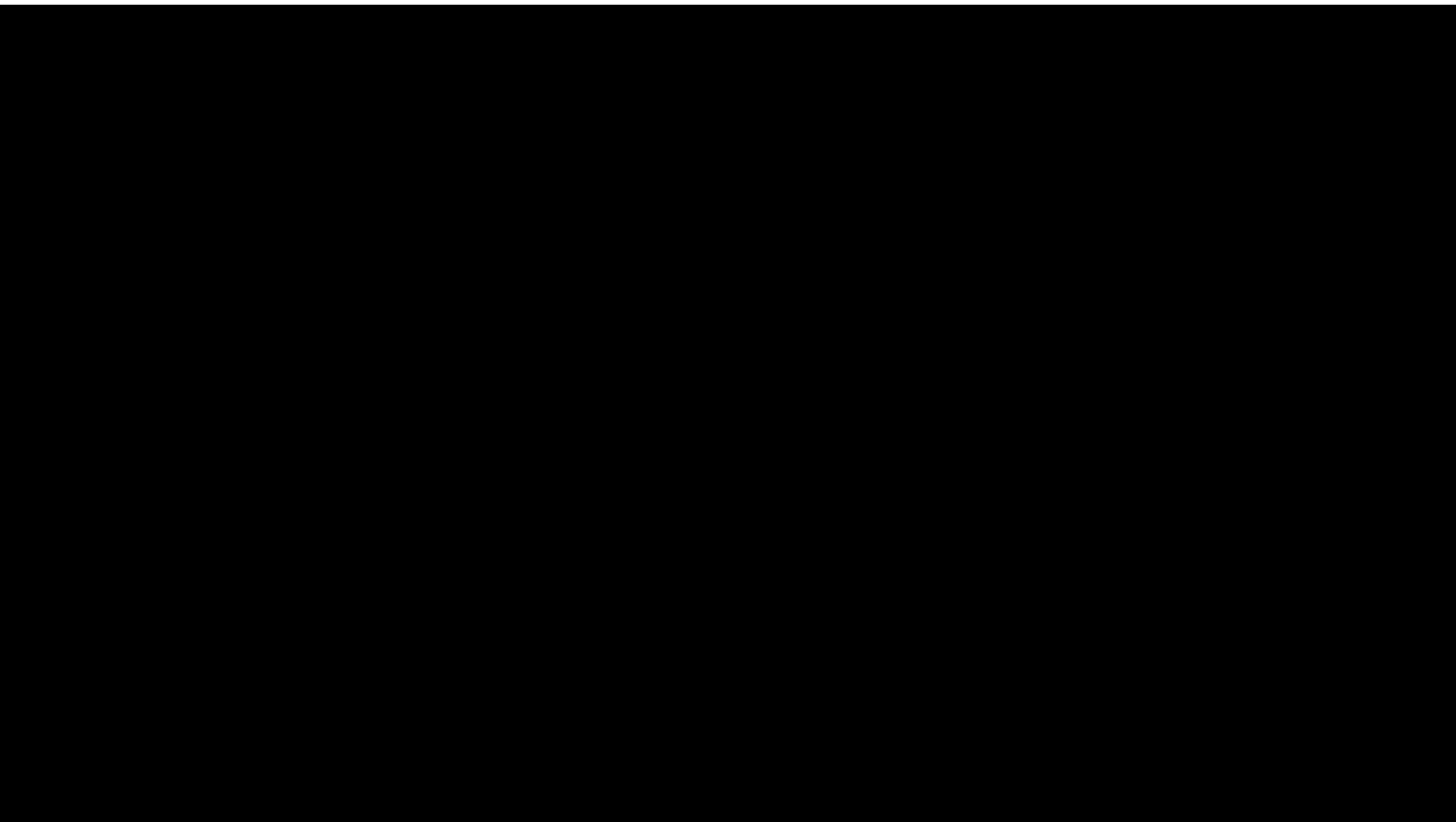


Other types of vehicular networks

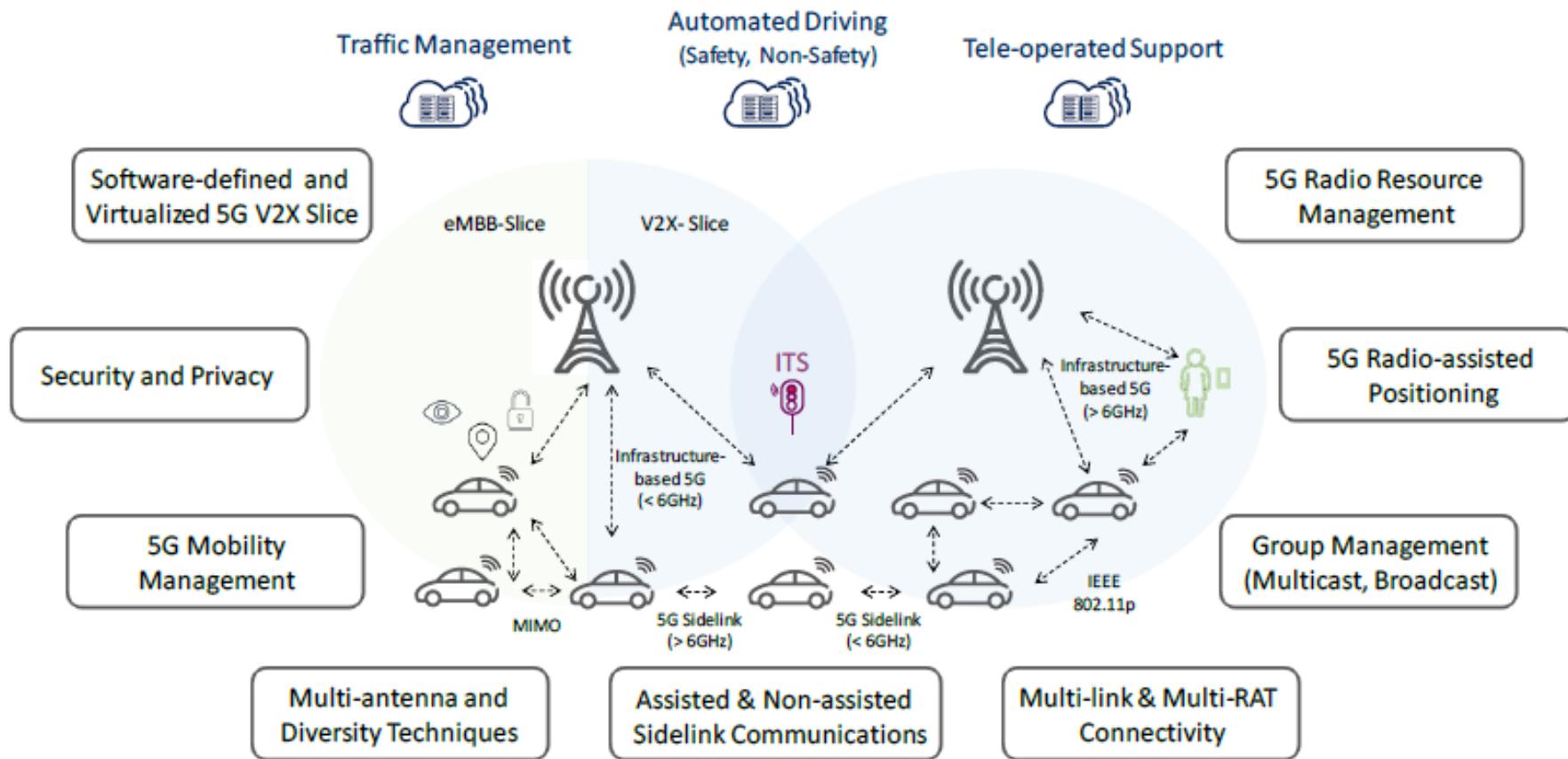


Other types of vehicular networks



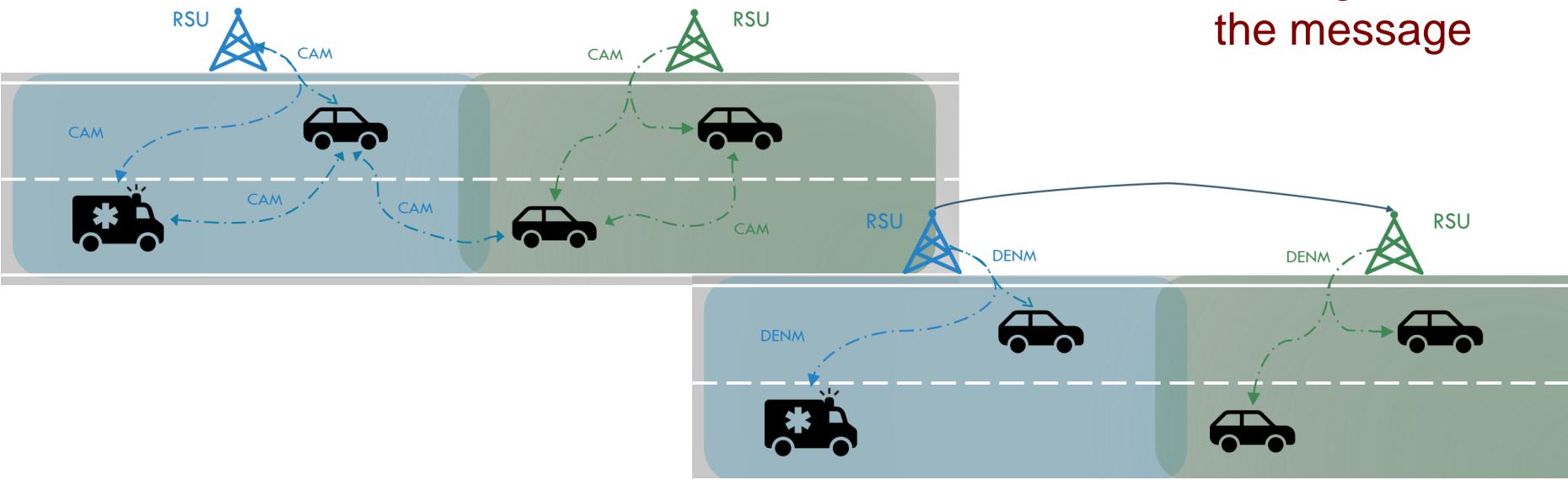


How do they work?

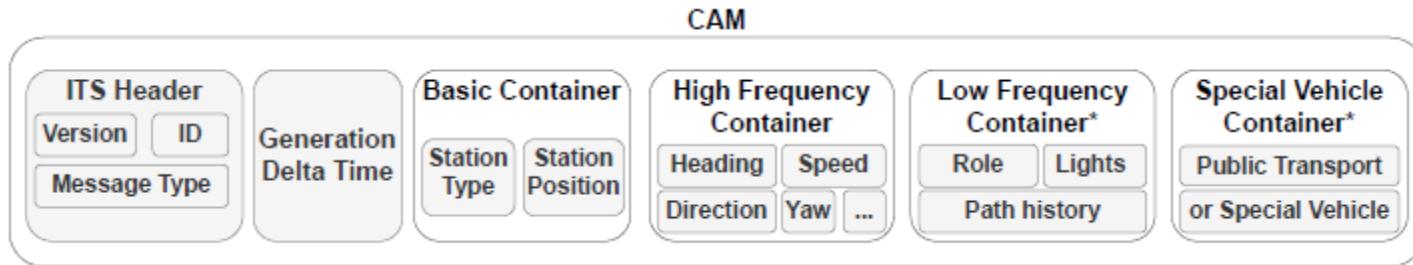


Awareness and warning information

- Cooperative Awareness Messages (CAM)
 - Periodic
 - Contain information about the station such as the position and speed
- Decentralized Environmental Notification Messages (DENM)
 - Asynchronous
 - Contain information about the event and the station that generated the message



Cooperative awareness messages



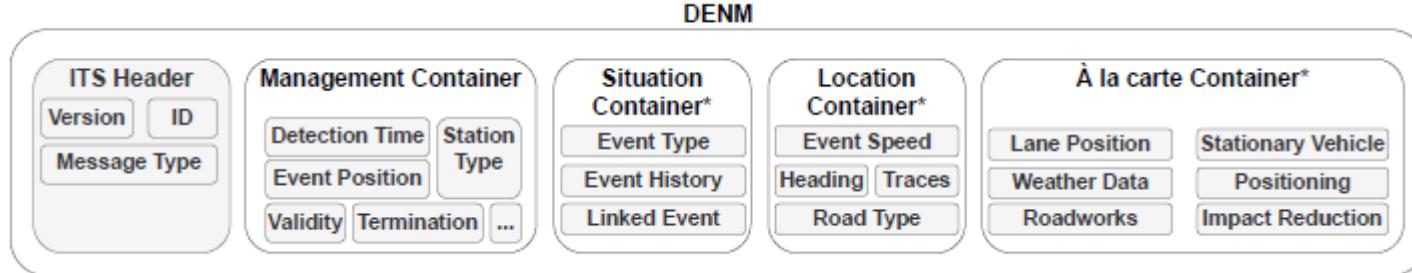
- Create and maintain awareness of vehicles using the road network or RSUs.
- The content varies depending on the type of ITS-S:
 - Vehicles: time, position, motion state, activated systems (*e.g.*, cruise control, pedals, and others), and the attribute information includes data about the dimensions, vehicle type, and role in the road traffic;
 - RSUs: station type and location.
- HF (High-Frequency) container with the fast-changing vehicle data (such as location, heading, or speed)
- LF (Low-Frequency) container with static or slow-changing data (such as the status of the exterior lights or pedals).

Cooperative awareness messages

- CAMs have generation requirements, with the generation frequency between 1 Hz and 10 Hz.
- The HF container must be in every CAM message, while the low-frequency container can be updated at a maximum of 5 Hz frequency.
- The generation process must be effective, since the difference between CAM generation time and the time at which the CAM is delivered to networking transport layer shall be less than 50 ms.

CAM Information	Basis Container	ITS-Station Type Last Geographic Position
	High Frequency Container	Speed Driving Direction Longitudinal Acceleration Curvature Vehicle Length Vehicle Width Steering Angle Lane Number ...
	Low Frequency Container	Vehicle Role Lights Trajectory Emergency Police Fire Service Road Works Dangerous Goods Safety Car ...
	Special Container	

Decentralized environmental notification messages

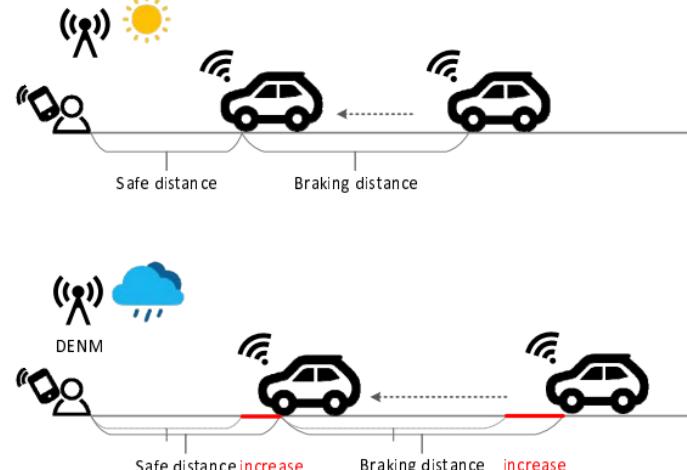


- Asynchronous messages to create and maintain awareness about a road event - *e.g.* road hazard or an abnormal traffic condition - such as its type, position, validity, timestamp and the history of the event.
- While the content varies depending on the type of event, it is expected that at least the detection time, the position of the event, the type of the related station and a set of cause codes identifying the type of event are present.
- Containers for certain types of events - such as the Road Works
- Stationary Vehicle Containers.

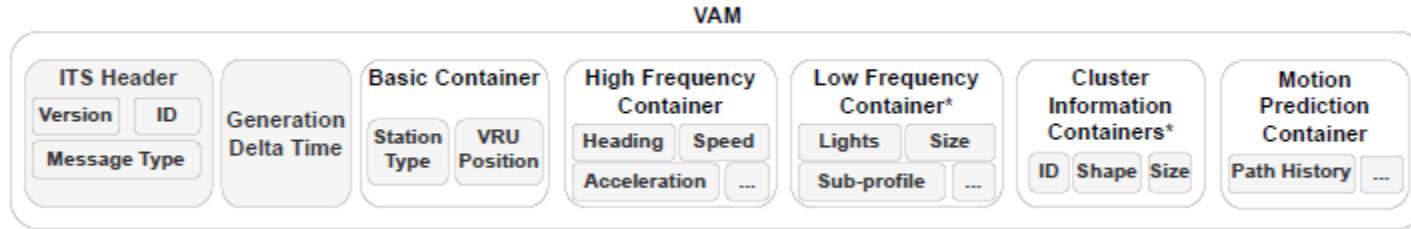
Decentralized environmental notification messages

- Unlike CAMs, DENMs are generated as events occur, and thus, they are not generated periodically.
- They have a validity period, which after ending, means the DENM can no longer be considered up-to-date.
- When an event is no longer occurring, a particular type of DENM, a termination DENM, can be used to signal the end of the event - *e.g.* the end of the road hazard or of adverse weather conditions.

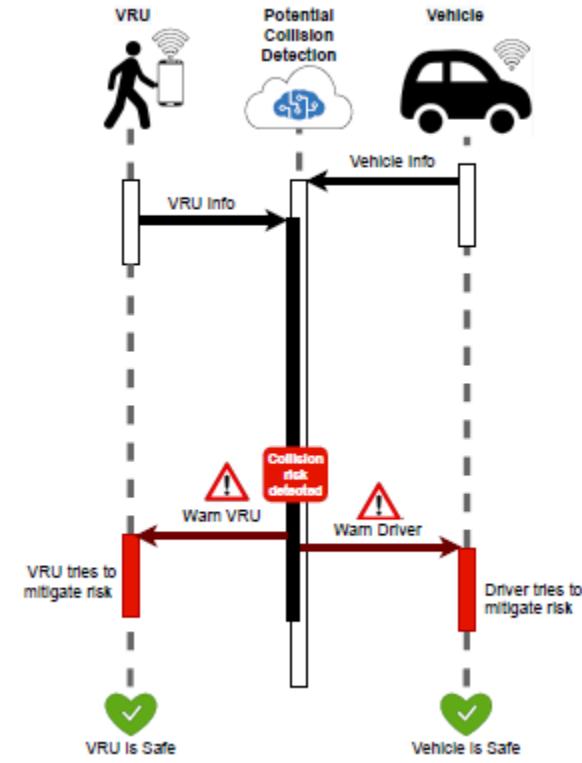
DENM Information	Management Container	Last Vehicle Position (GPS)
		Event Identifier
		Time of Detection
		Time of Message Transmission
		Event Position (GPS)
		Validity Period
		Station Type (Motor Cycle, Vehicle, Truck)
		Message Update / Removal
		Relevant Local Message Area (geographic)
		Traffic Direction (forward, backwards, both)
		Transmission Interval
	
	Situation Container	Information Quality (low -high, tbd)
		Event Type (Number)
		Linked Events
		Event Route (geographical)
	Location Container	Event Path
		Event Speed
		Event Direction
		Road Type
	A la carte Container	Road Works (Speed Limit, Lane Blockage....)
	



Vulnerable Road User Awareness Message



- Periodic messages exchanged in the ITS network between stations to create and maintain awareness on VRUs, and support the risk assessment
- Basic status: time, position, speed, heading, yaw rate and acceleration, orientation, lane position, dimensions and VRU type.
- Advantages of a VAM standard message over the usage of a CAM
 - flexibility in terms of fully specifying the VRU type and situation, which is not possible without changing the CAM standard (therefore defeating the purpose of using a standard).

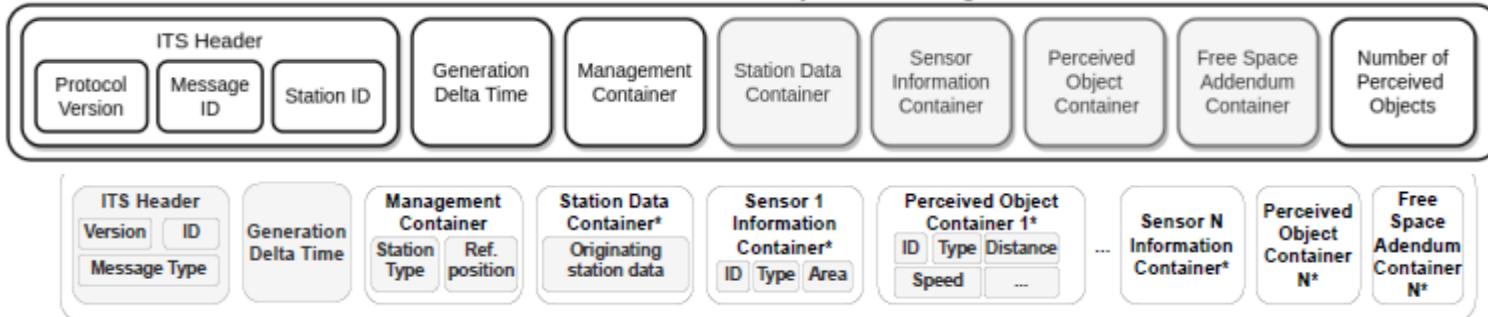


Vulnerable Road User Awareness Message

- VAMs can distinguish between several types of VRU - pedestrian, cyclist, motorcyclist, animal
- Within each category, they can distinguish several possible roles (*e.g.* for a cyclist VRU, between bicyclist, a wheelchair user, a horse rider, a roller skater, an e-scooter, and others).
- This distinction is crucial: several different VRUs - for example, a child pedestrian or a disabled pedestrian - have different dynamics from a typical pedestrian. That information can be used, for example, by safety services to fine-tune an accident prediction algorithm.

Cooperative Perception Message

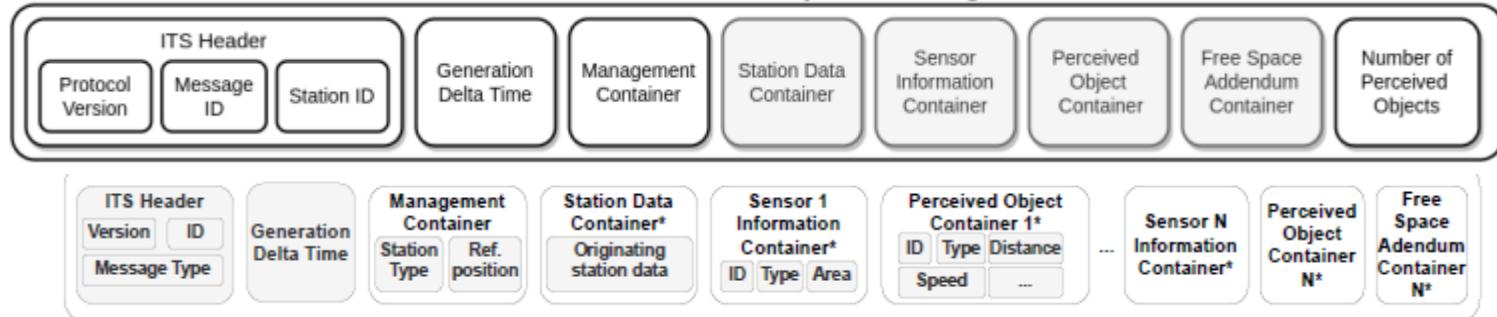
Collective Perception Message



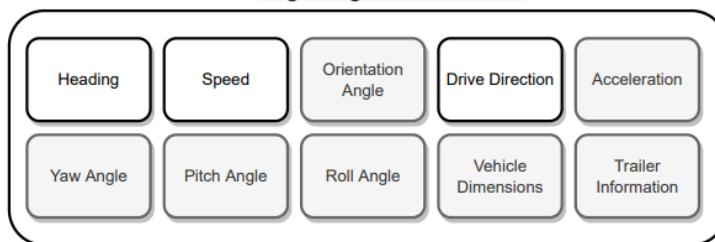
- Periodic messages between stations to broadcast information about the current environment perceived by 1 or more sensors.
- Sensors from a vehicle, a VRU and infrastructure can use CPMs to exchange the information obtained from their surroundings, improving the awareness of the situation.
- Sensor Information Container: sensor type - *e.g.* Radar, Lidar, video cameras or fusion algorithms, and the area the sensor covers.
- Perceived Object Container: object perceived by the sensor, the classification, the confidence of the classification, and several data about its dynamics, such as distance, speed, acceleration and angle.

