Integrating IoT and Satcom: Lessons Learnt from the ESA M2MSAT Project

Dr. Ridha SOUA
Research Associate
University of Luxembourg

Outline

1- Satellites for IoT: Potential and Challenges

2- ESA M2MSAT: Context & Motivation

- IoT-Satellite Hybrid Networks
- IoT Application Protocol: MQTT and CoAP

3- MQTT Protocol Optimization

- Efficient IoT-Satellite Architecture
- MQTT-MFA: Aggregation of MQTT Topics

4- MQTT Protocol Optimization

- Efficient IoT-Satellite Architecture
- Aggregation of unicast CoAP responses to Group Communication Request

Outline

1- Satellites for IoT: Potential and Challenges

2- ESA M2MSAT: Context & Motivation

- IoT-Satellite Hybrid Networks
- IoT Application Protocol: MQTT and CoAP

3- MQTT Protocol Optimization