Cooperative Perception Message

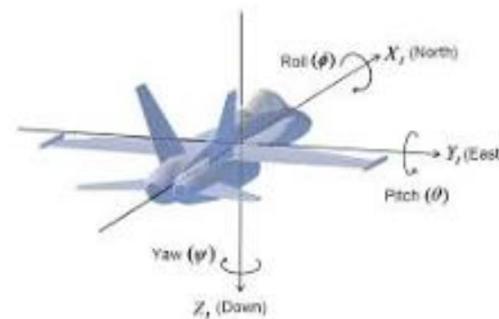
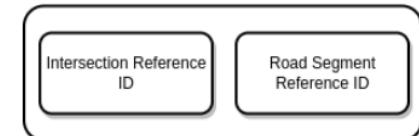
Collective Perception Message



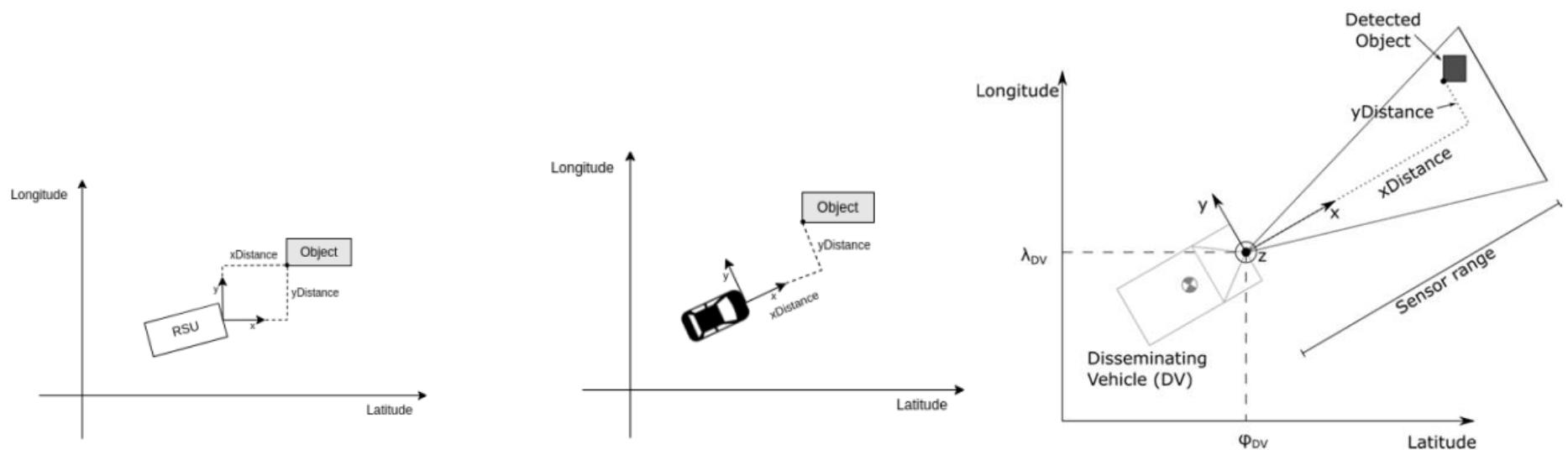
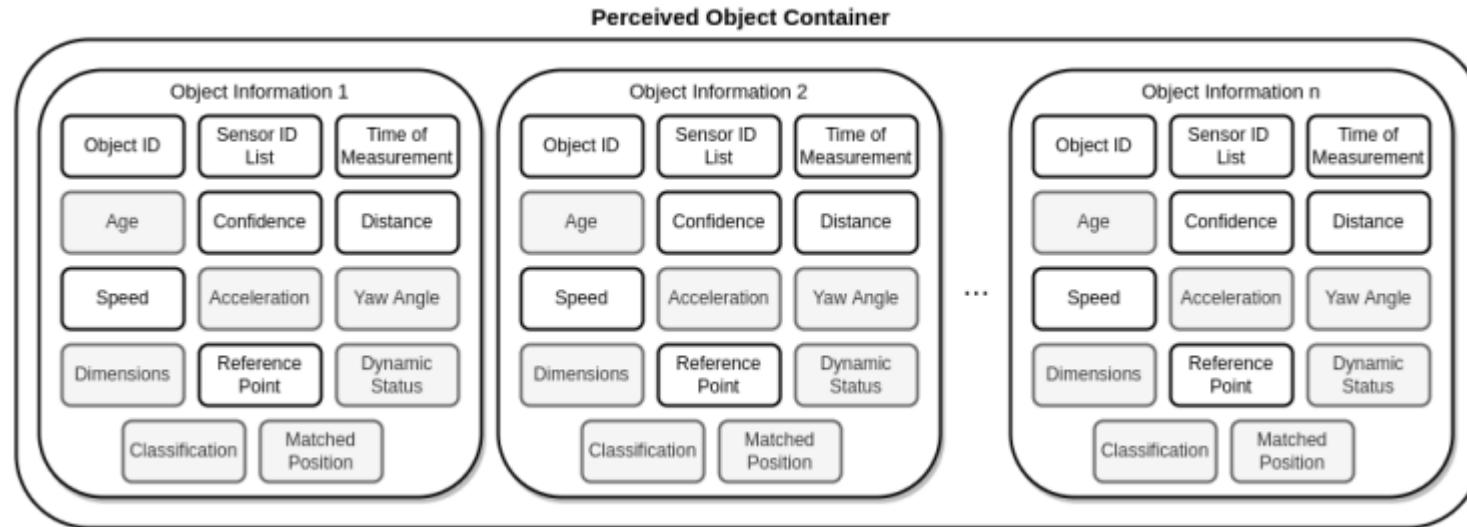
Originating Vehicle Container



Originating RSU Container



Objects information in CPMs



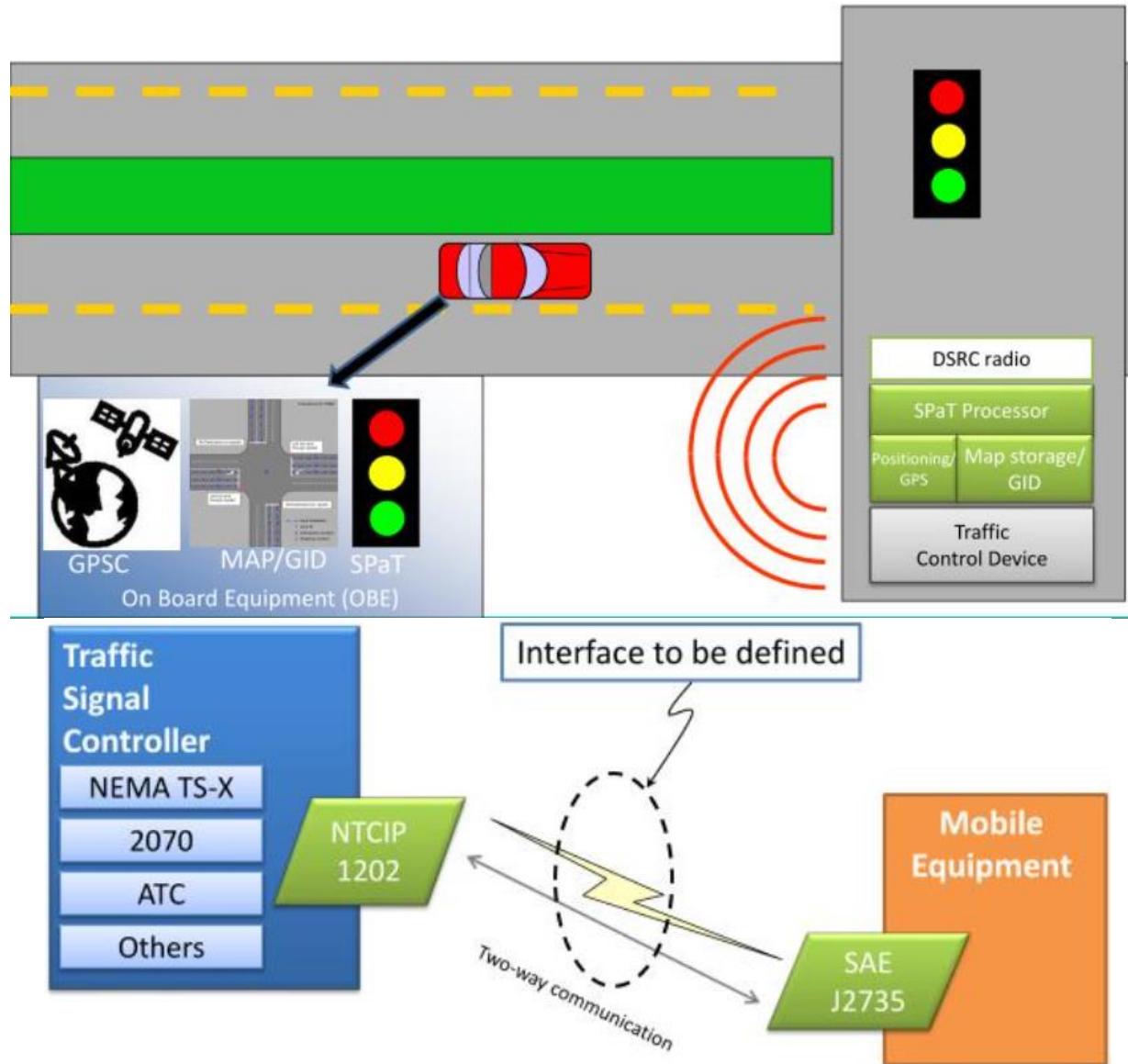
Coordinate System for detected object for vehicle

SPAT: Signal Phase And Timing

- Defines an open interface for two-way communication between traffic signal controller and mobile devices
- Current movement state of each active phase
 - Safety applications, such as warnings and alerts for crash avoidance, red light violations
 - Mobility applications to enable dynamic and efficient traffic management
 - Environment applications that allow savings in fuel consumption and reduction in CO₂ emissions
- Current state of all lanes at the intersection are provided, as well as any active preemption or priority

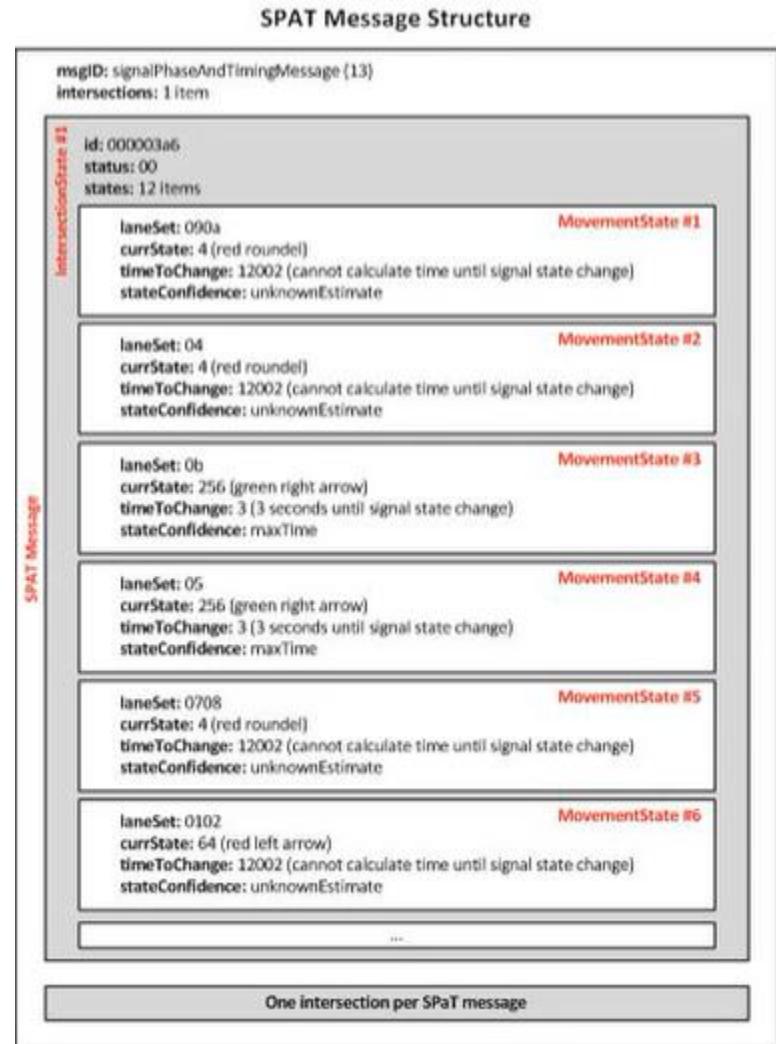


SPAT: Signal Phase And Timing

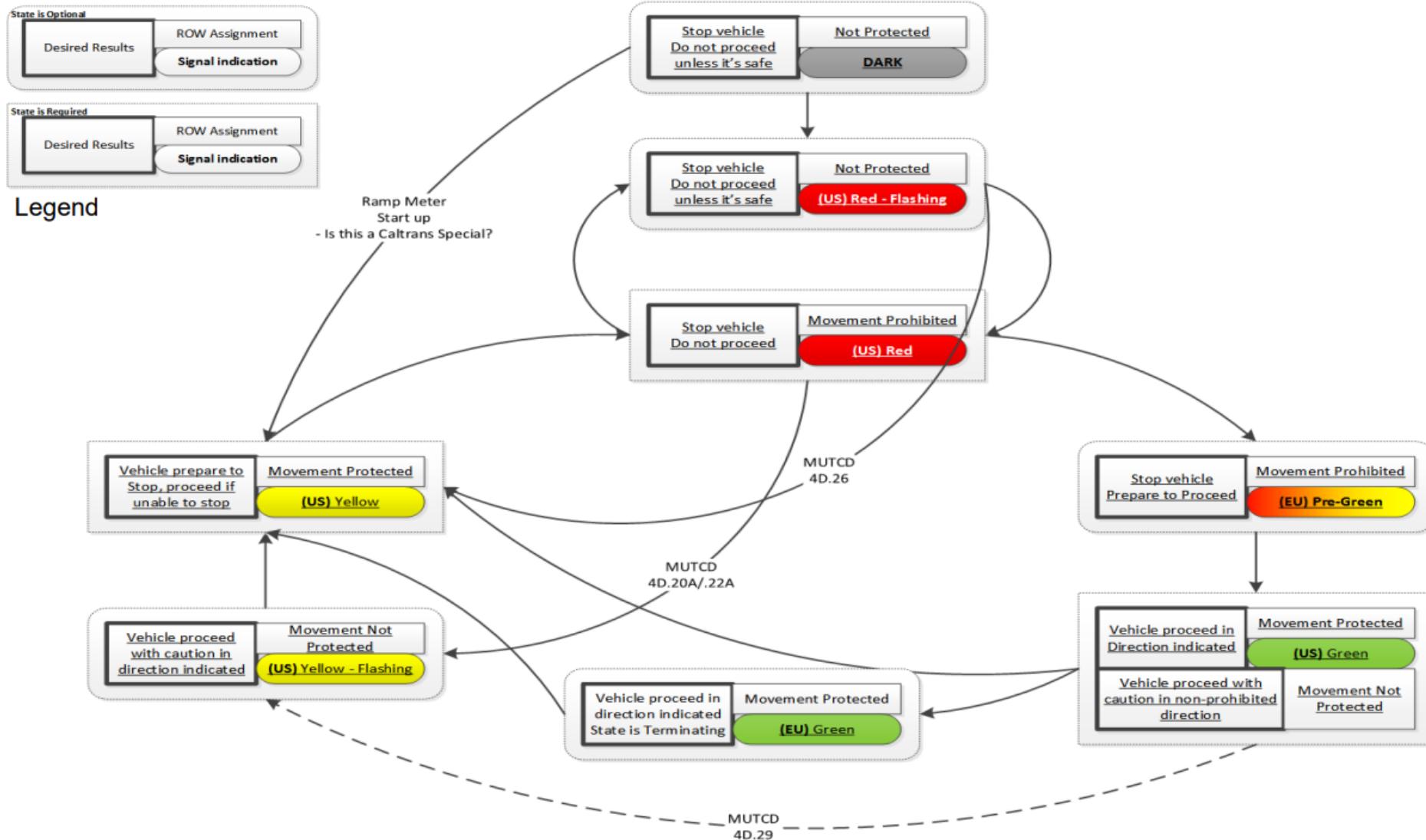


SPAT: Signal Phase And Timing

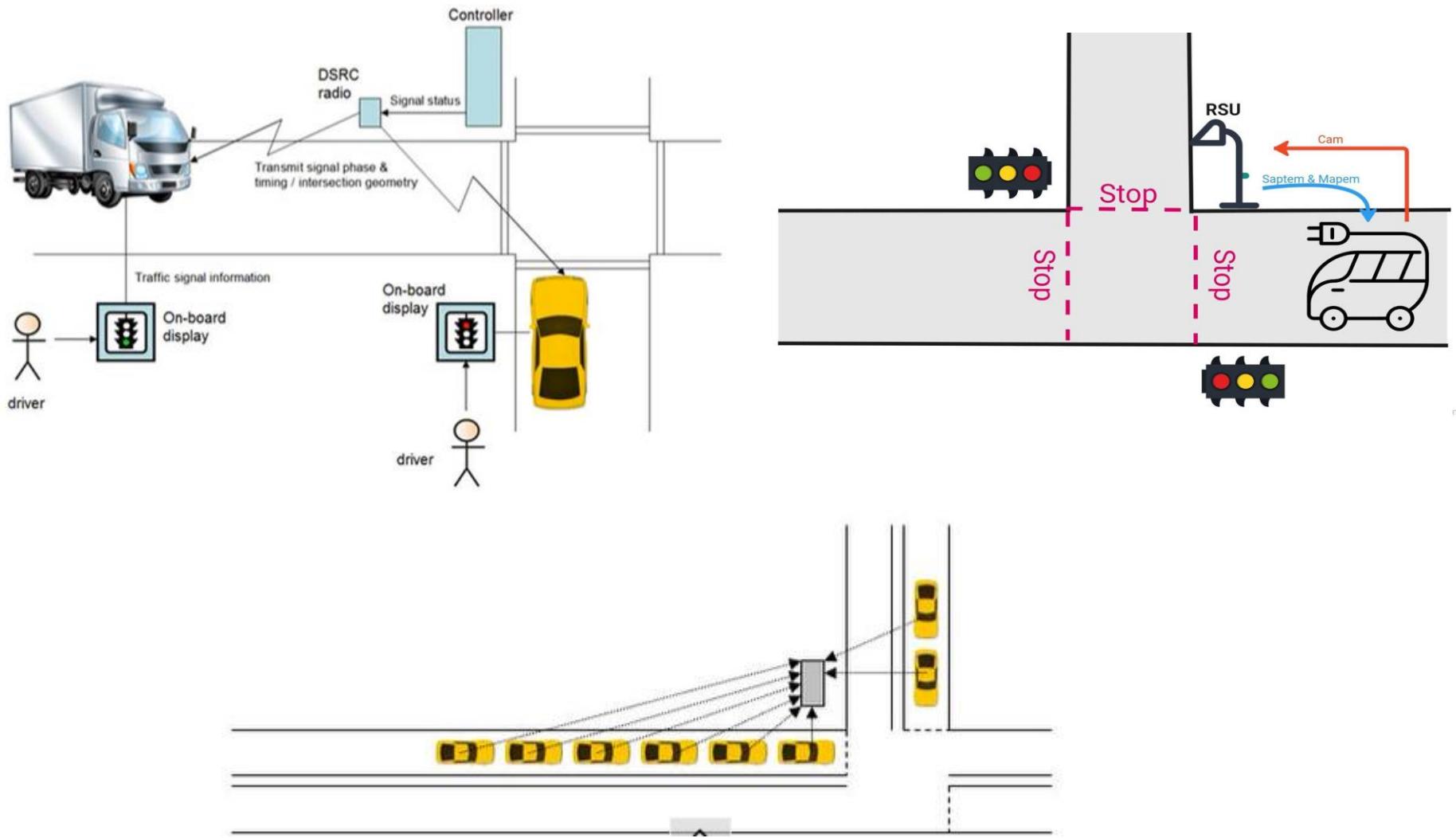
- Intersection state
- Movement state
 - Lane set (lanes 9-10 are for movement state 1)
 - Current state (green, yellow, red)
 - Time until current signal state changes
- Used in cooperation with a map



SPAT: State diagram

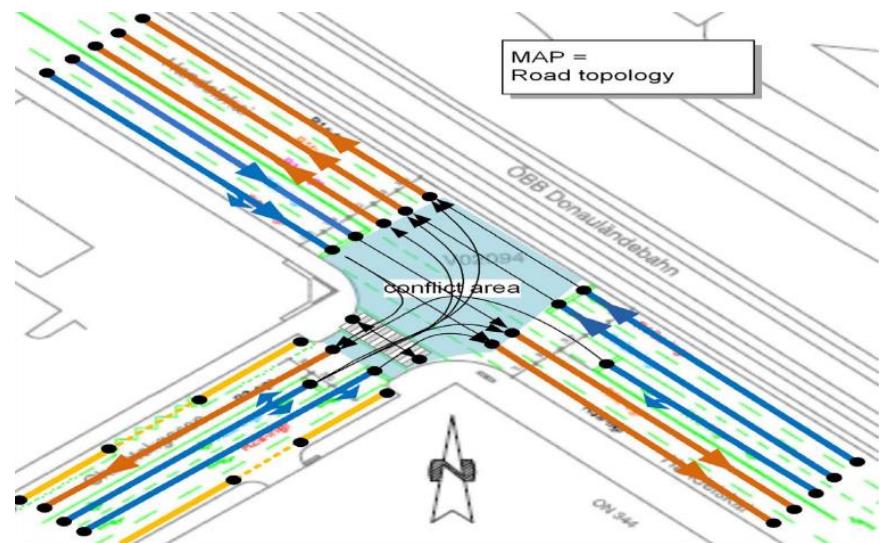


They can be used virtually



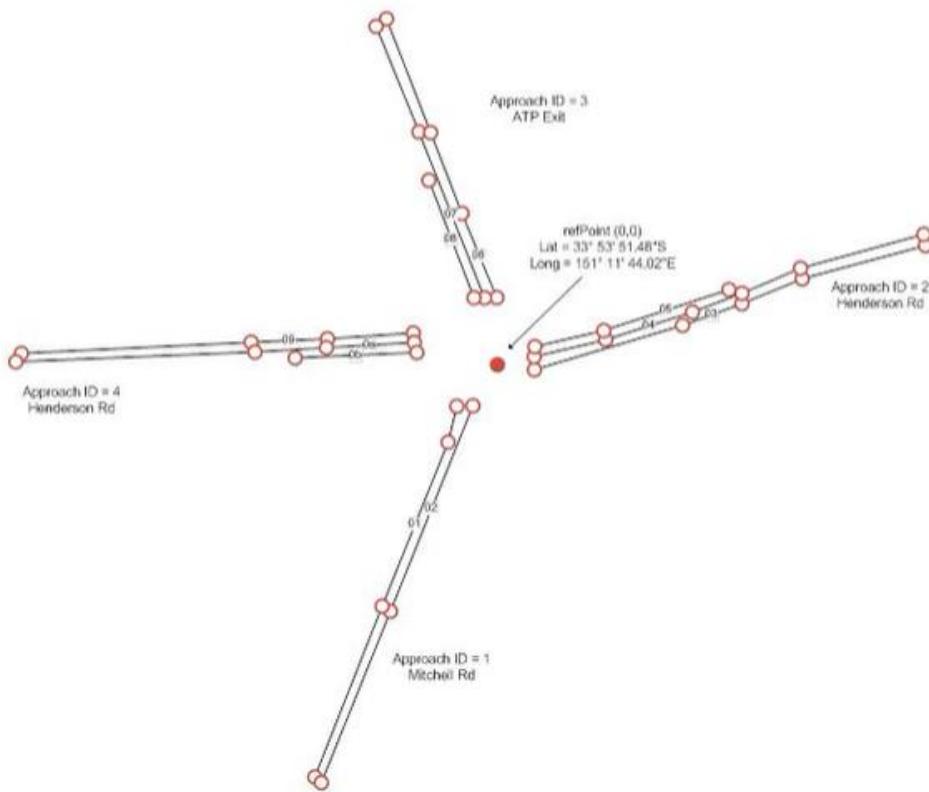
MAP: MAP

- Geometric layout of intersection
- Message data
 - Reference point (intersection center)
 - Number of approaches
 - Lane number
 - Lane width
 - Lane attributes
 - Straight, left, right, turn on red, speed limit, bus, etc...
 - Offsets
 - Points along each lane – used to detect vehicle position



Example of a MAP information

refPoint [Position3D]	The Position3D data frame provides a precise location in the WGS-84 coordinate system, from which short offsets may be used to create additional data using a flat earth projection centred on this location.	
laneWidth [LaneWidth]	The LaneWidth data element conveys the width of a lane in units of 1 cm.	
speedLimits [SpeedLimitList] (1..9)	The SpeedLimitList data frame consists of a list of SpeedLimit entries.	RegulatorySpeedLimit The RegulatorySpeedLimit data frame is used to convey a regulatory speed about a lane, lanes, or roadway segment.
[laneSet] LaneList (1..255)	The LaneList data frame consists of a list of GenericLane entries.	GenericLane The GenericLane data frame is used for all types of lanes, e.g. motorized vehicle lanes, crosswalks, medians. The GenericLane describes the basic attribute information of the lane.
preemptPriorityData [PreemptPriorityList] (1..32)	The PreemptPriorityList data frame consists of a list of RegionalSignalControl-Zone entries.	SignalControlZone

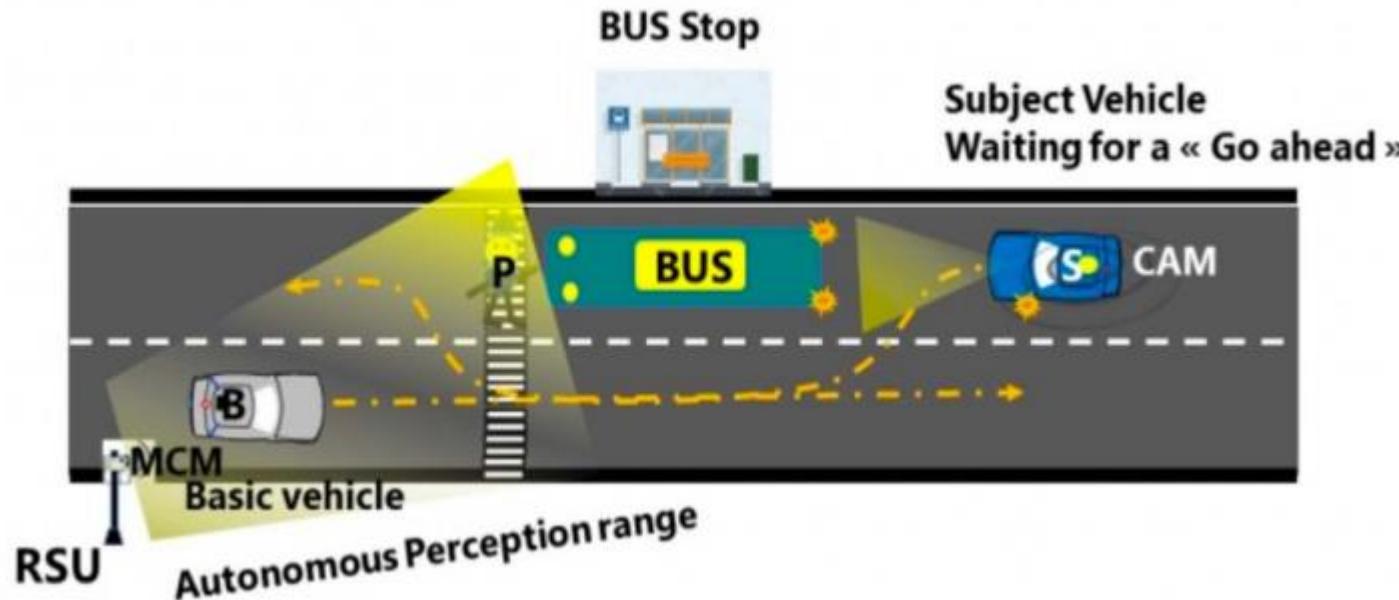


Manoeuvre Coordination Message (MCM)

- Includes the intended (or planned) Manoeuvres and one or more desired (or alternative) trajectories
 - Each trajectory is a spatial-temporal description of the vehicle trajectory in the next 5 to 10 seconds.
 - Planned trajectories are used by applications to improve the prediction of future locations of nearby vehicles and to detect conflicts
 - Desired trajectories are used to request a coordination between vehicles.
- It is expected that MCMs are generated continuously at a rate between 1 Hz and 10 Hz depending on the context
 - Early detection of the need of a Manoeuvre coordination.
 - The MC service can include triggering conditions while also having the possibility of being triggered by an application.
- For roadside applications, the MCMs are expected to include specific advices for specific vehicles, to e.g. suggest a given speed or a lane change
- MCMs transmitted by roadside units are expected to be smaller in size (although they can include advices for multiple vehicles) and transmitted less frequently than those transmitted by vehicles

Manoeuvre Coordination Message (MCM)

- Vehicle S wants to overtake the stationary Bus.
- Any road user or other equipped road safety stakeholder could analyse the road traffic situation based on received information combined with other sensor data
- Advice or manage the road users how to act to realize an efficient and safe resolution of the situation.

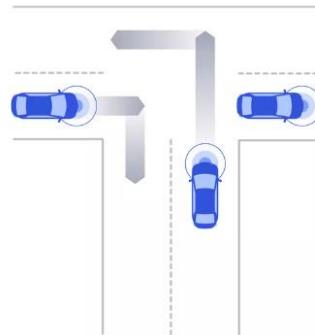


Manoeuvre Coordination Message (MCM) - examples

- Goto Maneuver
- Idle Maneuver
- Follow Path Maneuver
- Follow Trajectory
- Scheduled Goto
- Stop Maneuver
- Maneuver Done
- Teleoperation Maneuver
- Teleoperation Done
- ...
- <https://lsts.pt/docs/imc/master/Maneuvering.html>

Efficient maneuvers

Autonomous vehicles are able to make quicker, yet safe maneuvers by knowing the planned movements of surrounding vehicles



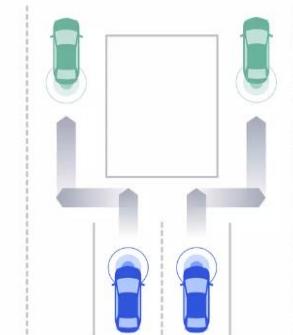
Advanced path planning

Supporting the level of predictability needed for advanced path planning for autonomous driving



Coordinated driving

Autonomous vehicles are able to choose time-efficient paths toward their given destinations as they know the planned movements of other vehicles



Communication Technologies

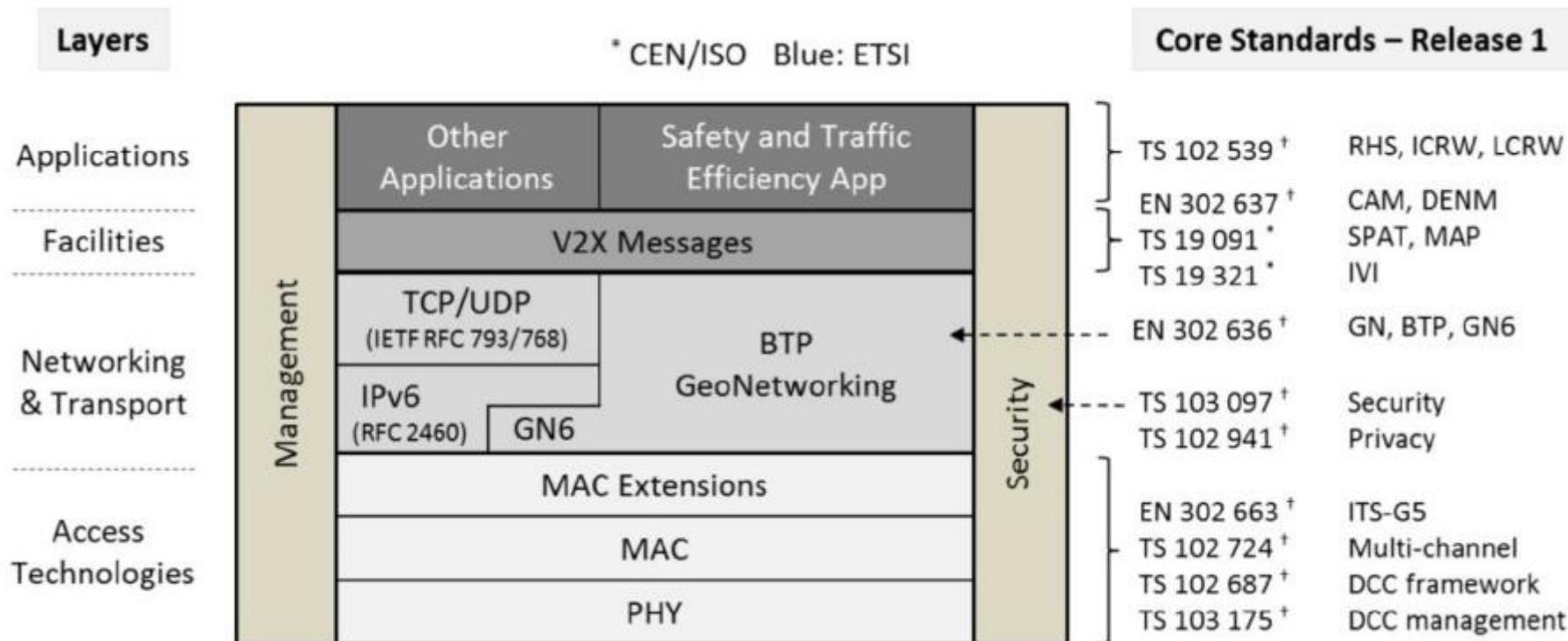
Critical Requirements for the Communication Technologies

- Range (>200-400m)
- Delays (<10msec)
- Time for communication when in range (<10-20msec)
- Bandwidth (>10Mb/sec.... → as much as possible)

ITS-G5 (DSRC, IEEE 802.11p)

- Developed for vehicle-to-vehicle communication (it also supports vehicle-to-infrastructure communication)
- Based on IEEE 802.11a with PHY and MAC extensions (based on CSMA/CA: Carrier Sense Multiple Access/Collision Avoidance)
- Adapted for latency-critical V2X communications in the 5.9 GHz band
- Frequency: Operates in the 75 MHz band of 5.9 GHz spectrum
- Range: in LoS it can go up to 1Km
 - Prone to obstructions: buildings, trees, cars
- Delay: < 10msec
- Time for communication when in range (10-20msec)
- Rate up to 27Mb/sec in the largest mode (usual it is 12Mb/sec)

ITS-G5



ITS-G5 Challenges

- Vehicle safety communication applications rely heavily on periodic broadcast of basic safety messages (BSM) which contain the positions, velocities, and other information about the vehicles.
- These messages with the PHY layer overheads typically measure around 300 bytes with the full security certificate header, and are expected to be transmitted up to once every 100 ms. The periodicity is chosen to meet latency and accuracy requirements of vehicle safety applications.
- Channel congestion in dense vehicular environments (packet collisions)
- Lack of handshake/ACK in delivering broadcast frames
- No QoS support
- Next generation of the specification, namely 802.11bd, just released (<https://standards.ieee.org/ieee/802.11bd/7451/>)

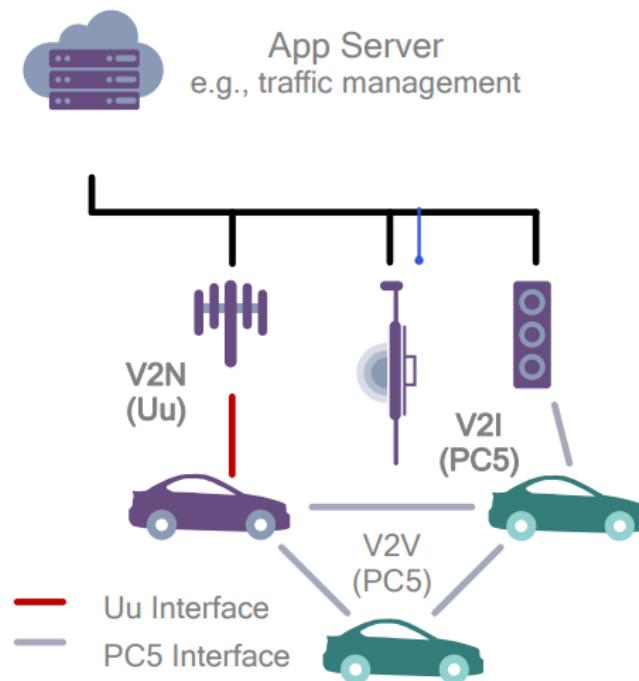
⁴⁰ Cellular-V2X (LTE-based 3GPP Rel 14)

- Based on 3GPP Rel 14
- Frequency: 5.9GHz
- Range: in LoS, it can go up to 1Km
 - Prone to obstructions: buildings, trees, cars
- Delay: < 20msec
- Time for communication when in range (~100msec?)
- Rate up to 150Mb/sec in the largest mode

⁴¹

Cellular-V2X (LTE-based 3GPP Rel 14)

- Cellular-V2X defines a new air interface called PC5 for V2V, V2I communication.
- V2N is still over the legacy LTE Uu air interface and provides over the top cloud services.

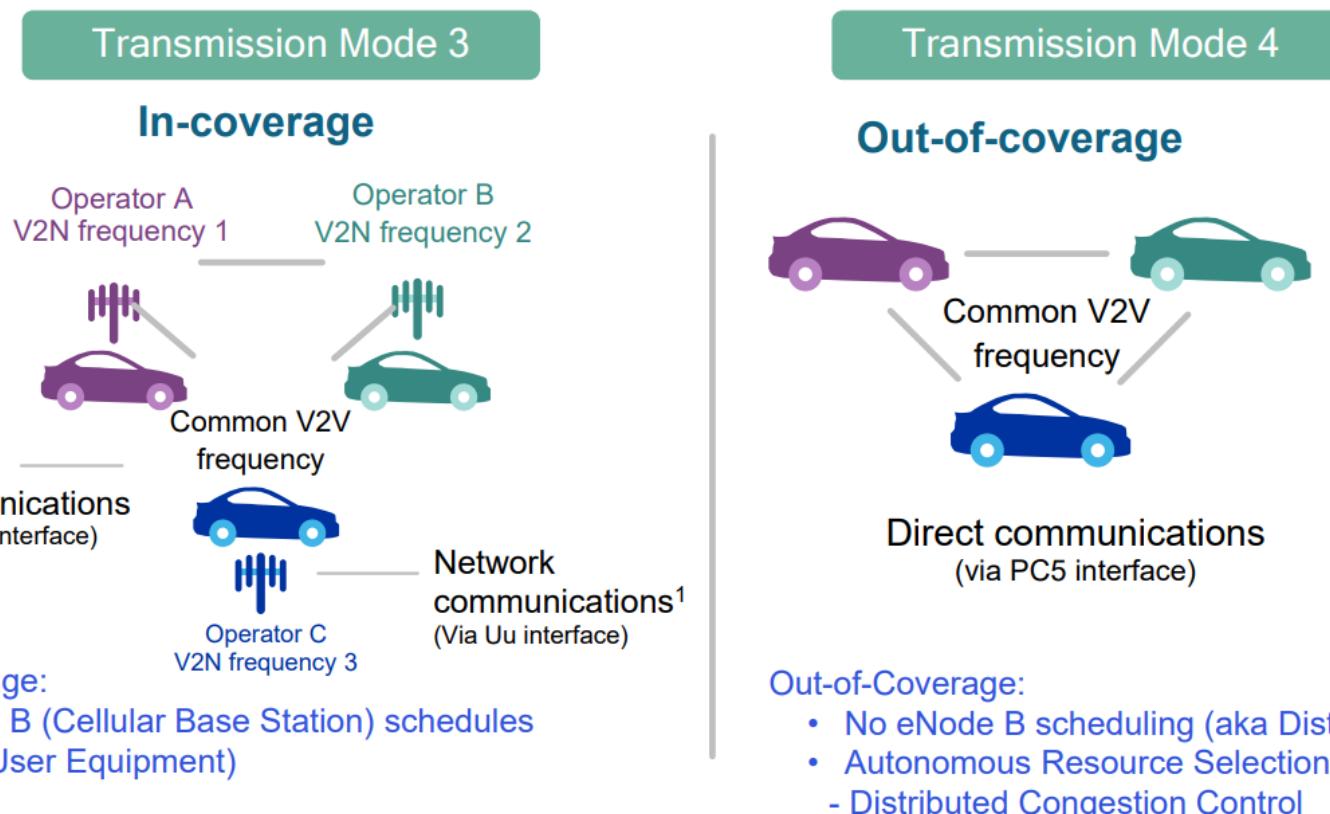


⁴² Cellular-V2X (LTE-based 3GPP Rel 14)

- C-V2X defines two Complementary Transmission Modes:
 - 1) Direct safety communication independent of cellular network
 - Low latency Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), and Vehicle-to-Person (V2P) operating in ITS bands (e.g., 5.9 GHz)
 - 2) Network communications for complementary services
 - Vehicle-to-Network (V2N) operates in the mobile operator's licensed spectrum
- Direct communications (V2V) via PC5 interface
 - Building upon LTE Direct device-to-device design with enhancements for high speeds / high Doppler, high density, improved synchronization, and low latency
 - Proximal direct communications (100s of meters)
 - Operates both in- and out-of-coverage
 - Latency-sensitive use cases, e.g., V2V safety
- Network communications (V2N) via Uu interface
 - Using LTE to broadcast messages from a V2X server to vehicles and beyond. Vehicles can send messages to the server via unicast
 - Wide area networks communications
 - Leverages existing LTE networks
 - More latency tolerant use cases, e.g., V2N situational awareness

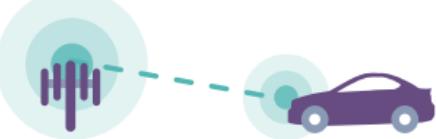
⁴³

Cellular-V2X (LTE-based 3GPP Rel 14)



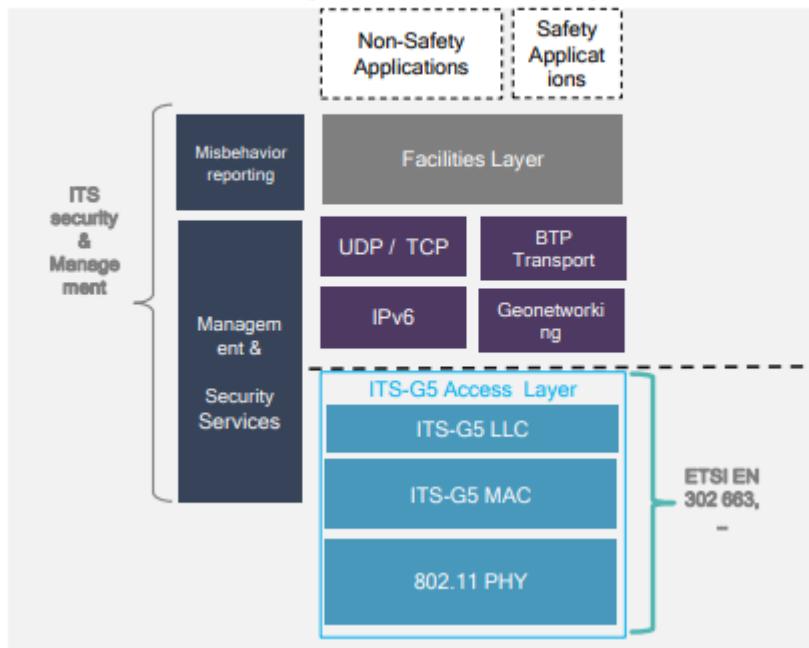
⁴⁴

Cellular-V2X (LTE-based 3GPP Rel 14)

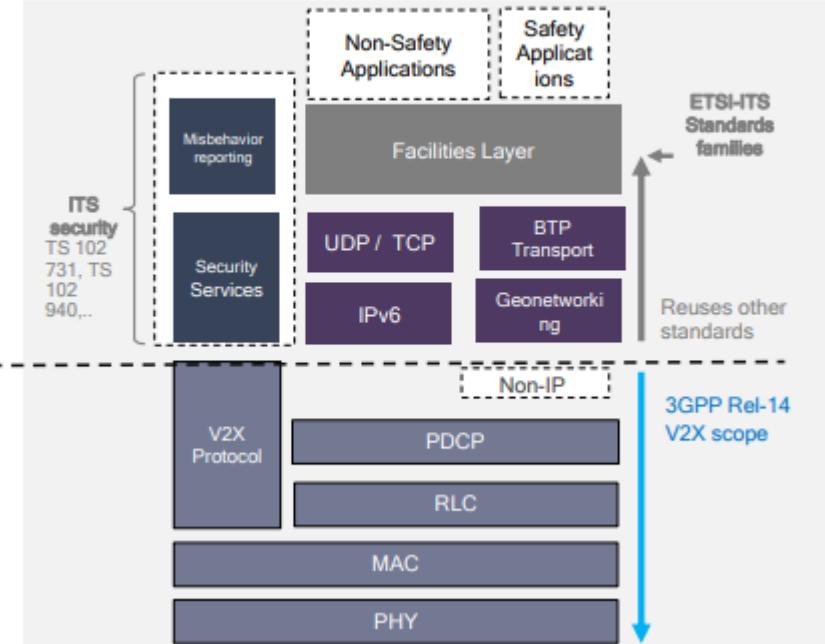
Connected Vehicle Challenges	C-V2X Solutions
	Improved signal design E.g. increasing # of ref signal symbols to improve synchronization and channel estimation
	Improved transmission structure Transmit control and data on the same sub-frame to reduce in-band emissions More efficient resource allocation New methods using sensing and semi-persistent resource selection
	Allow utilization of GPS timing Enhancements to use satellite (e.g., GNSS) when out-of-coverage

ITS-G5 vs C-V2X

ITS (ITS-G5 based)



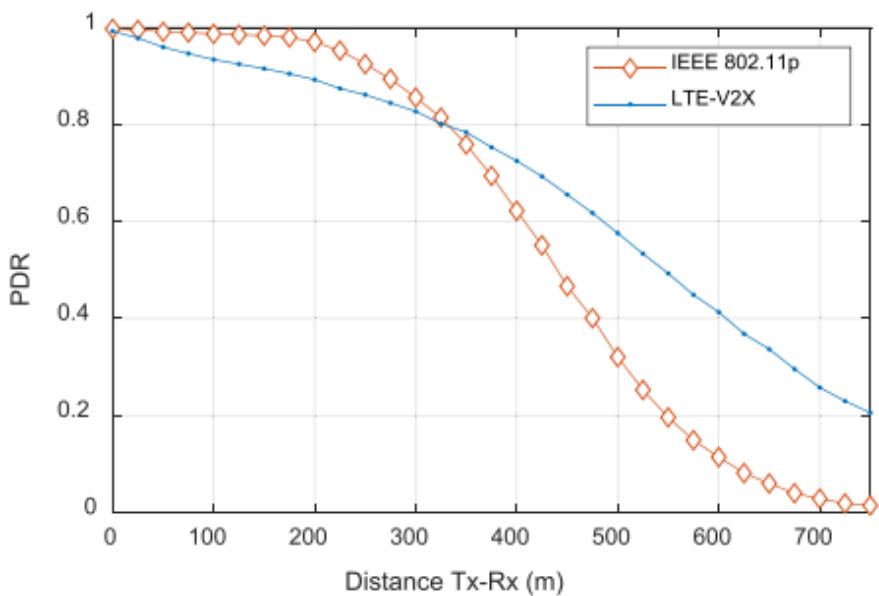
C-V2X (D2D/PC5 based)



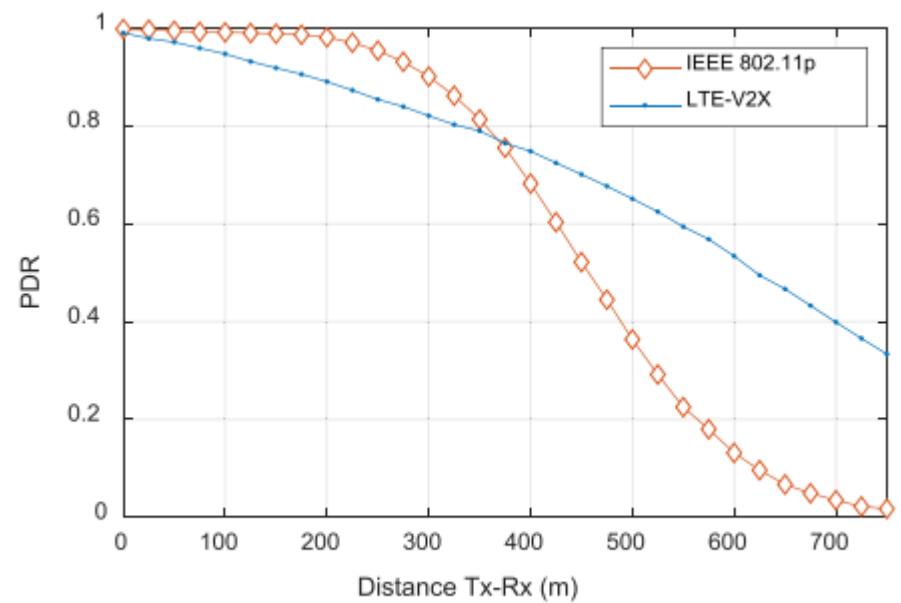
ITS-G5, C-V2X, 5G (Standalone)

Parameters	ITS-G5	C-V2X (LTE Rel. 14)	Future 5G SA
Currently available technology	Yes	Yes	No
Field trials (+10 years)	Yes	No	No
Applications	V2V, V2I	V2V, V2I, V2N	V2V, V2I, V2N
Latency	5 ms	20 ms	<5 ms
Data rate	3-27 Mbps	150 Mbps	10 Gbps
Multimedia and cloud services support	No	Yes	Yes

ITS-G5 vs C-V2X (Simulation)



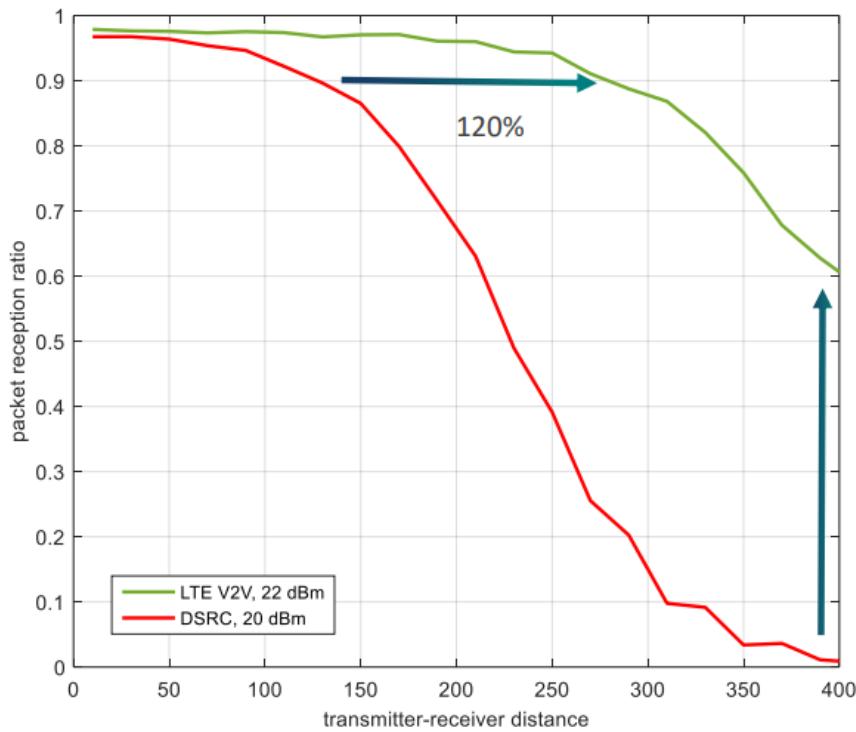
(a) *Empirical-size model. 120 veh/km (CBR~0.33)*



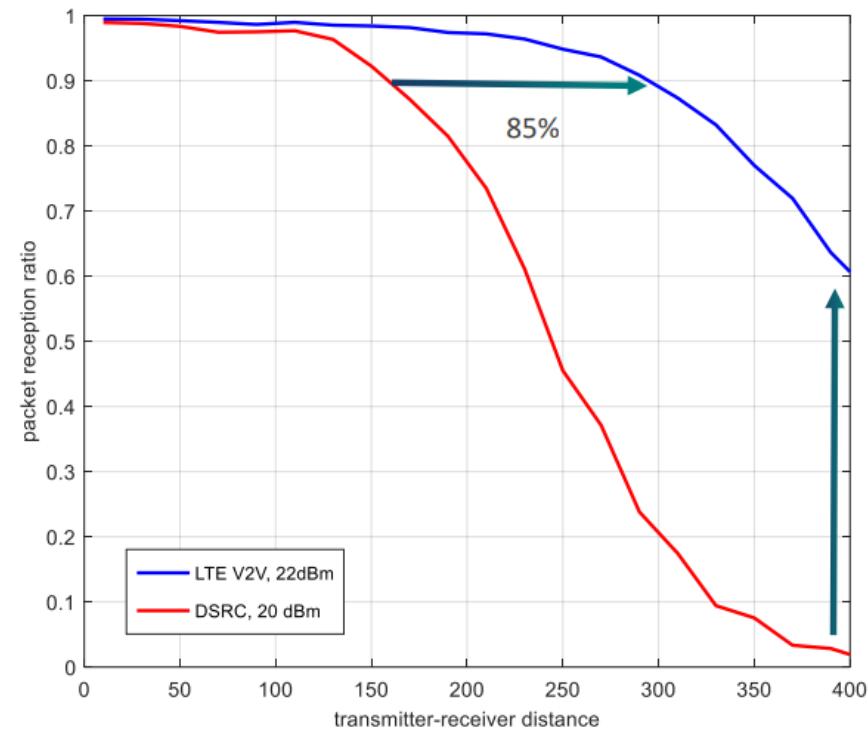
(b) *Empirical-time model. 200 veh/km (CBR~0.23)*

ITS-G5 vs C-V2X (Qualcomm)

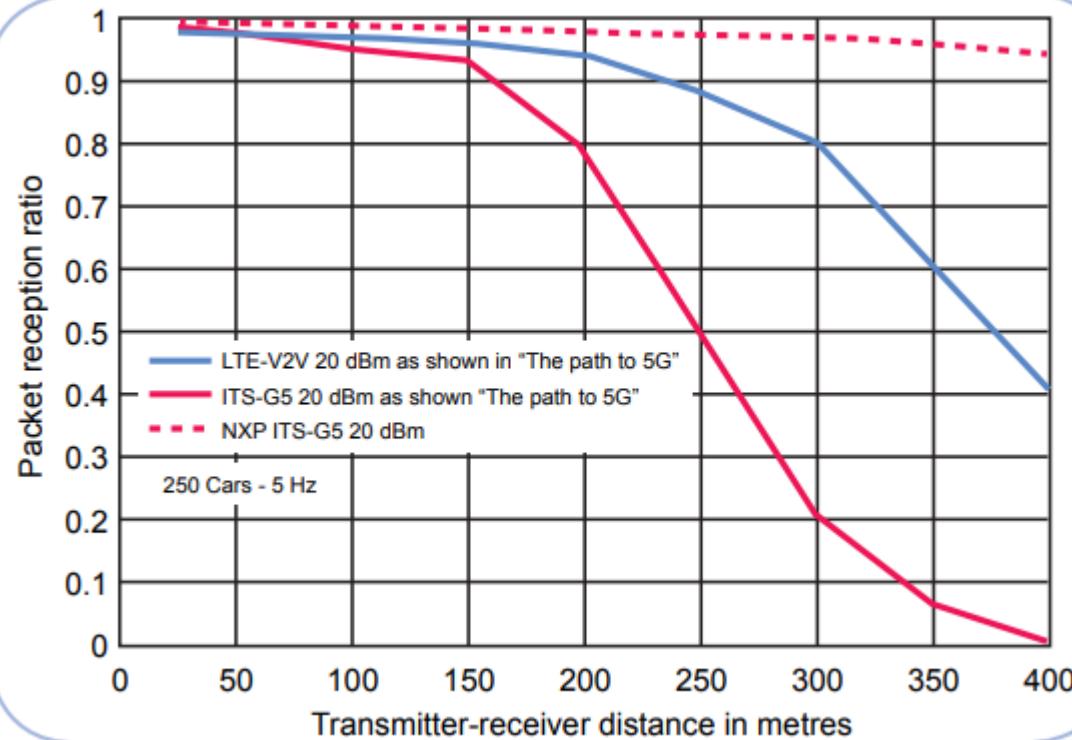
Freeway 250 km/hr, 69 cars



Freeway 140 km/hr, 123 cars



ITS-G5 vs C-V2X (NXP)



New radio Cellular-V2X (Rel 16)

NR Design

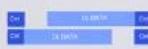
5G NR C-V2X capabilities for autonomous driving

Scalable OFDM-based air interface



5G C-V2X is expected to efficiently address diverse spectrum bands for different use cases. Leveraging wideband carrier support and OFDMA to deliver **higher data rates**.

Self-contained slot structure



Smaller slot structure with immediate feedback to enable **ultra reliable low latency communications**.

Advanced channel coding



State of the art LDPC/polar coding to deliver **higher reliability** with low complexity.

Wideband carrier support



Wideband carrier based **higher data rates and system capacity**.

Larger number of antenna



Efficiently utilize larger number of antennas than Rel-14 to deliver **higher data rate** and long range.

LDPC: Low-Density Parity Check

https://www.etsi.org/deliver/etsi_ts/102600_102699/10263702/01.02.01_60/ts_10263702v010201p.pdf

https://www.etsi.org/deliver/etsi_ts/102600_102699/10263703/01.01.01_60/ts_10263703v010101p.pdf

https://www.etsi.org/deliver/etsi_tr/103500_103599/103562/02.01.01_60/tr_103562v020101p.pdf

https://www.etsi.org/deliver/etsi_ts/103300_103399/10330003/02.01.01_60/ts_10330003v020101p.pdf

https://www.etsi.org/deliver/etsi_ts/103100_103199/10319103/01.01.01_60/ts_10319103v010101p.pdf

https://www.crow.nl/downloads/pdf/verkeer-en-vervoer/verkeersmanagement/landelijke-ivri-standaarden/d3046-2_spat-profile.aspx

https://www.crow.nl/downloads/pdf/verkeer-en-vervoer/verkeersmanagement/landelijke-ivri-standaarden/d3046-1_map-profile.aspx

QoS and Security

**Mestrado em Engenharia de
Computadores e Telemática**

2022/2023

Quality of Service

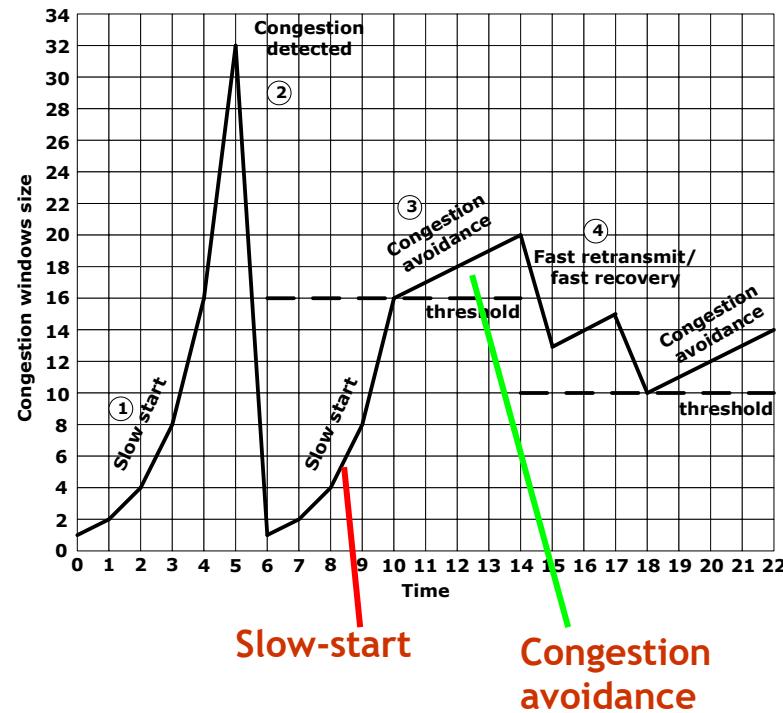
TCP- and UDP-based applications

Problem: Evaluate TCP

- Why does TCP perform *poorly* in ad-hoc/vehicular networks?
 - Developed for wire-line networks
 - Assumes all losses are due to congestion
- Many TCP variants have been proposed
 - How good are they? Are they sufficient?
- Are there any other alternatives?
 - Are non-TCP protocols the solution?

Overview of TCP concepts

- Conventional TCP: Tahoe, Reno, New-Reno
- Sending rate is controlled by
 - Congestion window (*cwnd*): limits the # of packets in flight
 - Slow-start threshold (*ssthresh*): when congestion avoidance starts
- Loss detection
 - 3 duplicate ACKs (faster, more efficient)
 - Retransmission timer expires (slower, less efficient)
- Overview of congestion control mechanisms
 - Slow-start phase: *cwnd* starts from 1 and increases exponentially
 - Congestion avoidance (CA): *cwnd* increases linearly
 - Fast retransmit and fast recovery: Triggered by 3 duplicate ACKs



What is different in wireless networks, and also in ad-hoc?

1. Mobility
 - Route stability and availability
2. High bit error rate
 - Packets can be lost due to “noise”
3. Unpredictability/Variability
 - Difficult to estimate time-out, RTT, bandwidth
4. Contention: packets compete for airtime
 - Intra-flow and inter-flow contentions
5. Long connections have poor performance
 - More than 4 hops throughput drops dramatically

Why does TCP fail in wireless and in ad-hoc networks?

- TCP misinterprets route failures as congestion
 - Effects: Reduce sending rate
 - Buffered packets (Data and ACKs) at intermediate nodes are dropped
 - Sender encounters timeouts
 - Under prolonged disconnection, a series of timeouts may be encountered
- TCP misinterprets wireless errors as congestion
 - Effects: Incorrect execution of congestion control → performance drops
 - Wireless channel is error-prone compared to wireline
 - Fading, interference, noise

Why does TCP fail in wireless and in ad-hoc networks?

- Delay spike causes TCP to invoke unnecessary retransmissions
 - Effects: Performance drops and many unnecessary retransmissions exist
 - Variability: Spikes are not uncommon here
 - Spikes throw off parameter estimation and tuning
 - RTO, window size, slow-start threshold
- Inefficiency due to the loss of retransmitted packet
 - Effects: performance drops significantly under high loss environment
 - Loosing a *retransmitted* packet hurts
 - TCP can recover from one loss (fast retransmission)
 - Wired networks: packet loss rate is low
 - Here, high packet loss makes the problem significant

New possibilities?

TCP-Cubic (1)

- CUBIC is RTT independent
- The window size is a cubic function of the time t , which is passed time since the last congestion occurrence

$$W_{CUBIC} = C \left(t - \sqrt[3]{\frac{\beta \cdot W_{max}}{C}} \right)^3 + W_{max}$$

- where W_{CUBIC} is the *cwnd* size for cubic congestion control mechanism,
- W_{max} , a *cwnd* size just before last window reduction,
- C , a predefined constant (scaling factor),
- β , a *cwnd* size decrease factor. Window size reduction at the time of loss event is

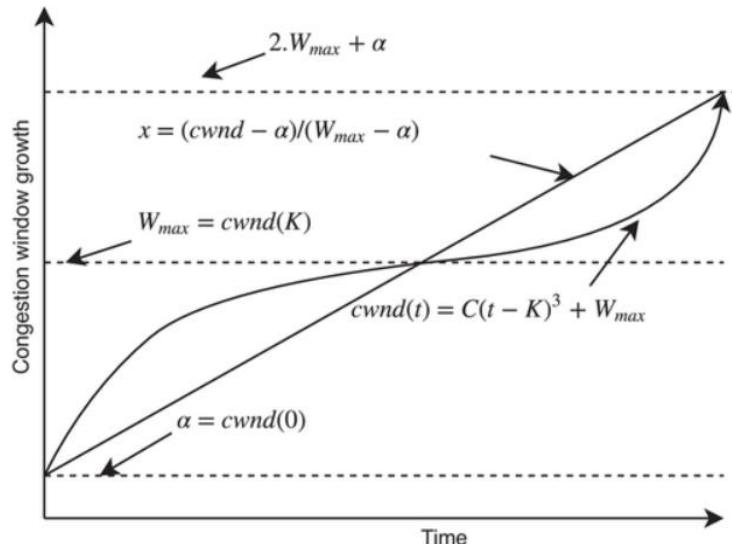
$$W(t) \leftarrow W(t^*) (1 - \beta),$$

- where $W(t^*)$ is the *cwnd* size at the time t^* of packet loss, that is, W_{max} .
- When a packet loss is detected, $W(t)$ is reduced as per the previous equation.

TCP-Cubic (2)

- Whenever a packet dropping event occurs, the cwnd is reduced by a factor β , otherwise increased by the α for a successful ACK. For TCP with CUBIC, $\beta=0.3$, whereas for QUIC with CUBIC, $\beta=0.3/2=0.15$

$$K = \sqrt[3]{\frac{\beta \cdot W_{max}}{C}}$$

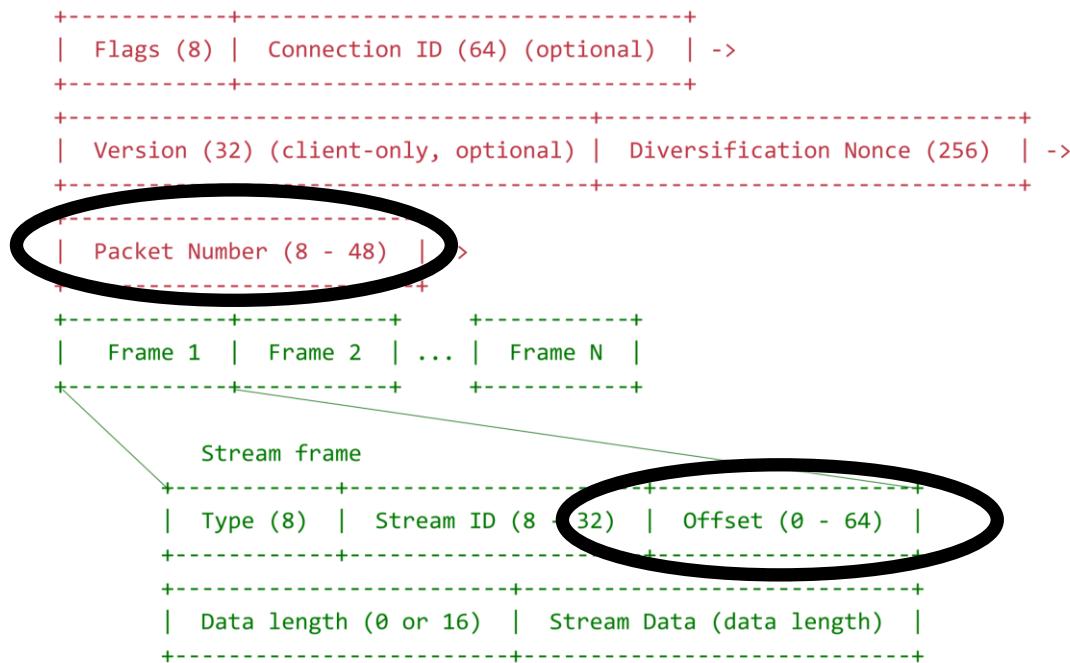


QUIC

- QUIC was developed and deployed by Google in 2013, but was presented as a standard in 2021 by RFC9000.
- QUIC packets are carried in UDP datagrams to better facilitate deployment in existing systems and networks.
- QUIC handshake combines negotiation of cryptographic (TLS) and transport parameters.
 - It is structured to allow the exchange of application data as soon as possible.
- Application protocols exchange information over a QUIC connection via streams which are ordered sequences of bytes. Two types of streams can be created:
 - Bidirectional streams, which allow both endpoints to send data.
 - Unidirectional streams, which allow a single endpoint to send data.
- Avoids head-of-line blocking across multiple streams
 - When a packet loss occurs, only streams with data in that packet are blocked waiting for a retransmission to be received, while other streams can continue making progress.
- Two levels of data flow control in QUIC:
 - Stream flow control, which prevents a single stream from consuming the entire receive buffer for a connection by limiting the amount of data that can be sent on each stream.
 - Connection flow control, which prevents senders from exceeding a receiver's buffer capacity for the connection by limiting the total bytes of stream data sent on all streams.

QUIC: RTT estimation

- Use **separate sequence numbers** for data and packet delivery
- Decouple reliability completely from ordered delivery



QUIC: RTT estimation

- Decouple reliability completely from ordered delivery
- Packet numbers are **monotonically increasing** (and hence unique)
 - Distinct from frame offsets within stream
- Use time between the transmission of a packet and its ACK (identifiable using unique packet number) for RTT estimate
 - Regardless of original transmission or retransmission
- Packet loss detected through lack of ACK packets
- However, the congestion control is independent of RTT
 - One of the most used algorithms is CUBIC
- Congestion control does not depend on the delays of the wireless medium
- Congestion control is applied upon detection of packet loss
 - Again, packet loss may not be a sign of congestion

Classification of Transport protocols

- How to make the transport protocol suited to wireless and ad-hoc networks?
- TCP variants try to improve the performance by the following ways
 - Estimating the available bandwidth
 - Exploiting buffering capability

TCP-Vegas

- TCP Vegas senses the congestion in the network before any packet loss occurs, and instantly it decreases the window size. So, TCP Vegas handles the congestion without any packet loss occur.
- Uses $diff = expected\ rate - actual\ rate$ to regulate the sending rate
- $Expected\ rate = \text{window size} / \text{baseRTT}$,
- $Actual\ rate = \text{window size} / \text{currentRTT}$
- baseRTT = It is the minimum RTT measured so far.
- $Expected\ rate$ is always larger then the $Actual\ rate$

TCP-Vegas

- Rate-based congestion control
 - $\text{diff} = \text{expected rate} - \text{actual rate}$
 - If $\text{diff} < a$, Vegas increases $cwnd$ linearly
 - If $\text{diff} > b$, Vegas decreases $cwnd$ linearly
 - If $a < \text{diff} < b$, Vegas keeps $cwnd$ unchanged
- Modified slow-start
 - Allows $cwnd$ to grow exponentially only in every other RTT
 - If $\text{diff} > c$, Vegas switches from slow-start to congestion avoidance
- New retransmission
 - Reads and records transmission time
 - When DUPACK arrives, checks if it is expired
 - Retransmits without waiting for third DUPACK

Exploiting Buffering Capabilities

- Buffering capability and Sequence information (TCP-BuS)
 - Uses explicit route failure notification (routing error) to detect route failures
 - When route failure occurs, intermediate nodes buffer the pending packets and TCP sender doubles the retransmission timeout (*RTO*) value
 - Makes use of special messages such as localized query and reply to find a path
 - Messages modified to carry TCP connection and segment information
 - Avoid timeouts and unnecessary retransmissions
 - Pro: Reduce the number of timeout events → reduce the number of retransmissions
 - Con: Require assistance from intermediate nodes
 - Special routing protocol is used

QoS in UDP: trade-offs

- In IntServ-like mechanisms
 - Application specifies traffic and QoS parameters
 - A resource reservation protocol estimates and reserves sufficient resources at each node on the path
- In a multi-hop wireless network, however
 - It is hard to estimate available resources
 - Shared medium
 - All traffic in the transmission range reduces available bandwidth
 - Dynamic bandwidth due to node mobility and contention
 - It is hard to do resource reservation
 - Shared medium reservation requires global coordination
 - Violations can occur as bandwidth fluctuates
 - Resource reservation is pinned to a route
 - Must be redone whenever route changes
 - This might NOT be a good idea
 - Time for recovering a route and reserve resources in the new route

QoS in UDP: trade-offs

- In DiffServ-like mechanisms
 - Application chooses a class of service
 - Network needs admission control to avoid class overload
 - Examples
 - Assured Forwarding Per-hop behavior assures per-hop throughput
 - Expedited Forwarding Per-hop behavior assures per-hop low delay
- In a multi-hop wireless network, however
 - It is hard to do admission control
 - Flows do not go through common ingress nodes
 - It is hard to maintain assurances
 - Flow distribution varies as routes change
 - Bandwidth fluctuates

QoS Routing

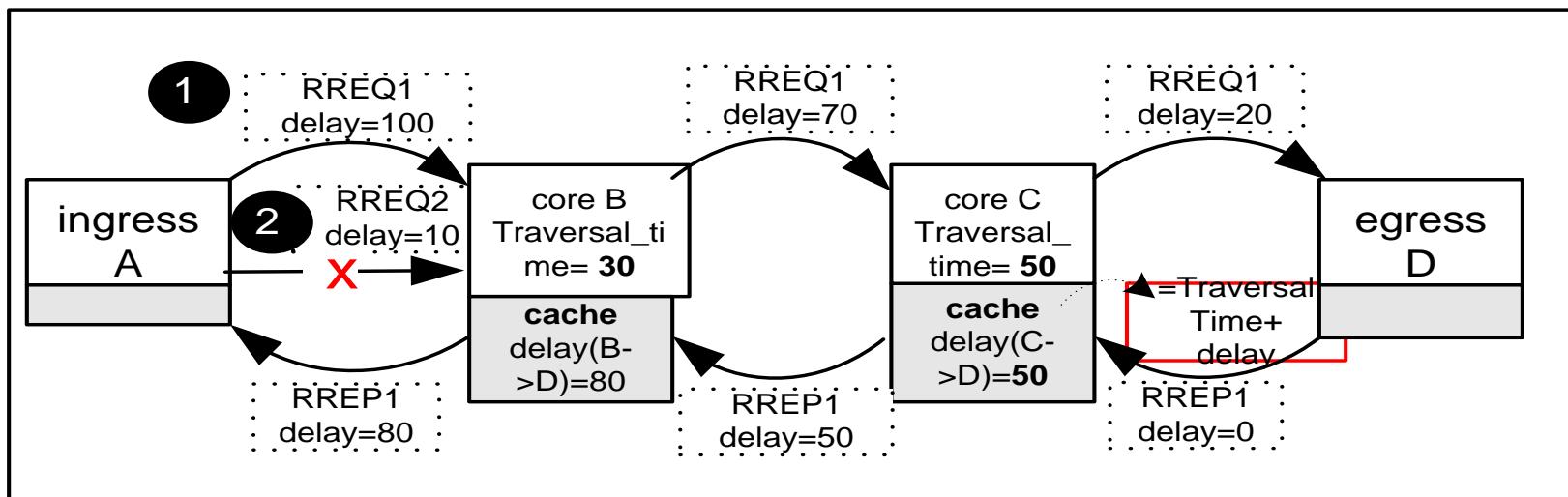
- Routing is an essential component for QoS
- It can inform a source node of the bandwidth and QoS availability of a destination node and of the path to the destination node
- Add QoS requirements in routing metrics
 - Difficult to route maintenance
 - Overhead of QoS routing
 - Reserved resources may not be guaranteed
 - Needs to be responsive to nodes mobility

QoS for AODV

- Add extensions to the route messages (RREQ, RREP)
- Node that receives a RREQ + QoS extension must be able to meet the service requirement in order to rebroadcast the RREQ (if not in cache)
- In order to handle the QoS extensions, some changes need to be done on the routing tables
- AODV current fields
 - Destination sequence number, Interface, Hop count, Next hop, List of precursors
- AODV new fields (4 new fields)
 - Maximum delay
 - Minimum available bandwidth
 - List of sources requesting delay guarantees
 - List of sources requesting bandwidth guarantees

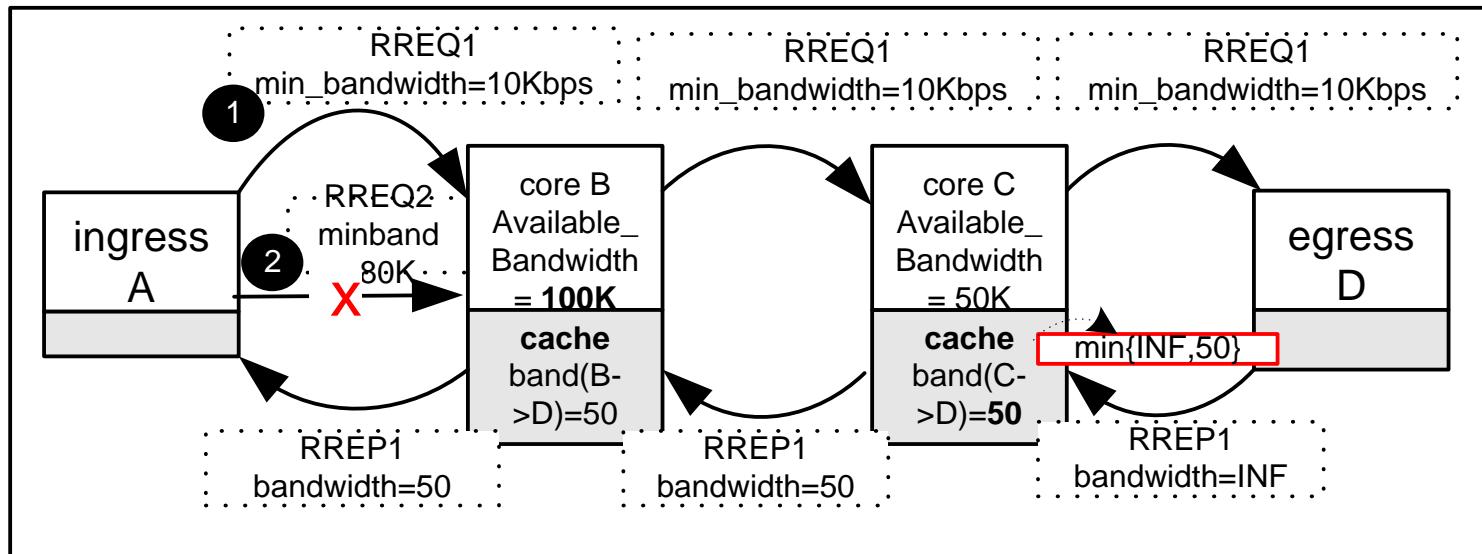
QoS for AODV- Delay

- Handling Delay
 - Maximum delay extension
 - List of sources requesting delay guarantees
 - Used during route discovery process



QoS for AODV- Bandwidth

- Handling Bandwidth
 - Similar to delay requests
 - RREQ can include both types
 - Used during route discovery process



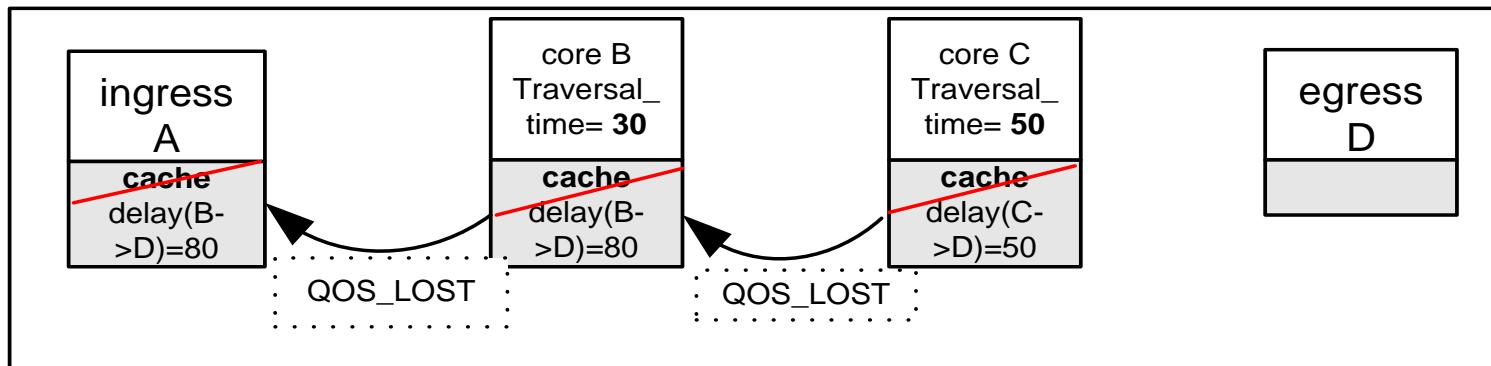
QoS for AODV- Loosing QoS

Loosing QoS parameters

- If, after establishment, a node detects that the QoS cannot be maintained any more, it originates an ICMP QoS_LOST message to all depending nodes
 - Reason to keep a List of Sources requesting delay/bandwidth guarantees

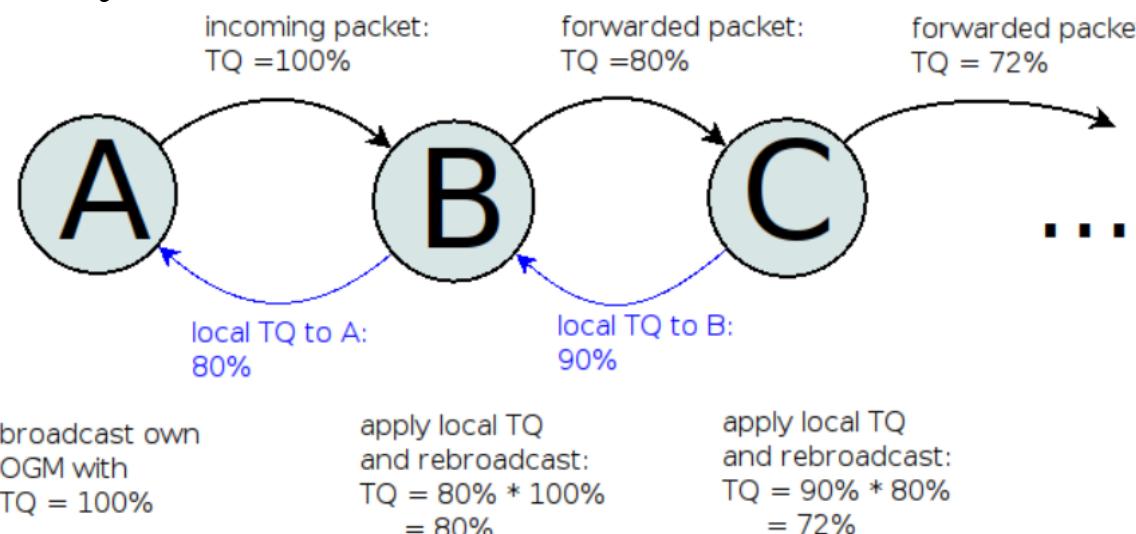
Reasons for loosing QoS parameters

- Increased load of a node
- Why would a node take over more jobs than it can handle?



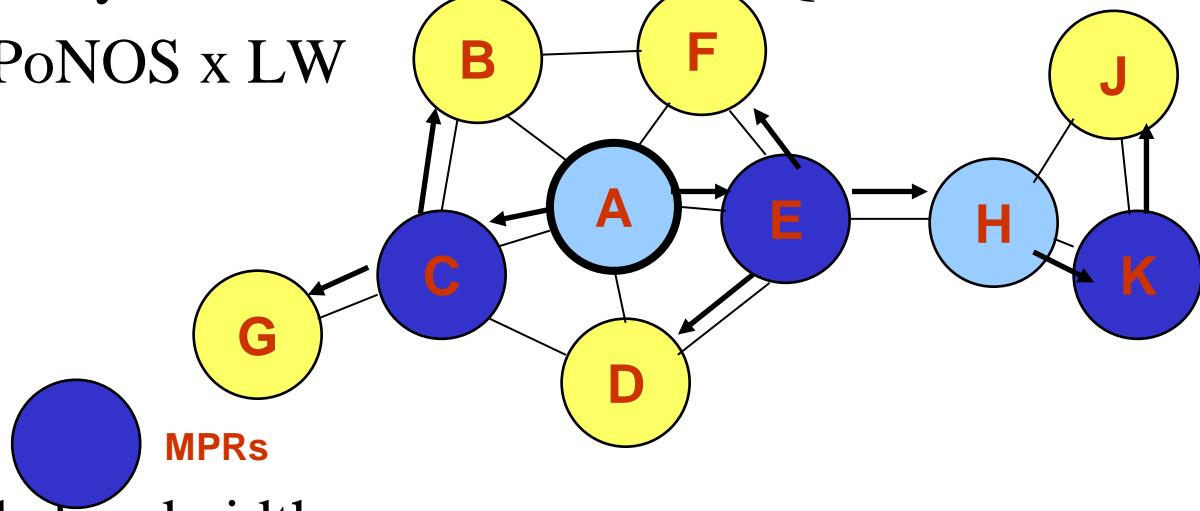
Transmission Quality (Batman v.3)

- To add the local link quality in the TQ value the following calculation is performed:
- $TQ = TQ_{\text{incoming}} * TQ_{\text{local}}$
- Example: Node A broadcasts the packet with TQ max. Node B receives it, applies the TQ calculation and rebroadcasts it. When node C gets the packet it knows about the transmit quality towards node A.



QoS-OLSR

- Multi-Point relays are the nodes with best QoS
- $\text{QoS} = \text{AB} \times \text{PoNOS} \times \text{LW}$



- AB: available bandwidth
- PoNOS: Percentage of Neighbors on Other Street, presents the level of neighbors' diversity.
- LW: Lane Weight is used to favor the selection of MPRs from lanes that carry the majority of the traffic flow to increase their stability.

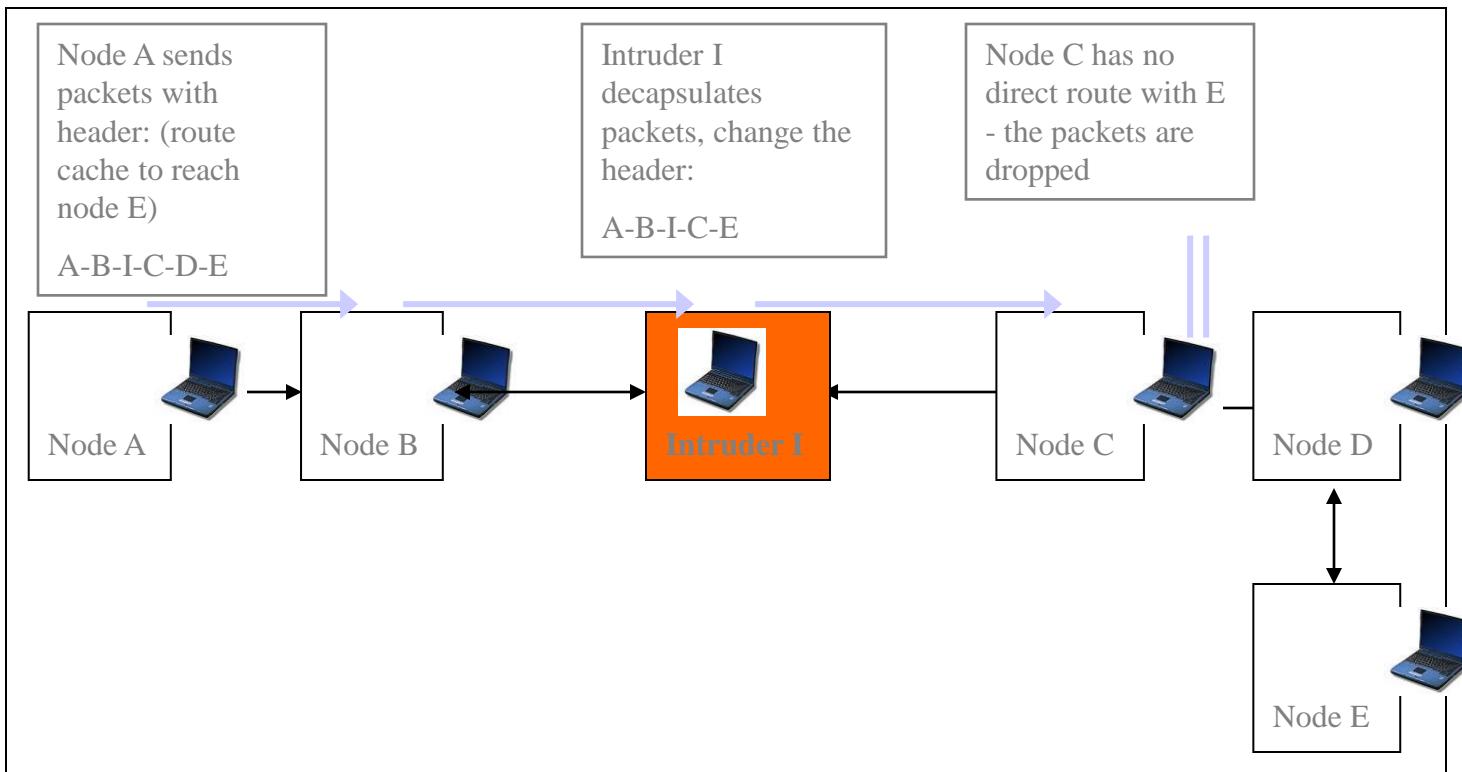
Security

Possible routing attacks: attacks using modification

- Malicious node announces better routes than the other nodes in order to be inserted in the network
 - Redirection by changing the route sequence number
 - Redirection with modified hop count
 - Denial Of Service (DOS) attacks with modified source routes
 - A malicious node is inserted in the network through one of the previous techniques
 - The malicious node changes the packet headers it receives
 - The packets will not reach the destination
 - The transmission is aborted

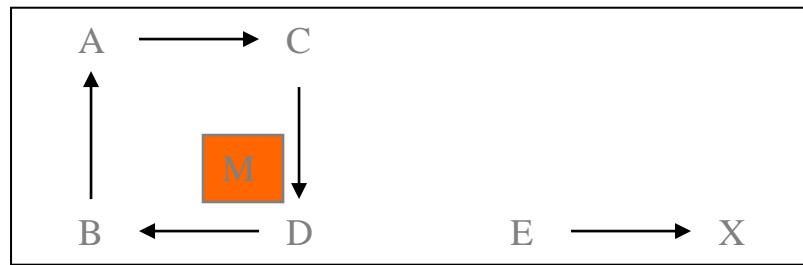
Possible routing attacks: attacks using modification

- DOS attacks with modified source routes



Possible routing attacks: attacks using impersonation

- Usurpation of the identity of another node to perform changes
 - Spoofing MAC address of other nodes
 - Forming loops by spoofing MAC address
 - A malicious node M can listen to all nodes
 - It changes its MAC address to the MAC address of another node
 - It announces to several nodes a shorter path to reach X



- X is now unreachable because of the loop formed

Possible routing attacks: attacks using fabrication

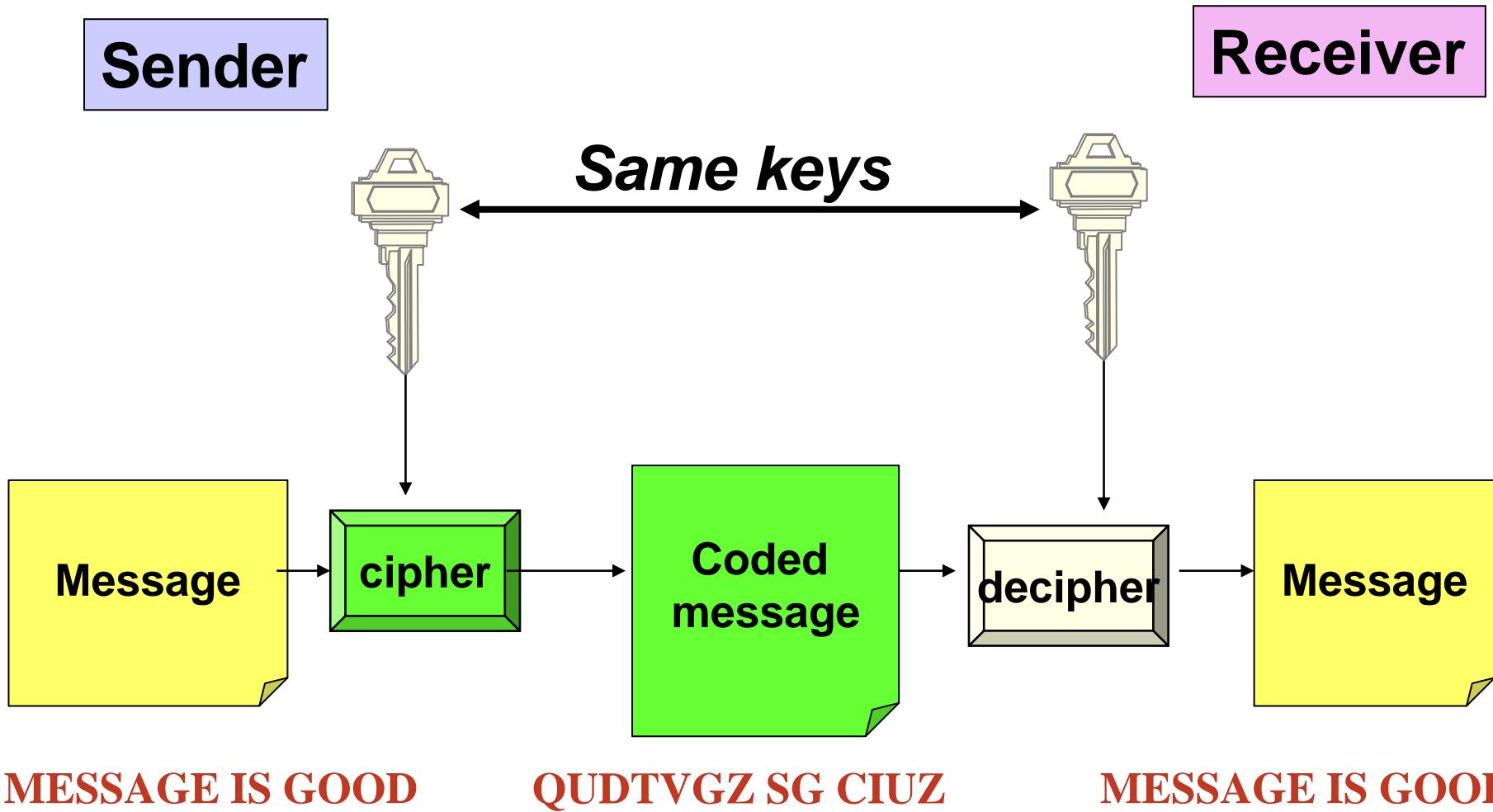
- Generates traffic to disturb the good operation of an ad-hoc network
 - Falsifying route error messages
 - Isolate nodes
 - Corrupting routing state
 - Hacker can easily broadcast a message with a spoofed IP address such that the other nodes add this new route to reach a special node S
 - The malicious node will receive the packets intended to S
 - Routing table overflow attack
 - Hacker can send in the network a lot of routes to non-existent nodes until overwhelm the protocol
 - Replay attack
 - Hacker sends old advertisements to a node
 - Black hole attack
 - Hacker advertises a zero metric route for all destinations
 - All the nodes around it will route packets towards it

Key Management - basics

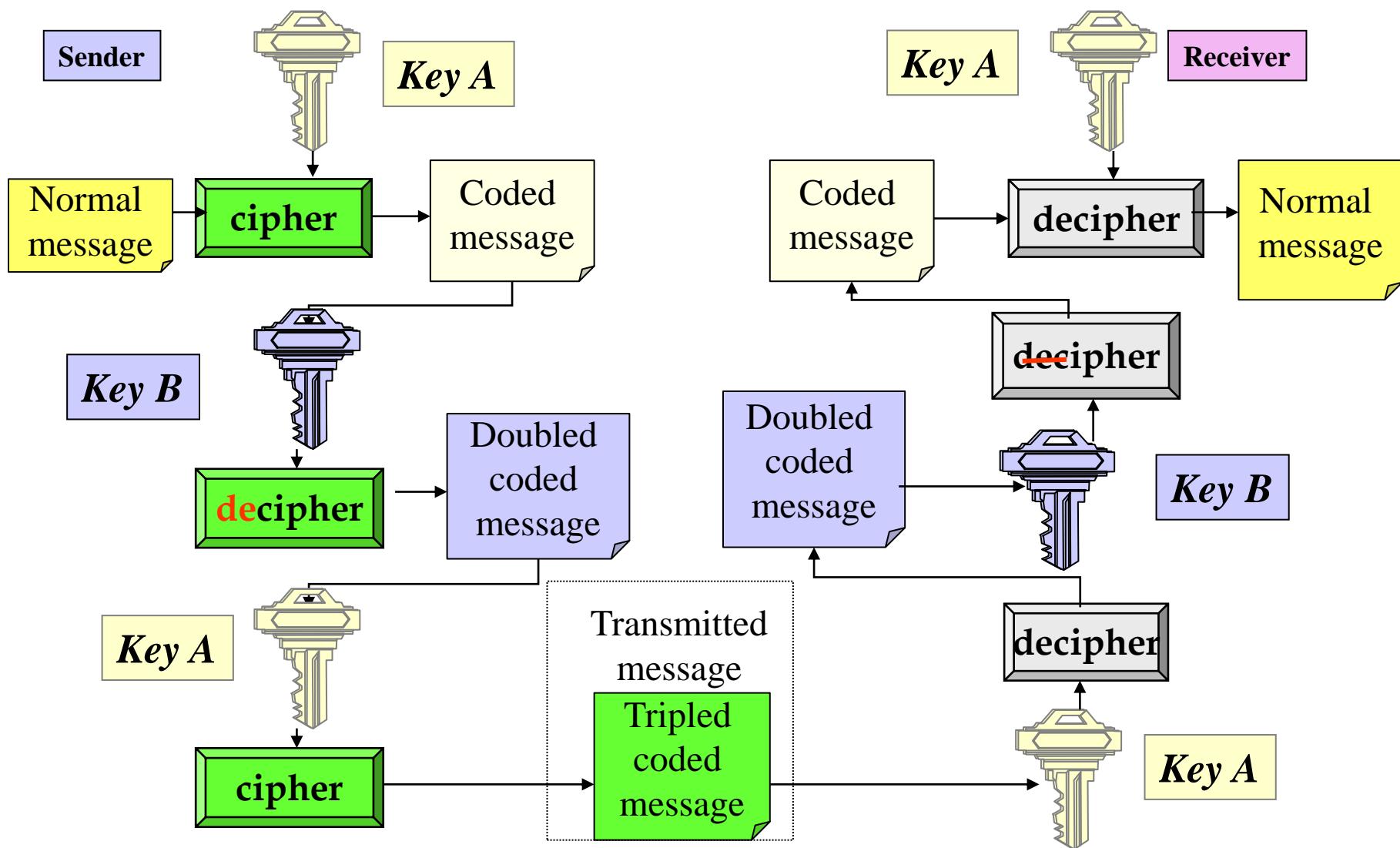
Symmetric cipher

- Advantages
 - Fast and relatively secure
 - Provides integrity and privacy
 - Larger key length provides larger security
- Disadvantages
 - Requires the share of a secret key
 - How?
 - Complex administration and non-scalable
 - It is needed to distribute the keys
 - A key for each receiver

Symmetric cipher



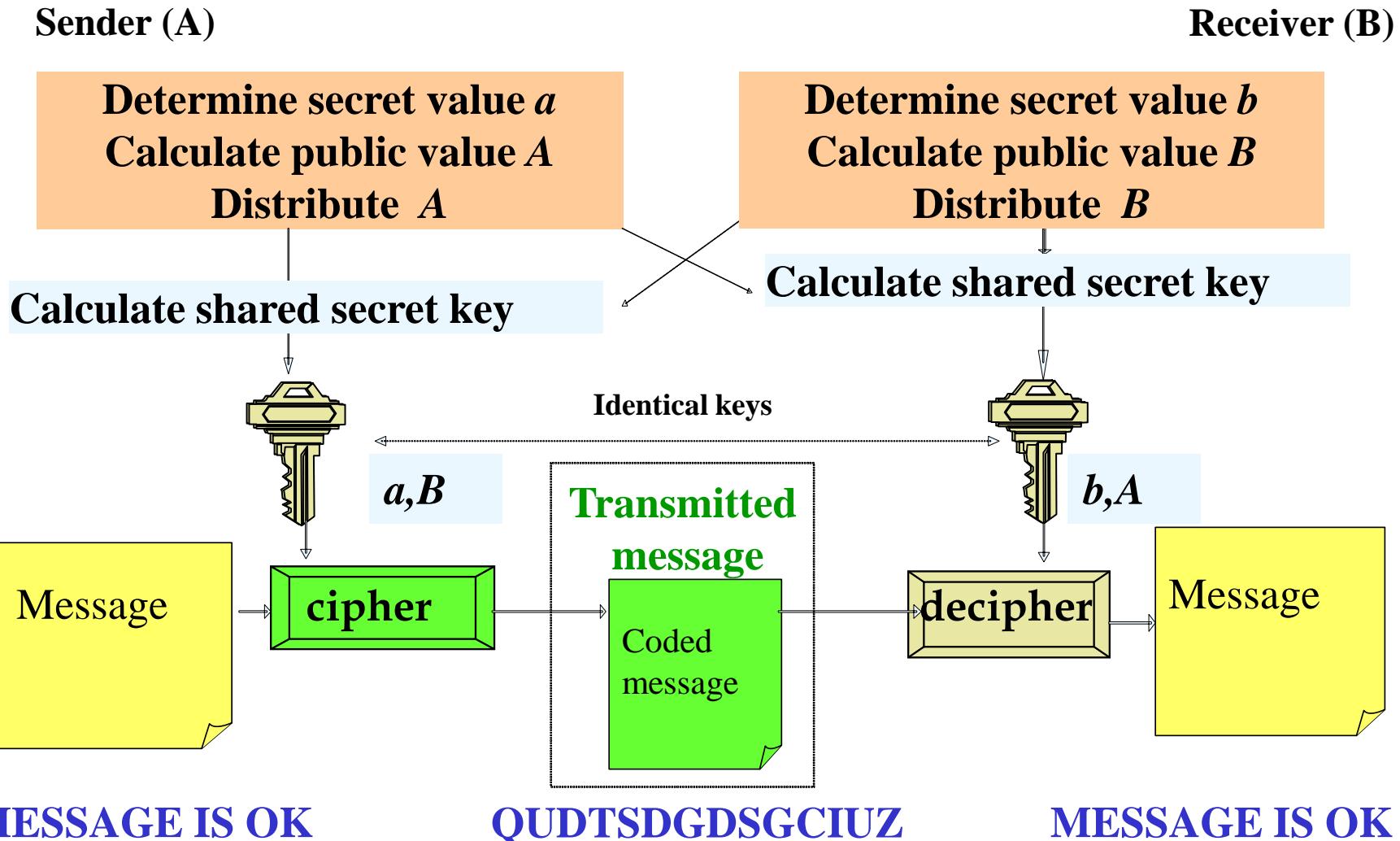
Triple symmetric mechanisms (e.g. “3-DES”)



Asymmetric cipher

- Also known as PKE - public key encryption
- Advantages
 - It is not needed to share secret keys *à priori*
 - It is scalable and versatile
- Disadvantages
 - Generally computationally intensive
 - It may require a certificate authority
 - Private keys have to be confidential

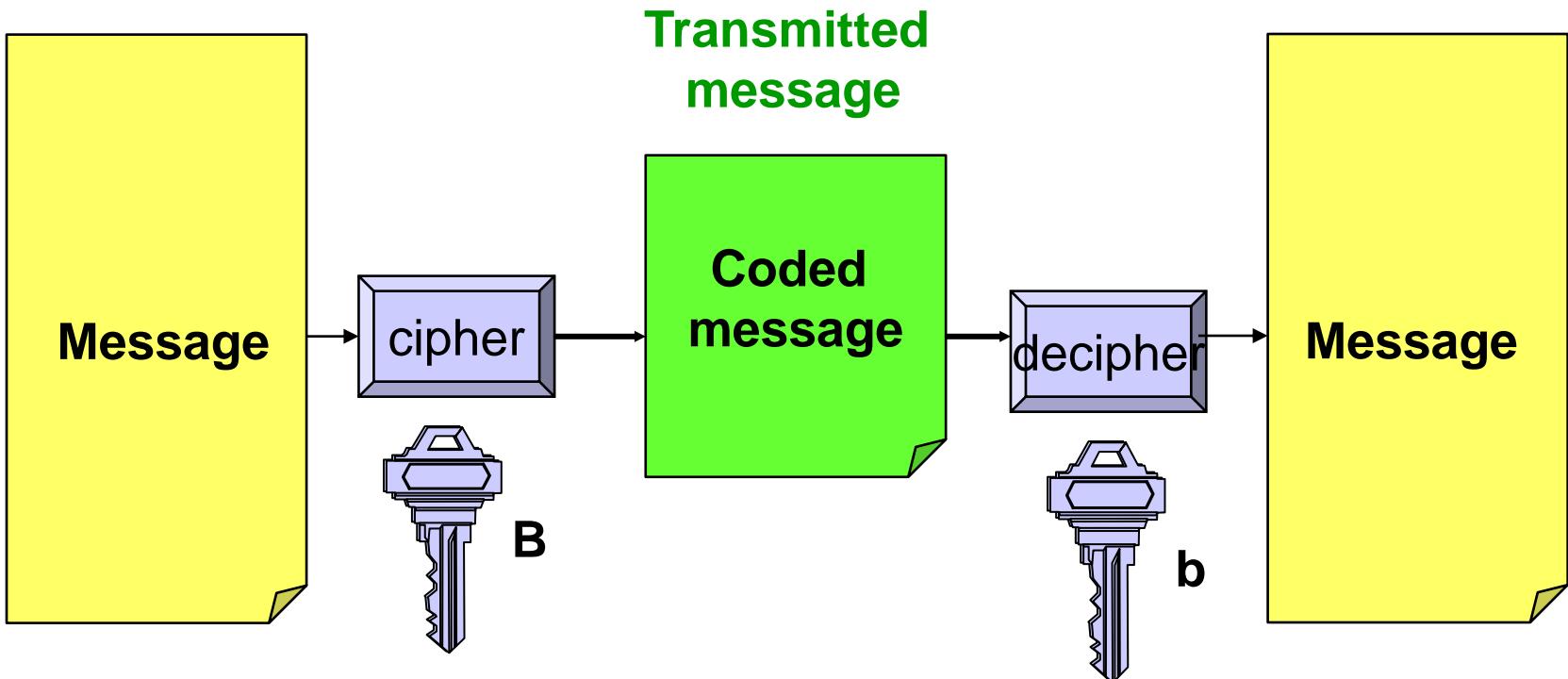
Diffie-Hellman



Public-private pair for confidentiality

Sender (A,a)

Receiver (B,b)

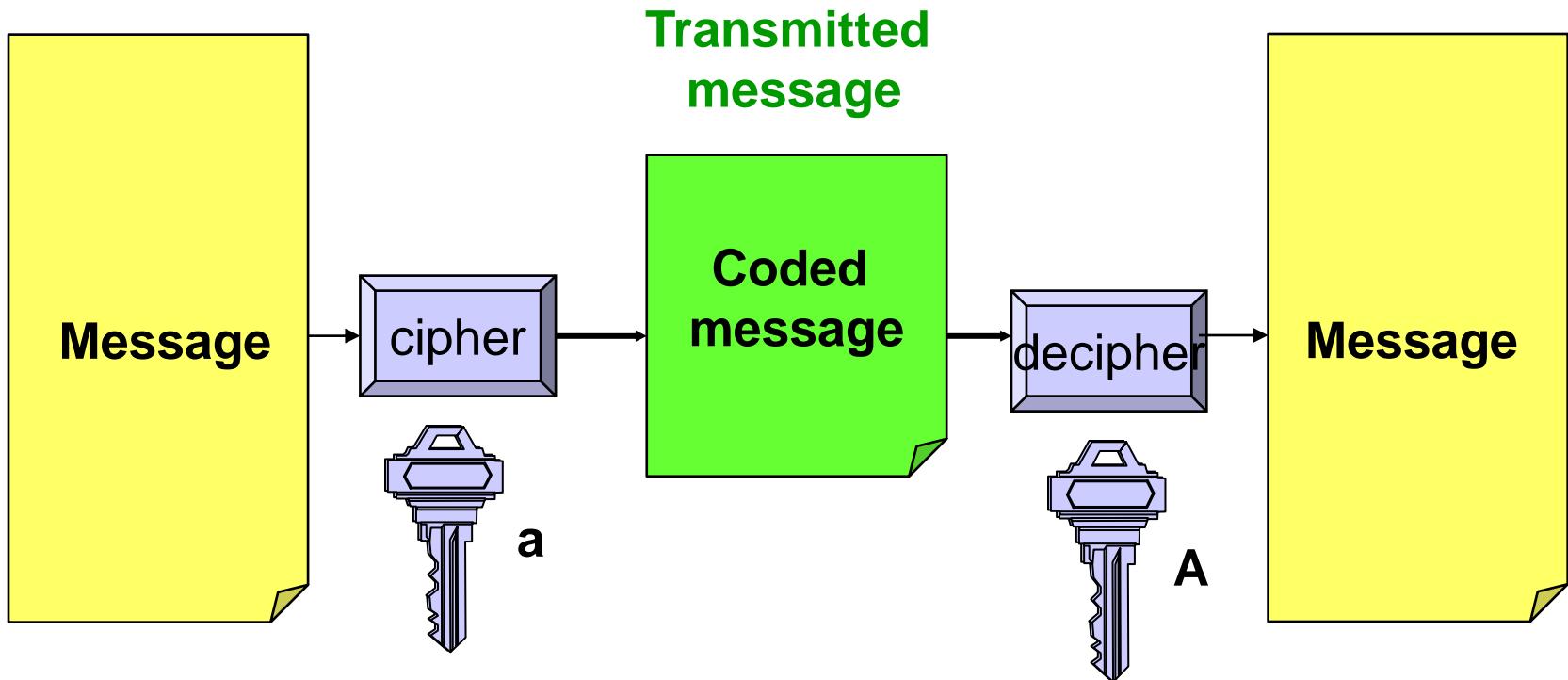


Guarantees confidentiality without sender guarantee

Public-private pair for authentication

Sender (A,a)

Receiver (B,b)

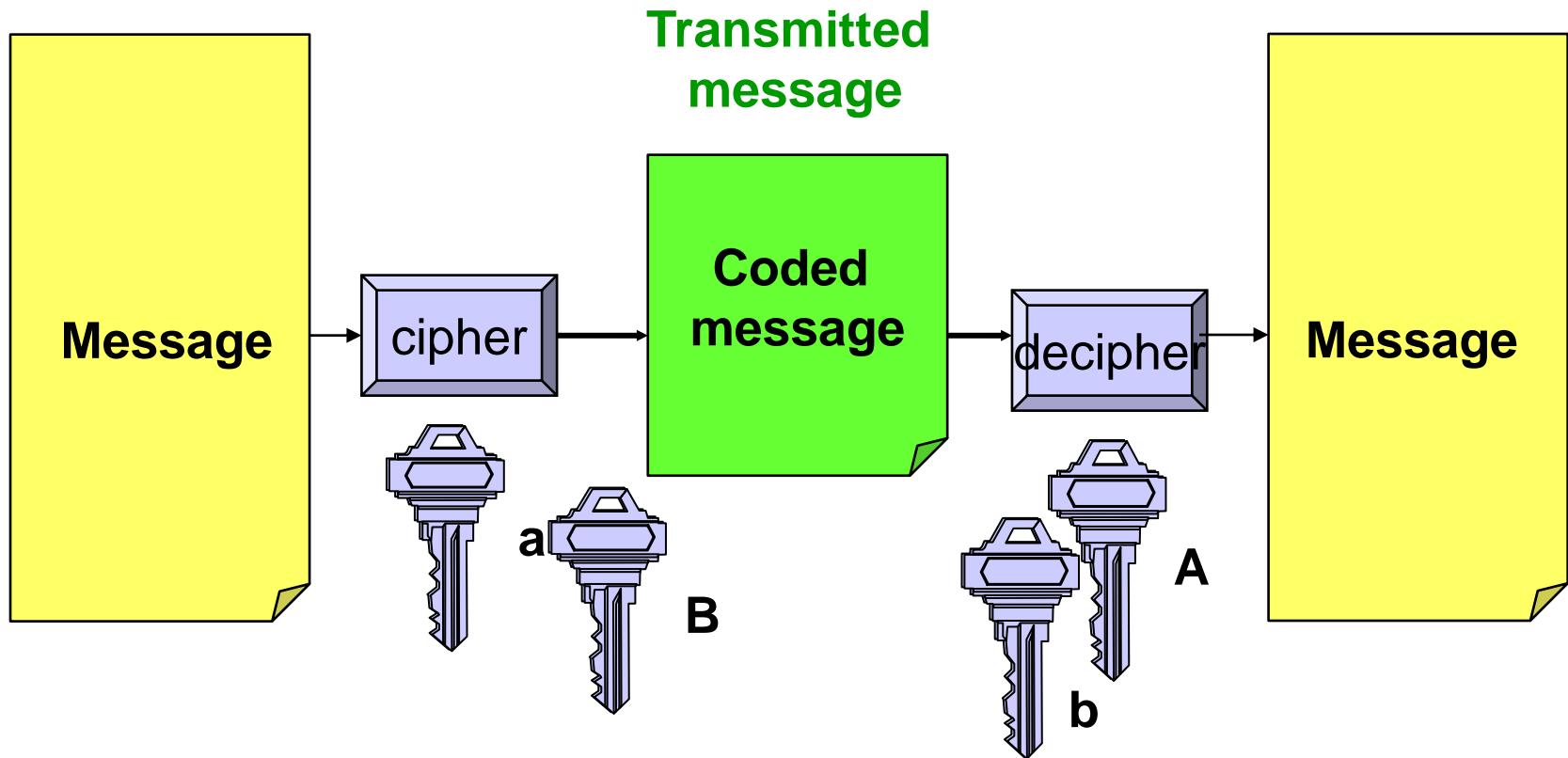


Authenticates the sender, since only the sender will have the secret key a , corresponding to the public key A

Public-private pair for confidentiality and authentication

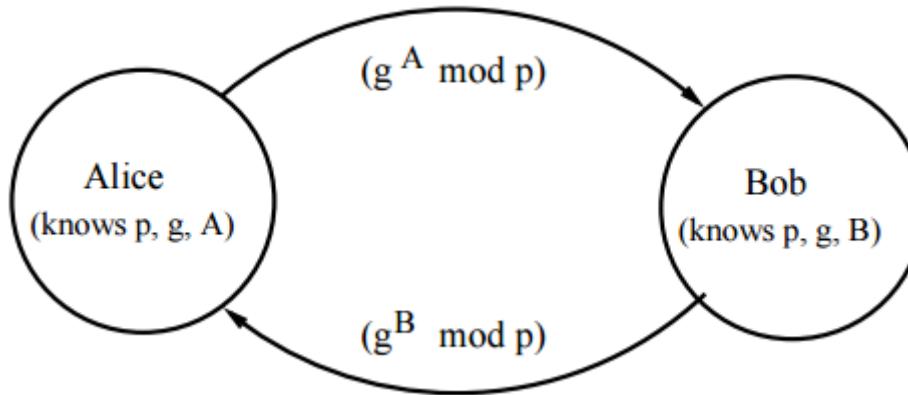
Sender (A,a)

Receiver (B,b)



Guarantees confidentiality and authenticates the sender, since only the sender will have the secret key a, corresponding to the public key A

Diffie-Hellman



- Alice and Bob agree on a prime number **p** and a base **g**.
- Alice chooses the secret number **a**, and sends to Bob $(g^a \bmod p)$.
- Bob chooses the secret number **b**, and sends to Alice $(g^b \bmod p)$.
- Alice calculates $((g^b \bmod p)^a \bmod p)$.
- Bob calculates $((g^a \bmod p)^b \bmod p)$.

- Alice and Bob use this value as their session key. **p** and **g** do not have to be protected.

Example Diffie-Hellman

- Alice and Bob choose $p = 23$ e $g = 5$.
- Alice chooses $a = 6$ and sends $5^6 \text{ mod } 23 = 8$.
- Bob chooses $b = 15$ and sends $5^{15} \text{ mod } 23 = 19$.
- Alice calculates $19^6 \text{ mod } 23 = 2$.
- Bob calculates $8^{15} \text{ mod } 23 = 2$.
- 2 is the shared key.

RSA (Rivest-Shamir-Adleman) Key

- Each user generates the pair public/private key
- Randomly generates 2 large prime numbers - p, q
- Calculates the modulus of the system $N=p \cdot q$
 - $\phi(N)=(p-1)(q-1)$
- Selects a random key e
 - where $1 < e < \phi(N)$, $\gcd(e, \phi(N)) = 1$ gcd = greatest common divisor
- Solves the equation to find the key and deciphers d
 - $e \cdot d \equiv 1 \pmod{\phi(N)}$ and $0 \leq d \leq N$
- Publishes the public key: $KU=\{e, N\}$
- Maintains the private key to decipher: $KR=\{d, p, q\}$

Example RSA

1. Selects prime numbers: $p=17$ & $q=11$
2. Calculates $n = pq = 17 \times 11 = 187$
3. Calculates $\phi(n) = (p-1)(q-1) = 16 \times 10 = 160$
4. Selects e : $\gcd(e, 160) = 1$; chooses $e=7$
5. Determines d : $de=1 \bmod 160$ and $d < 160$: value is $d=23$ because $23 \times 7 = 161 = 160 + 1$
6. Publishes public key $KU = \{e, N\} = \{7, 187\}$
7. Maintains secret key $KR = \{d, p, q\} = \{23, 17, 11\}$

Use of RSA

- To cipher the message M , the sender:
 - Obtains a public key of the receiver $KU = \{ e, N \}$
 - Calculates: $C = M^e \text{ mod } N$, where $0 \leq M < N$
- To decipher C :
 - Uses its private key $KR = \{ d, p, q \}$
 - Calculates: $M = C^d \text{ mod } N$
- Message M has to be lower than the modulus N
- Message $M = 88$ ($88 < N = 187$)
- Cipher: $C = 88^7 \text{ mod } 187 = 11$; $e=7$
- Decipher: $M = 11^{23} \text{ mod } 187 = 88$; $d=23$

RSA Real

p

12131072439211271897323671531612440428472427633701410925634549312301964
37304208561932419736532241686654101705736136521417171171379797429933487
1062829803541

q

12027524255478748885956220793734512128733387803682075433653899983955179
85098879789986914690080913161115334681705083209602216014636634639181247
0987105415233

n

14590676800758332323018693934907063529240187237535716439958187101987343
87990053589383695714026701498021218180862924674228281570229220767469065
43401224889672472407926969987100581290103199317858753663710862357656510
50788371429711563734278891146353510271203276516651841172685983798867211
1837205085526346618740053

$\phi(n)$

14590676800758332323018693934907063529240187237535716439958187101987343
87990053589383695714026701498021218180862924674228281570229220767469065
43401224889648313811232279966317301397777852365301547848273478871297222
05858745715289160645926971811926897116355507080264399952954964411681194
7516513938184296683521280

e - the public key

65537

d - the private key

89489425009274444368228545921773093919669586065884257445497854456487674
83962981839093494197326287961679797060891728367987549933157416111385408
88132754881105882471930775825272784379065040156806234235500672400424666
65654232383502922215493623289472138866445818789127946123407807725702626
644091036502372545139713

Encryption/Decryption

Encryption: $1976620216402300889624482718775150^e \bmod n$

35052111338673026690212423937053328511880760811579981620642802346685810
62310985023594304908097338624111378404079470419397821537849976541308364
64387847409523069325349451950801838615742252262188798272324539128205968
86440377536082465681750074417459151485407445862511023472235560823053497
791518928820272257787786

Decryption:

35052111338673026690212423937053328511880760811579981620642802346685810
62310985023594304908097338624111378404079470419397821537849976541308364
64387847409523069325349451950801838615742252262188798272324539128205968
86440377536082465681750074417459151485407445862511023472235560823053497
791518928820272257787786 $^d \bmod n$

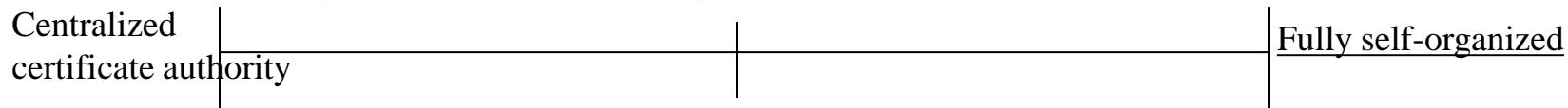
1976620216402300889624482718775150 (which is our plaintext)

Key Management in ad-hoc networks

- Challenges
 - Vulnerable mobile nodes
 - More exposed to physical attacks
 - Mobility-induced unstable network topology
 - Rapid change in connectivity
 - Potential network partition
 - No infrastructure to send key information to nodes

Self-organized public key management (SOPKM)

- The scope of key management



- Public key management system
 - How to get the authentic public key of a user?
- Users issue certificates based on personal acquaintance
 - Certificate: binding between node and its public key: public key, identity and signature of the issuer
 - Certificates stored and distributed by the users themselves
- Updated certificate repository
- Non-updated certificate repository
 - Expired certificates

SOPKM: Public Keys and Public-Key Certificates

- If a user u believes that a given public key K_v belongs to a given user v , then u can issue a public-key certificate in which K_v is bound to v by the signature of u .
 - Certificates are issued with a limited validity period T_v
 - Each certificate contains its issuing and expiration times.
-
- u gets the public key K_v of v through a side channel
 - u creates certificate for v to bind the K_v and v
 - u stores the certificate of v in its local repository G_u and sends the certificate to v
 - Each time a user u issues a certificate that binds another user v to her public-key K_v , u sends the certificate to v
 - *In this way, each certificate is stored at least twice: by its issuer and by the user to whom it is issued.*

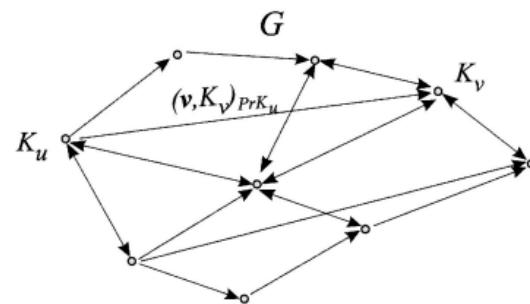
SOPKM: Update repositories of certificate graphs

Certificate of

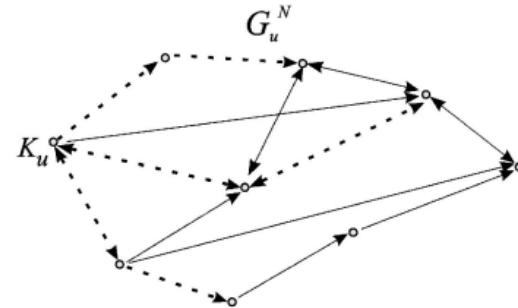
u to v , using u
private key
 $\text{pr } K_u(v, K_v)$

G_u Updated
global
certificate of u
 G_u^N non-updated
global
certificate of u

Step 0. Each user creates her own public/private key pair $K/\text{Pr}K$

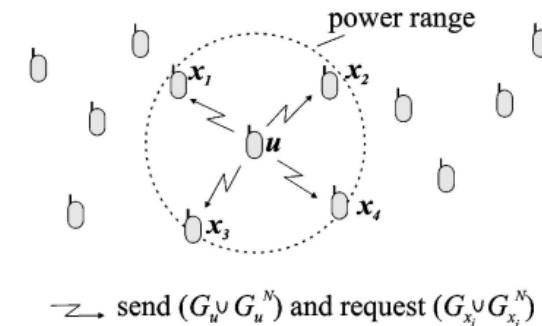


Step 1. Issuing of public-key certificates
(creation of the certificate graph G)

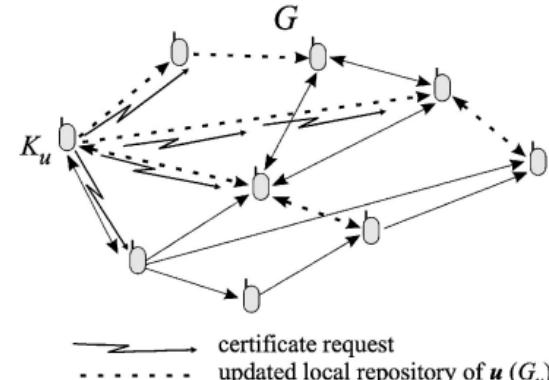


----- updated local repository of u (G_u^N)

Step 3a. Node u constructs its updated repository from G_u^N



Step 2. Certificate exchange



Step 3b. Node u constructs its updated repository by communicating with other nodes

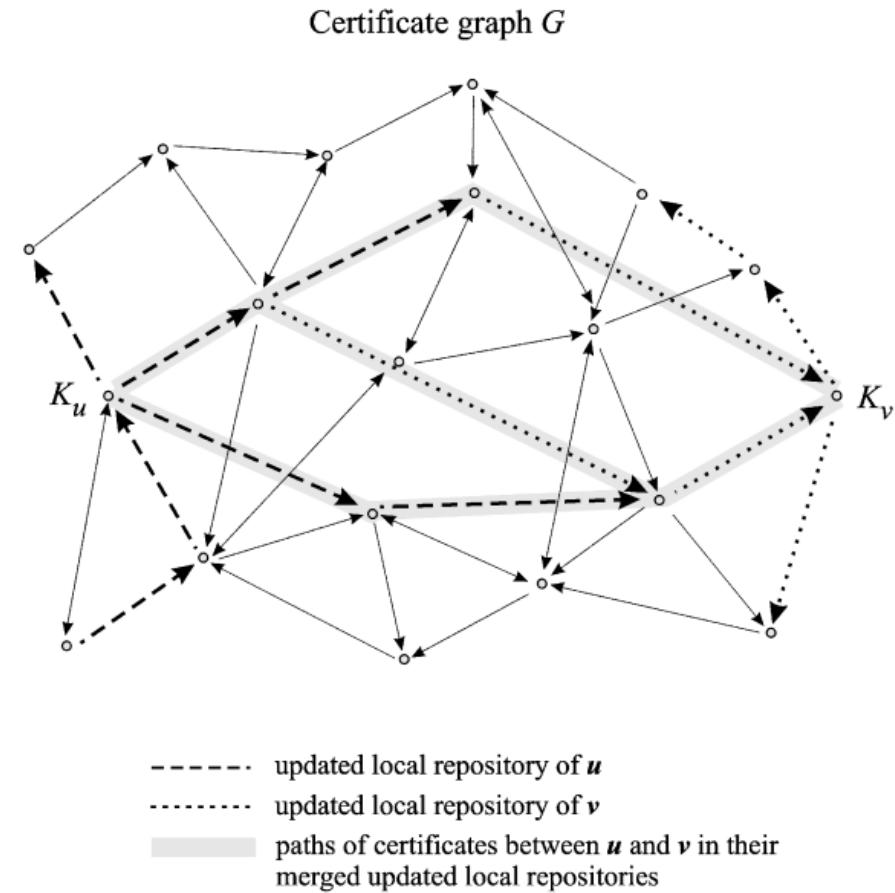
SOPKM: Certificate Exchange

- Each node u multicasts its subgraphs G_u and G_u^N to its physical neighbors (only 1 hop away)
- In this message, u does not send actual certificates, but only appropriate unique identifiers (e.g., their hash values)
- The neighbors of node u that receive the message from u reply with the hash values of the certificates in their updated and non-updated repositories.
- Node u then crosschecks the received values with the certificates that it holds and requests from its neighbors only the certificates that it does not hold.

SOPKM: Global Connectivity Graph

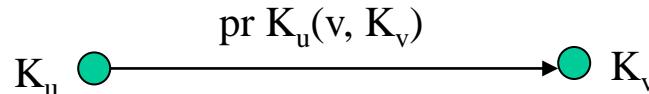
- Global certificate graph
- Connectivity of certificate graph
 - Small world phenomenon
 - Due to social relationships
 - Mobility increases the connectivity
 - Mobile nodes/users exchange their public keys whenever they meet
- Vertices: public keys of some nodes
- Edges: public keys certificates issued by users

$$K_u \xrightarrow{\text{pr } K_u(v, K_v)} K_v$$

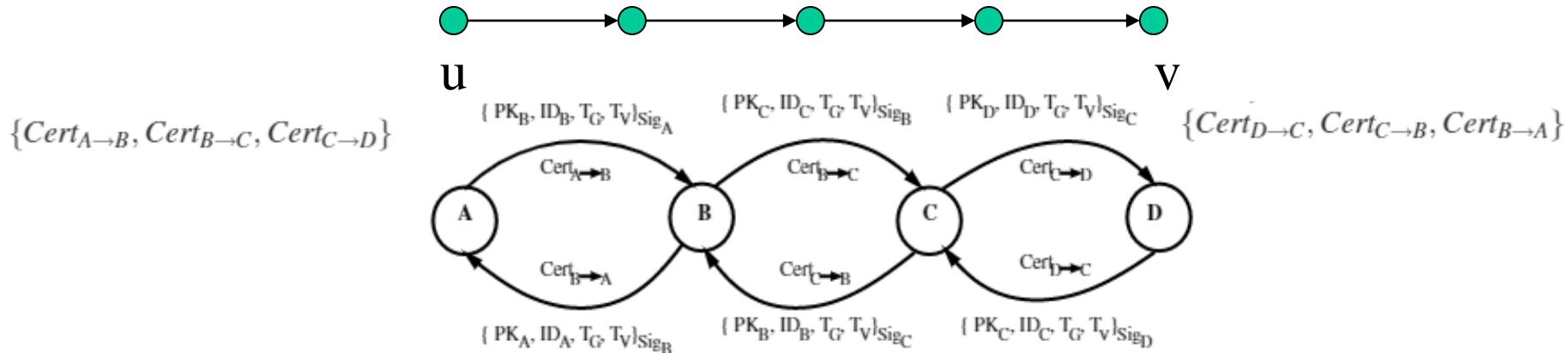


SOPKM: Certificate graph

- Vertices: public keys of the nodes
- Edges: public keys certificates issued by users



- User u wants to obtain public key of user v
 - Find a chain of valid public keys certificates leading to v
 - First hop uses edge from u (certificate issued by u)
 - Last hop is a certificate issued by v
 - Intermediate nodes trusted through the previous certificate in the path



SOPKM: Certificate Revocation

- Certificate revocation
 - **Explicitly**
 - Each node has a list of nodes that request updates for the certificates they issue
 - **Implicitly**
 - Expiration time of the certificate
 - Simple but need time synchronization of the nodes and decide the expiration time properly
- Key revocation
 - Inform the issuers of its certificate, then the certificate is invalid

SOPKM: Malicious Users

- The certificate exchange mechanism allows nodes to gather virtually all certificates from G
 - Nodes cross-check user-key bindings in certificates that they hold and detect any inconsistencies (i.e., conflicting certificates).
 - If they contain inconsistent user-key bindings (i.e., if both certificates contain the same username but different public keys)
 - If they contain the same public-key, but are bound to different usernames.
- Several certificate exchanges to solve conflicts, if they are needed

Self-securing ad-hoc wireless networks (SSAWN)

- Usually, an entity is trusted only if it is verified by a central authority, which cannot be the case in wireless and ad-hoc networks
- Goal of a self-securing network
 - Achieve high security assurance
 - High success ratio
 - Efficient communication
- Localized trust model, an entity is trusted if any k trusted entities claim so within a certain time period
 - *k* entities typically among the entity's one-hop neighbors
 - Cares most the trustworthiness of its immediate neighbors in practice - a node will communicate with the rest of the world via its one-hop neighbors.
 - Once a node is trusted by its local community, it is globally accepted as a trusted node.
 - Otherwise, a locally distrusted entity is regarded as untrustworthy in the entire network.

SSAWN: Shared secrets

- Encryption mechanism uses RSA asymmetric keys
- Global Secret Key (SK) and the corresponding Public Key (PK)
 - SK functionality is 'distributed' among nodes
 - Any K nodes holding a partial secret form a distributed Certificate Authority (CA)
- SK is used to sign certificates for all nodes in the network.
- A certificate signed by SK can be verified by the well-known public key P K.
- Threshold secret sharing
 - Each node has a part of the secret
 - Unique ID, derived from the node's address
 - Mechanism for local detection of misbehaving nodes
 - At least K one-hop neighbouring nodes
 - Key pair for each node (public and secret keys)

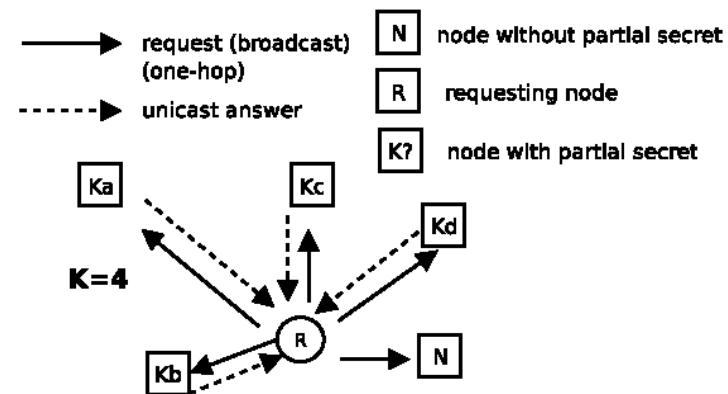
SSAWN: Basic Operation

- Basic operation
 - **Distributed PKI**
 - System private key is split to server nodes
 - Quorum of k servers produce certificate updating
 - Structure of certificate
 - **Operates in phases**
 - Server group formation/maintenance
 - Certificate updating/revocation
 - Shared key updating/renewing
- SK is not visible, known or recoverable by any network node
- Each node carries a certificate signed with SK.
- PK is assumed to be well-known for certificate verification.
- Nodes without valid certificates are denied from access to any network resources such as routing and packet forwarding

ID_i	T_{valid}	K_i	$flag$	ver	$sign.$	$issuer$	$Algo.$
--------	-------------	-------	--------	-------	---------	----------	---------

SSAWN: Shared secrets

- Partial secret key as a function of nodes IDs
 - Generation of a polynomial of order $K-1$, known only in the initial setup
 - K nodes holding a partial secret share recover SK using Lagrange interpolation
 - Coalition of $K-1$ nodes holding a partial secret share does not have any information about SK
- Node wanting to use the distributed CA
 - Contact K nodes that have a partial secret share
 - K one-hop neighbouring nodes
 - It is easier to collect reliable information about misbehaviour of closer nodes
 - PK is known by all nodes



SSAWN: Shared secrets

- Upon the receipt of v_i 's certification request, a node checks its records.
 - If its record shows v_i as a well-behaving legitimate node, it returns a “partial” certificate by applying its share of SK.
 - Otherwise, the request is dropped.
- By collecting k partial certificates, v_i combines them together to generate the full new certificate as if it were from a CA server
 - Upon receiving k partial certificates from coalition, node v_i multiplies them together to recover its full certificate (lagrange interpolation – polynomial that passes through the several points,
<https://mathworld.wolfram.com/LagrangeInterpolatingPolynomial.html>)
- A misbehaving or broken node will be unable to renew its certificate.
- A valid certificate represents the trust from a coalition of k nodes
 - Nodes with valid certificates are globally trusted.
 - Each node contributes to the overall trust management and maintenance by monitoring and certifying its neighboring nodes

Reputation approaches

Well Behaving Nodes?

- Explore the reliability of each node
 - Routing and communication through high reputation nodes
 - Protect network traffic from misbehaving nodes
 - Minimizing interaction with misbehaving nodes
-
- Evaluate the reputation of this node
 - This needs to be done in a distributed approach by each node
 - Each node monitors the behavior of each neighboring node
 - Nodes send their information to other nodes

Behavior and Reputation

- Explore the way that a node handles packet forwarding: forward, change, inject packets;
- Evaluate through observation of a node's behavior
- Reputation information periodically exchanged among neighbors
- Level of trust: the level of trustworthiness in the reports coming from a particular node
- Network interfaces of the nodes need to support promiscuous mode
 - If node A is within range of node B, it can overhear communications to and from node B

Good or bad nodes?

- Self-organized system
 - Decentralized: cooperation can be enforced, but not guaranteed
 - Recognize and sanction intolerable behavior, reward obedient members
- Each node observes behavior and evaluates the reputation of its neighbors
- Reputation can be used to select the most reliable and secure path
- Friendly, selfish (do not transmit) and malicious nodes (misroute, inject packets)
 - If the behavior changes, node's reputation changes accordingly

Reputation

- Total reputation is a combination of
 - First-hand reputation (neighbors)
 - Second-hand reputation (information from neighbors)
- A node's behavior follows a distribution
 - Number of observed packets from a node
 - Number of packets not transmitted or changed
 - Probability of well-transmitted packets (first-hand reputation) $> x$
 - Friendly behavior (Bayesian approach)
- Collaborative monitoring
 - Exchange first-hand reputation with neighbors
 - Deviation test to detect false reports
 - Probability of well-transmitted x seen from different nodes $<$ deviation value d
 - Trustworthy reports (Bayesian approach)

Reputation

- Trustworthy report from A concerning B
 - $T=trustworthy$
 - $N=non-trustworthy$
- Trust value $\alpha = T/(T+N)$
- Merge first-hand and second-hand reputation information
- Second-hand reputation is discounted by the trust factor
- Reputation of node B seen by node A is the first-hand reputation plus the trust factor of a second-hand reputation of B seen by C (and of other second-handed reputations)
 - $R_{AB} = F_{AB} + \alpha F_{CB}$

Reputation into the normal operation of the network

- Choice of nodes to form a path from the source to a destination
 - Reputation of the nodes in the choice algorithm
- Choice of nodes to join a certification graph
 - Nodes with high reputation
- Choice of nodes to get the key parts or key pairs
 - Nodes with high reputation
- Reputation of a node changes over time
- Nodes can be out of the network communication for a specific time

Edge Computing, 5G and 6G

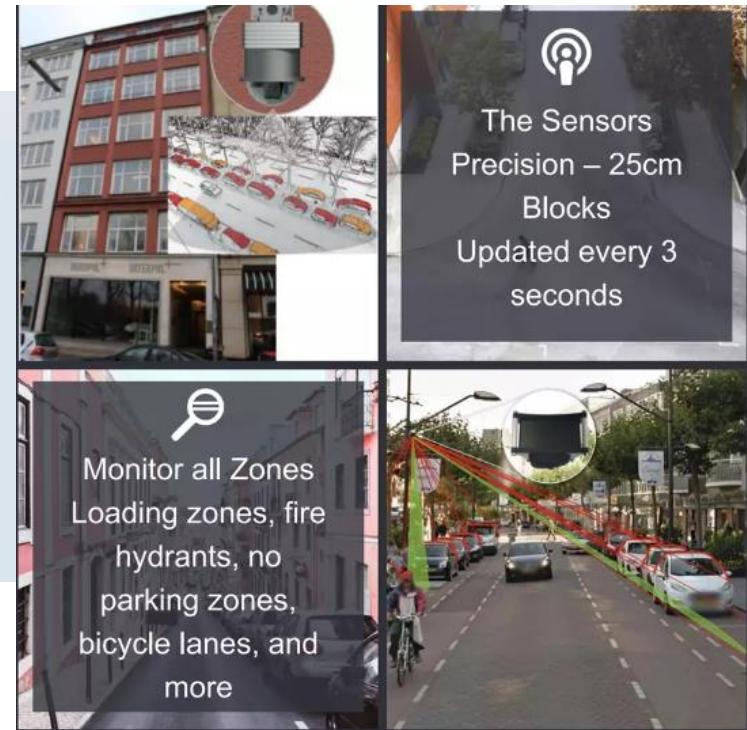
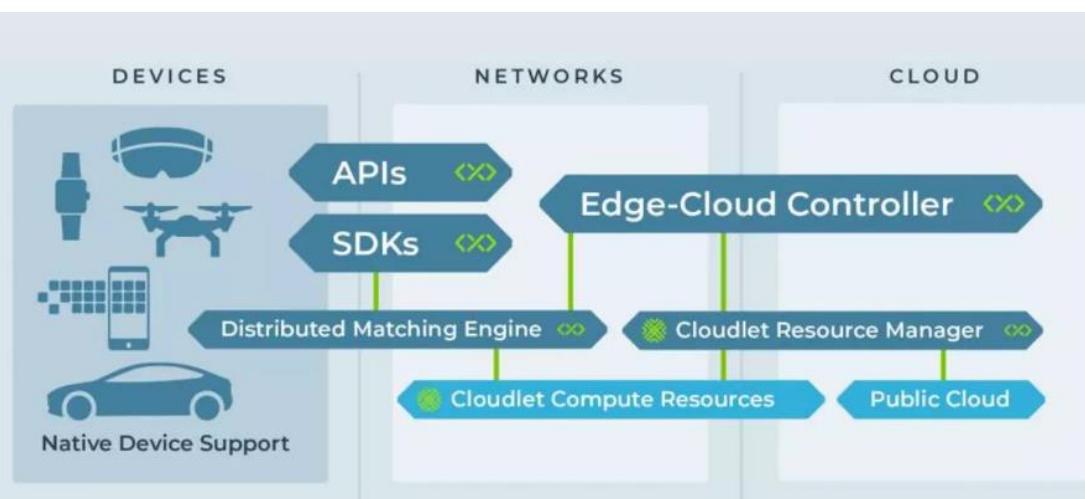
**Mestrado em Engenharia de
Computadores e Telemática**

2022/2023

Mobile Edge Computing (MEC)

- Concept in 5G
- Brings the cloud closer to the network edge
- Opens the edge for applications from 3rd parties
- Provides services to enhance applications with context information (network information, location)
- Traffic redirection
- Ultra-low latency
- Facilitates running applications at the right location and at the right time
- Different IoT solutions have different network requirements
 - Autonomous vehicles and medical equipment with strong latency requirements with critical decisions
 - Edge computing to enable strong processing and decisions near the needs of the reactions

Simplified MEC vision

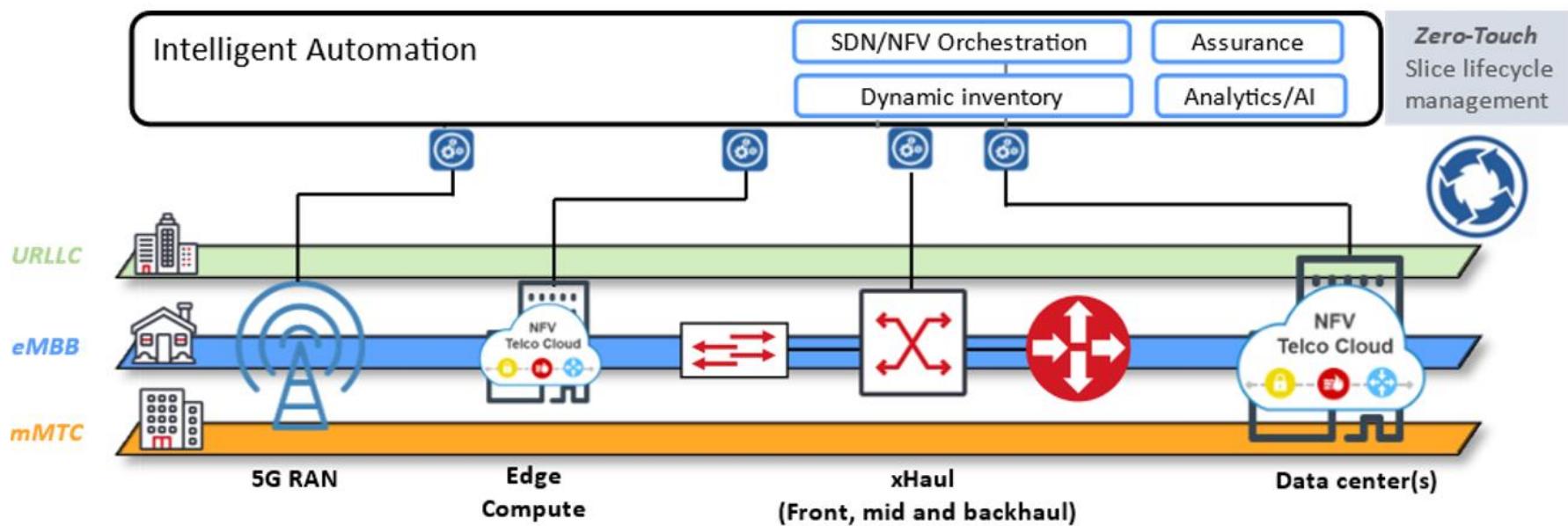


- iPhone security
 - Host sensitive biometrics or encryption information on user device and not in the cloud
 - Does not need to handle user security information
- AWS DeepLens
 - Camera with Linux OS, machine learning in the device
 - Near-time set of intelligence tools integrated into APIs

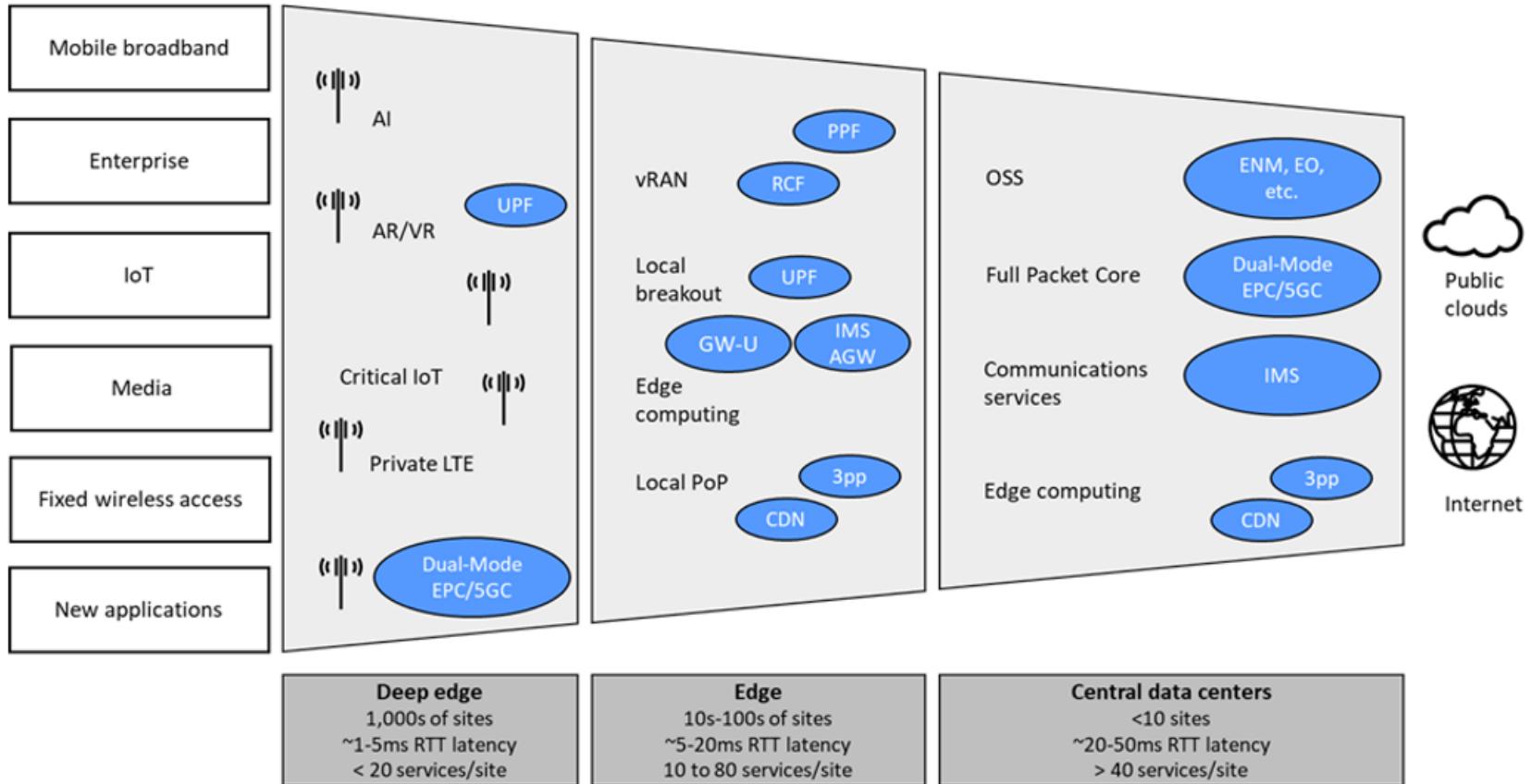
Edge and Cloud

- Edge computing devices depend on network access to the cloud to receive machine learning and complex event processing models
- They also need to send sensors and status data back to the cloud
- Strong bandwidth requirements → 5G to 6G
- Low latencies
- Massive amounts of nodes

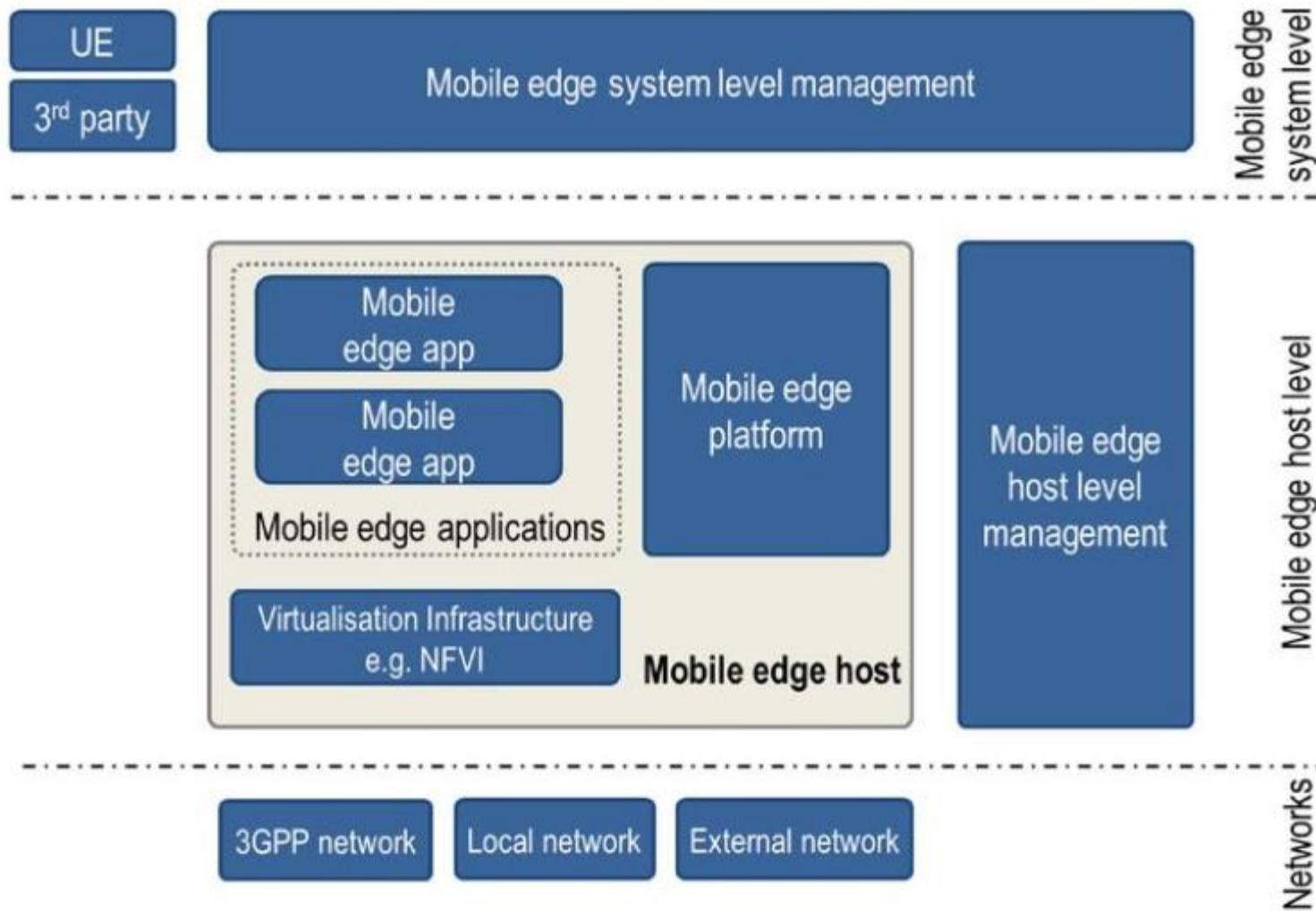
Distributed intelligence and 5G



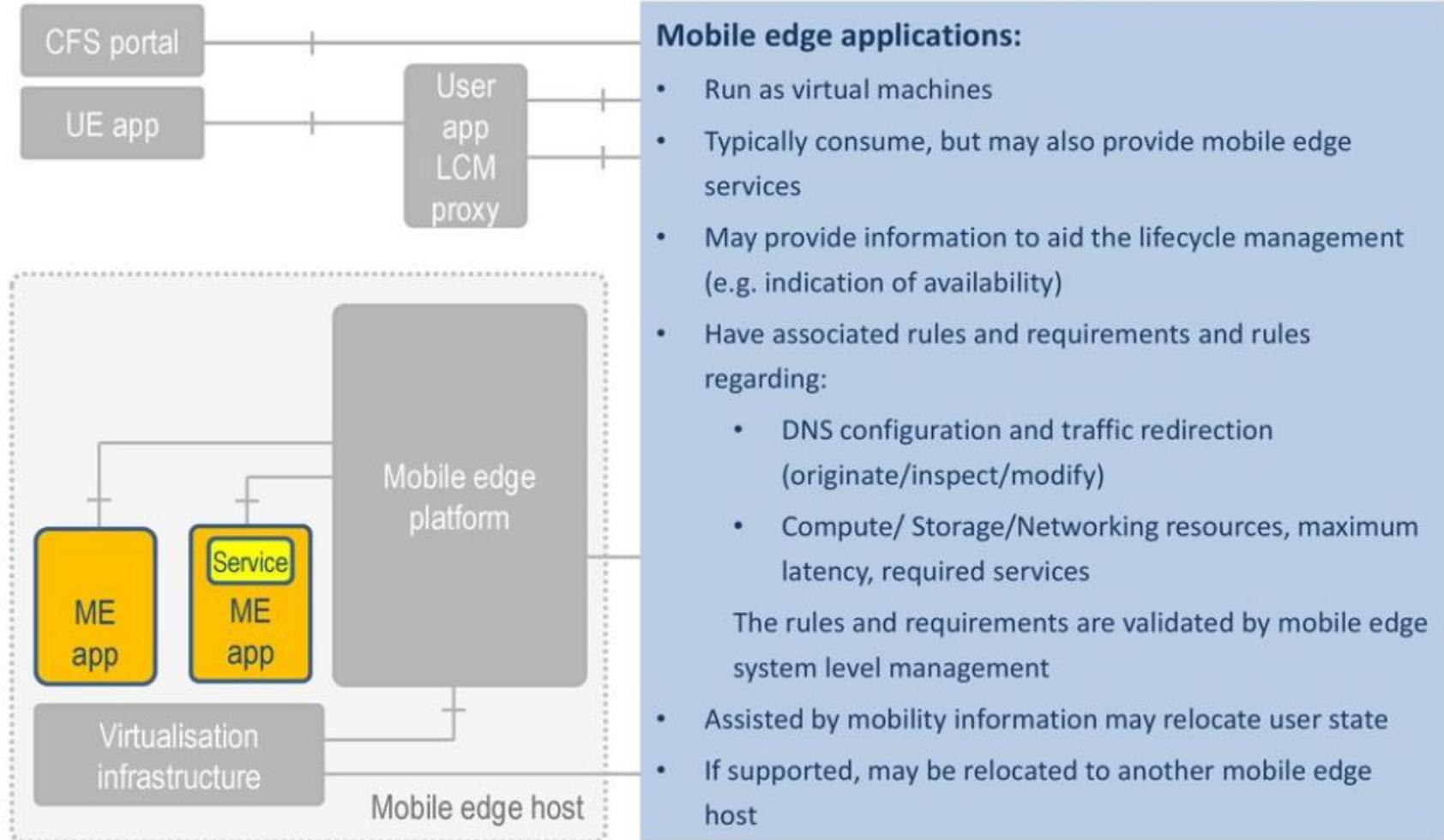
Distributed Edge and 5G



⁷ Edge Computing: ETSI MEC Reference Architecture

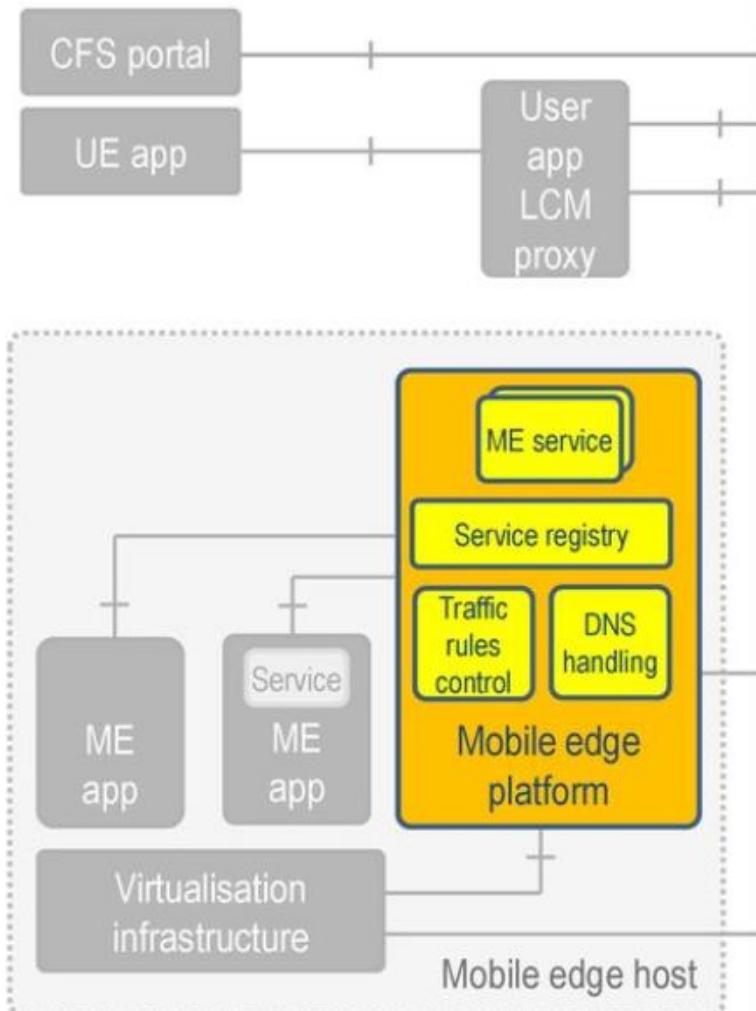


MEC: Mobile Edge Apps



Examples: handover, caching, ML processing, video processing, etc.

MEC: Mobile Edge Platform



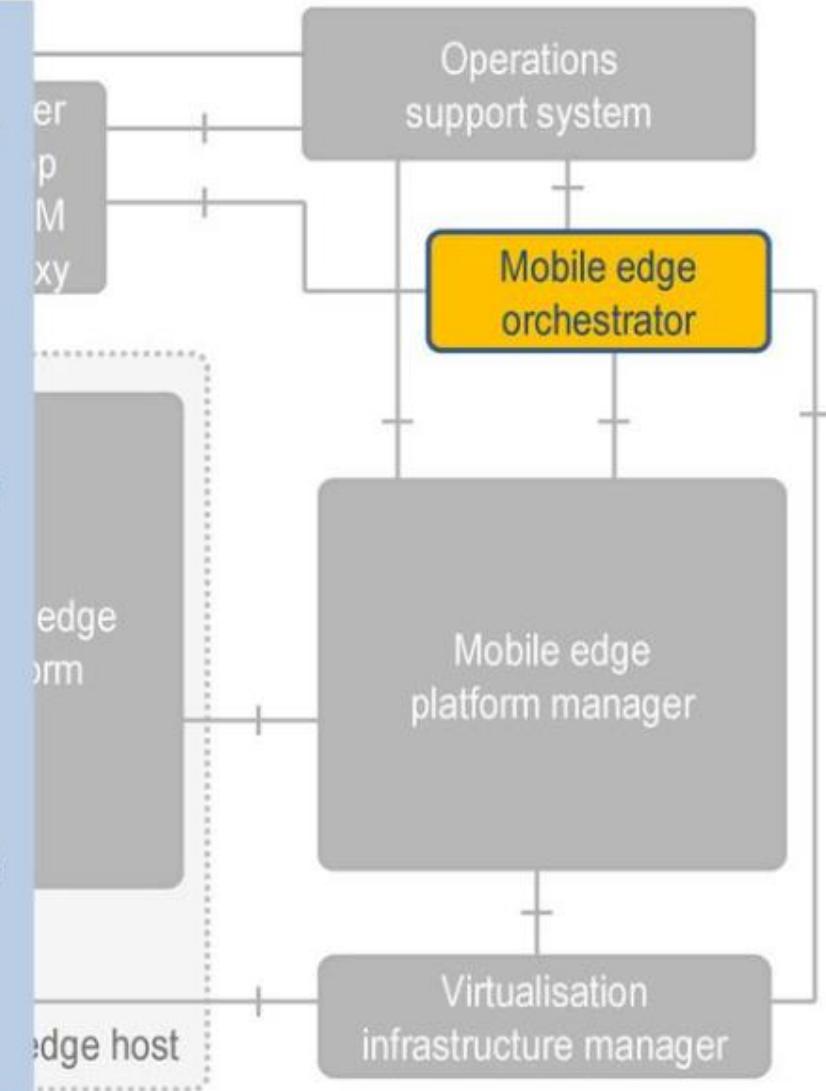
Mobile edge platform:

- An environment where applications can discover, advertise, consume and offer mobile edge services
- Controls data-plane in the virtualization infrastructure (“SDN”) based on traffic rules
- Configures DNS proxy/server based on DNS records from the mobile edge platform manager
- Provides mobile edge services. May also consume mobile edge services provided by the applications
- Provides access to persistent storage and time of day information

MEC: Mobile Edge Orchestrator

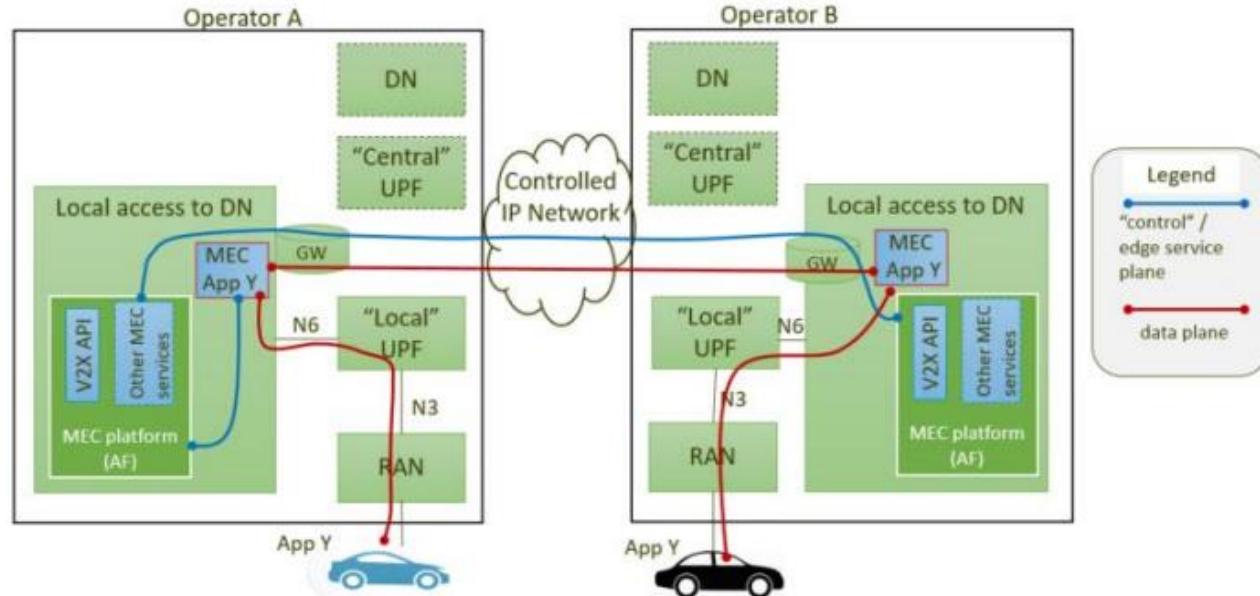
Orchestrator:

- Maintains an overall view of the system and mobile edge hosts, available resources, available services and topology
- On-boards application packages, including
 - an integrity check and authenticity
 - validation of the application rules and requirements; if necessary adjusts them to comply with operator policies
 - Maintenance of a record of on-boarded packages
 - Preparation of the virtualization infrastructure manager(s) to handle the apps
 - Selection of the appropriate host for the application, satisfying its rules and requirements
- Triggers application instantiation and termination
- Optionally, triggers application relocation



Examples using Edge Computing

- A V2X application instance may be running on a car connected to MNO 1 which is equipped with a MEC system from vendor 1, and communicating with another V2X application instance, running on a second car connected to MNO 2, which, in its turn, is equipped with a MEC system from vendor 2.
- V2X service is implemented with two instances of the "MEC App Y", each of which communicates with its corresponding Client App, i.e. "App Y", and is also connected with a MEC platform in each respective MEC system (domain).



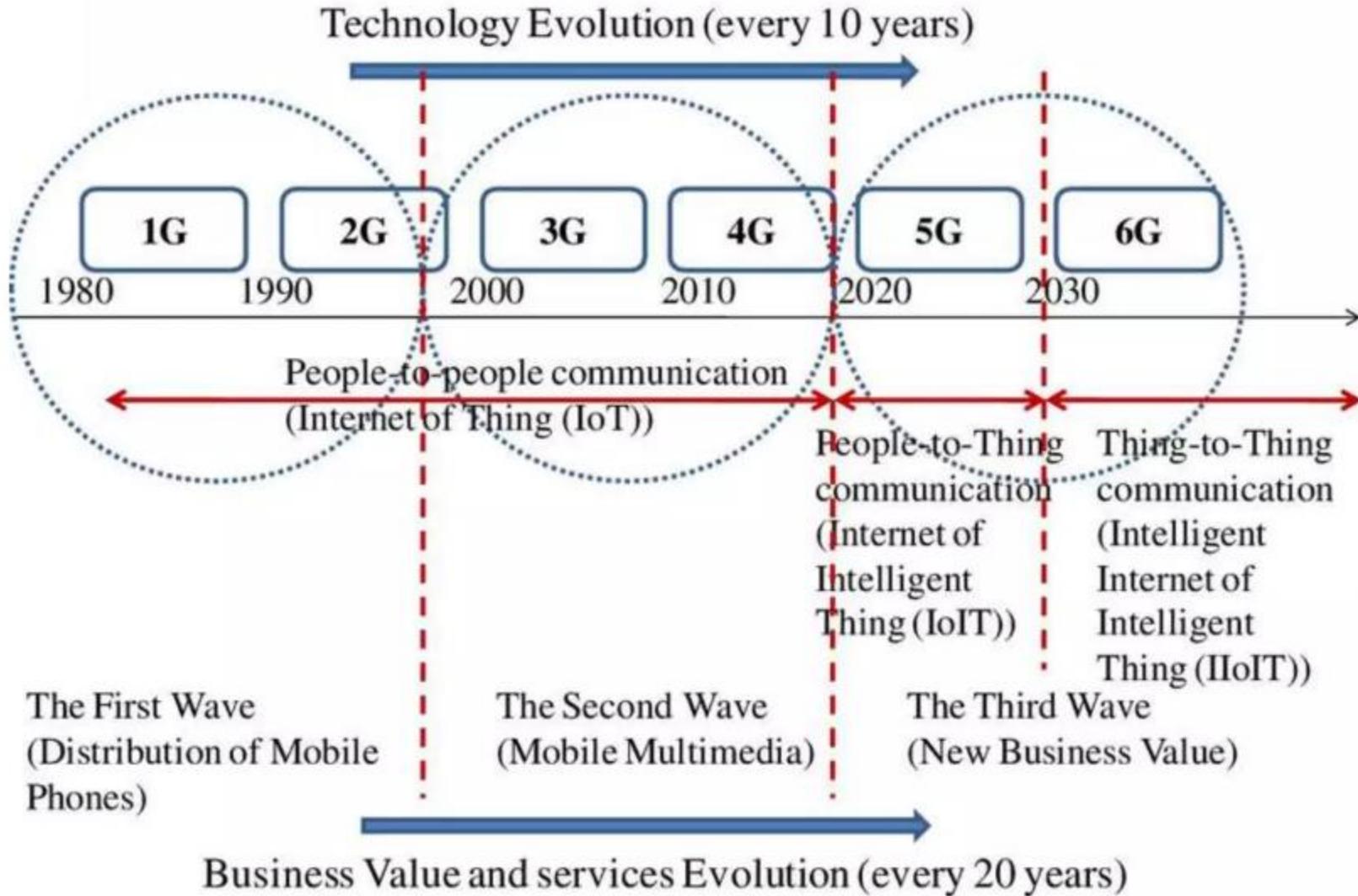
Examples using Edge Computing

- MEC is envisioned as a promising means to deliver better Quality of Experience (QoE) for immersive AR applications
 - Reduces the delay
 - Addresses computation-intensive and battery-consuming tasks offloaded from the mobile devices.
- 2 MEC application Xs, instantiated on MEC hosts of MEC system A and MEC system B respectively, communicate and coordinate together for synchronizing the game scenario. Information to be exchanged between the two MEC applications for coordination mostly include users' game play actions such as players' position, movement, direction, game control and the status of game contents virtually created.
-
- The diagram illustrates a dual-MEC system architecture for an AR game, divided into Area I (MNO-1) and Area II (MNO-2).
- Area I (MNO-1):** Contains MEC system A (green box). It includes MEC host A, MEC platform A, Data Plane, and MEC App. X. A Client App. X (User 1) is connected via N6 and UPF to the MEC host A.
 - Area II (MNO-2):** Contains MEC system B (yellow box). It includes MEC host B, MEC platform B, Data Plane, and MEC App. X. A Client App. X (User 4) is connected via N6 and UPF to the MEC host B.
 - Communication:** MEC App. X in MEC system A is connected to MEC App. X in MEC system B via a blue double-headed arrow labeled Mp1. Both MEC hosts are connected via a vertical orange double-headed arrow labeled Mp2.
 - Network Elements:** The diagram shows RAN (of MNO-2) and UPFs (N3, N6) connecting the MEC hosts to the external network.
 - Legend:**
 - MEC App. X (AR game): Represented by a black square icon.
 - Client App. X: Represented by a black circle icon.

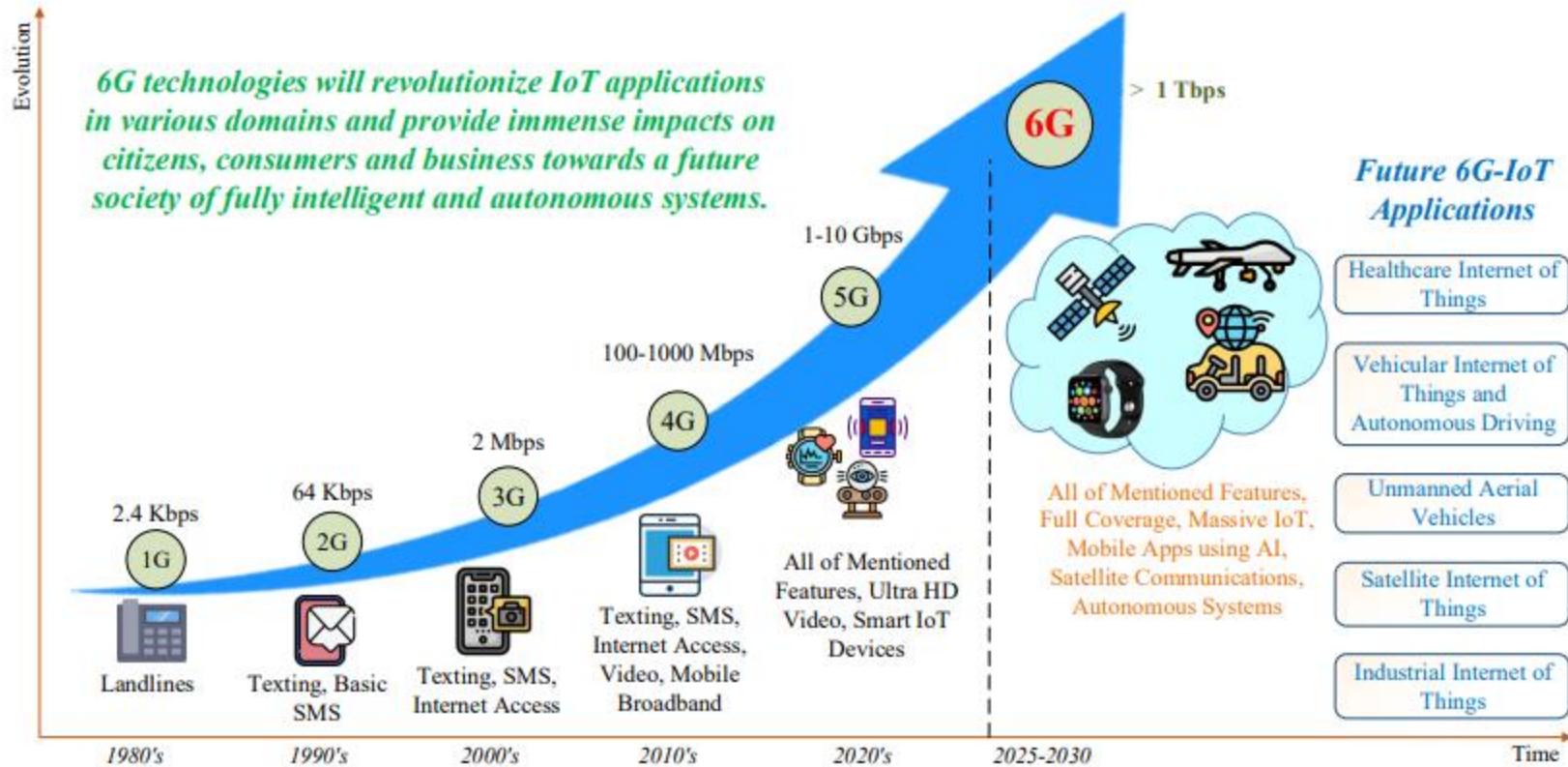
MEC future

- MEC is identified as a key enabler for IoT and mission critical vertical solutions
- Key architectural concept and technology for 5G
- Enables applications to be deployed and run in a virtualized environment
 - Services orchestrated and instantiated on demand where needed
- Myriad of different scenarios: health, industry, IoT, automotive, environment, etc
 - Change the application deployment
- Enabled autonomous networks and systems!

What about 6G?



What about 6G?



6G: self-organization, p2p relationships

6G networking and connectivity

Integrated Space Aerial, Terrestrial and Undersea Network

Ubiquitous 3d coverage

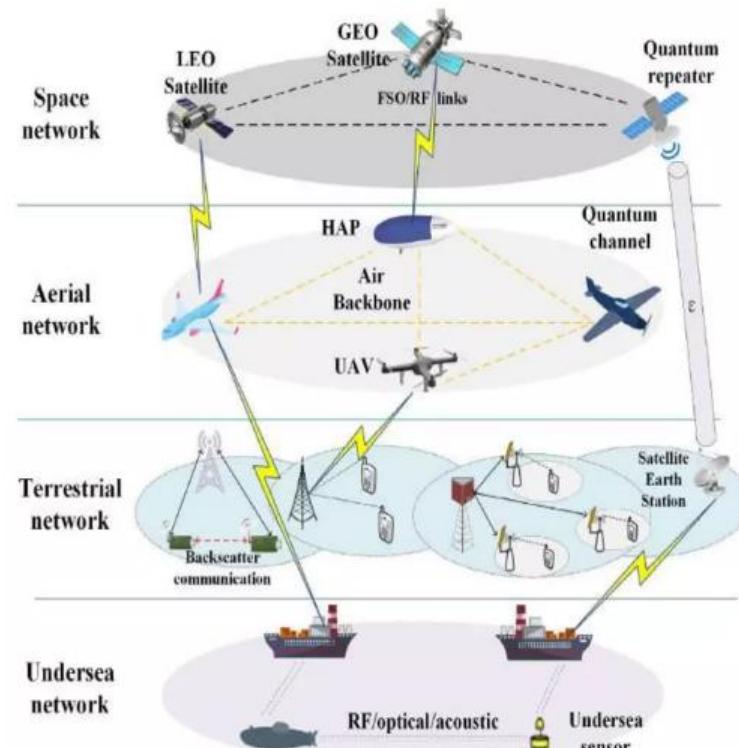
LEO- Low Earth Orbit

MEO- Medium Earth Orbit

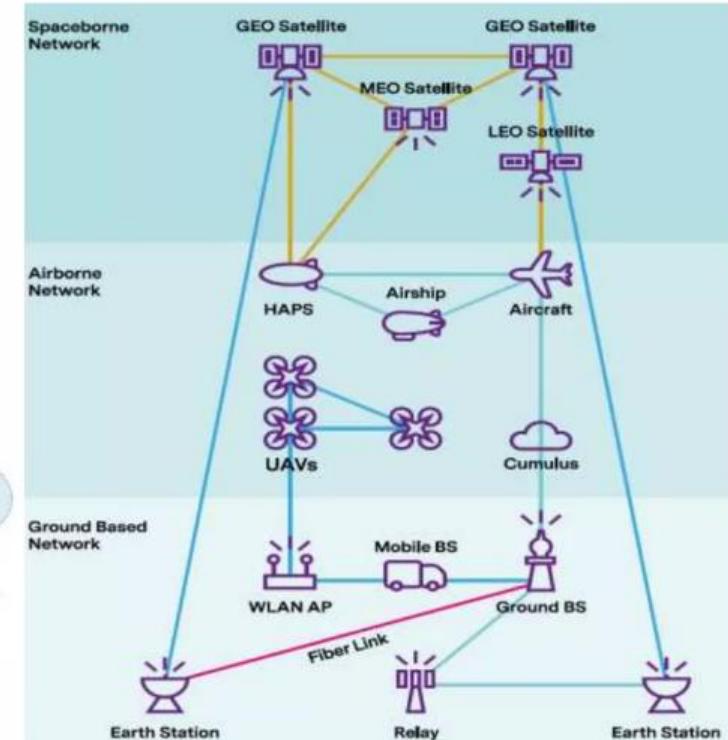
GEO- Geostationary Earth Orbit

HAP- High Altitude Platforms

UAV- Unmanned Aerial Vehicle

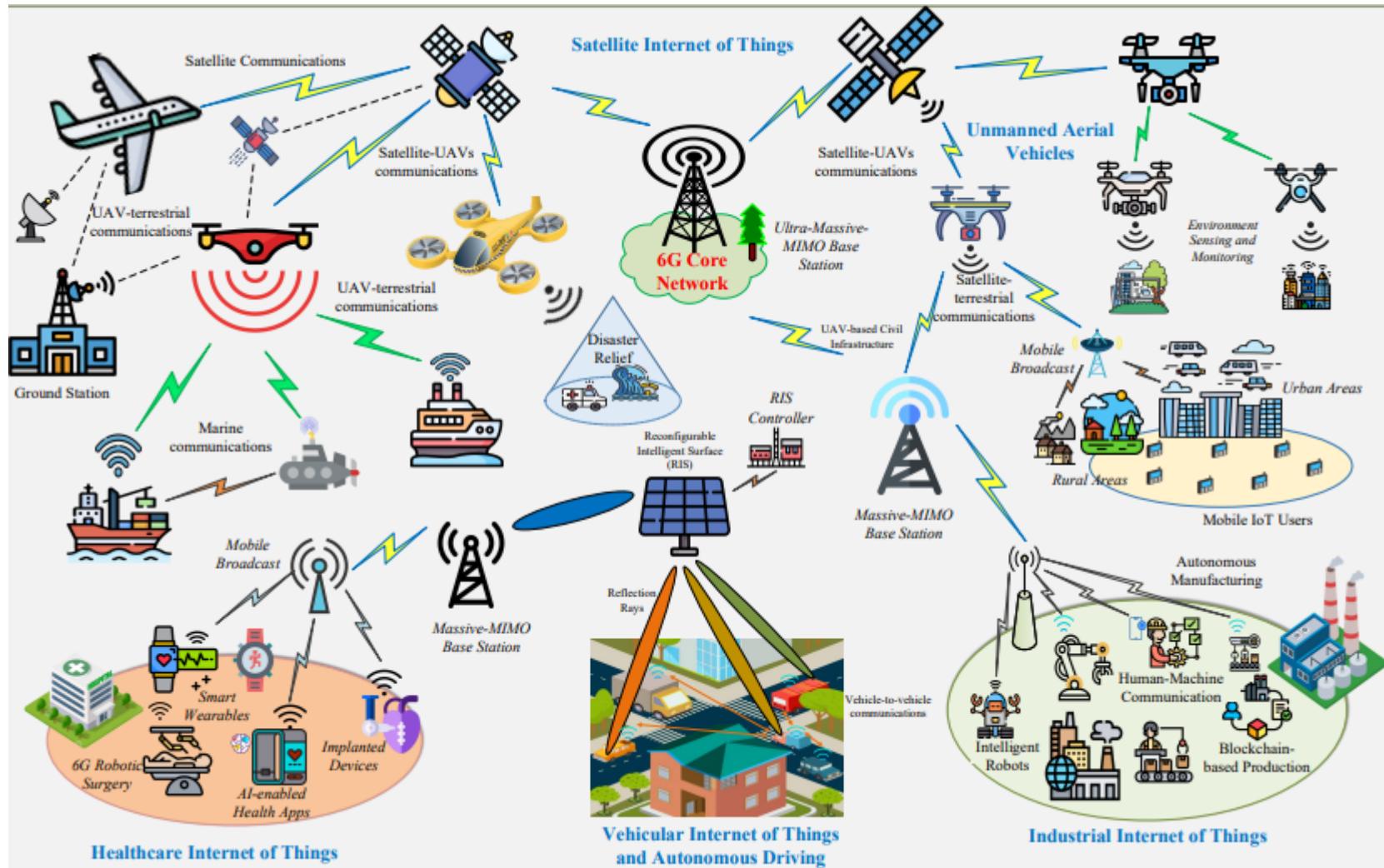


Source: T.Huang, et al., "A Survey on Green 6G Network: Architecture and Technologies", IEEE Access, VOLUME 7, 2019, <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8922617>



Source: 6G FlagShip, "WHITE PAPER ON BROADBAND CONNECTIVITY IN 6G", 6G Research Visions, No. 10 June 2020, Univ. of Oulu

6G in a 3D space



- https://www.etsi.org/deliver/etsi_gr/MEC/001_099/035/03.01.01_60/gr_mec035v030101p.pdf
- https://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/02.02.01_60/gs_MEC003v020201p.pdf
- https://5gaa.org/wp-content/uploads/2017/12/5GAA_T-170219-whitepaper-EdgeComputing_5GAA.pdf
- <https://link.springer.com/content/pdf/10.1007/978-3-030-83944-4.pdf>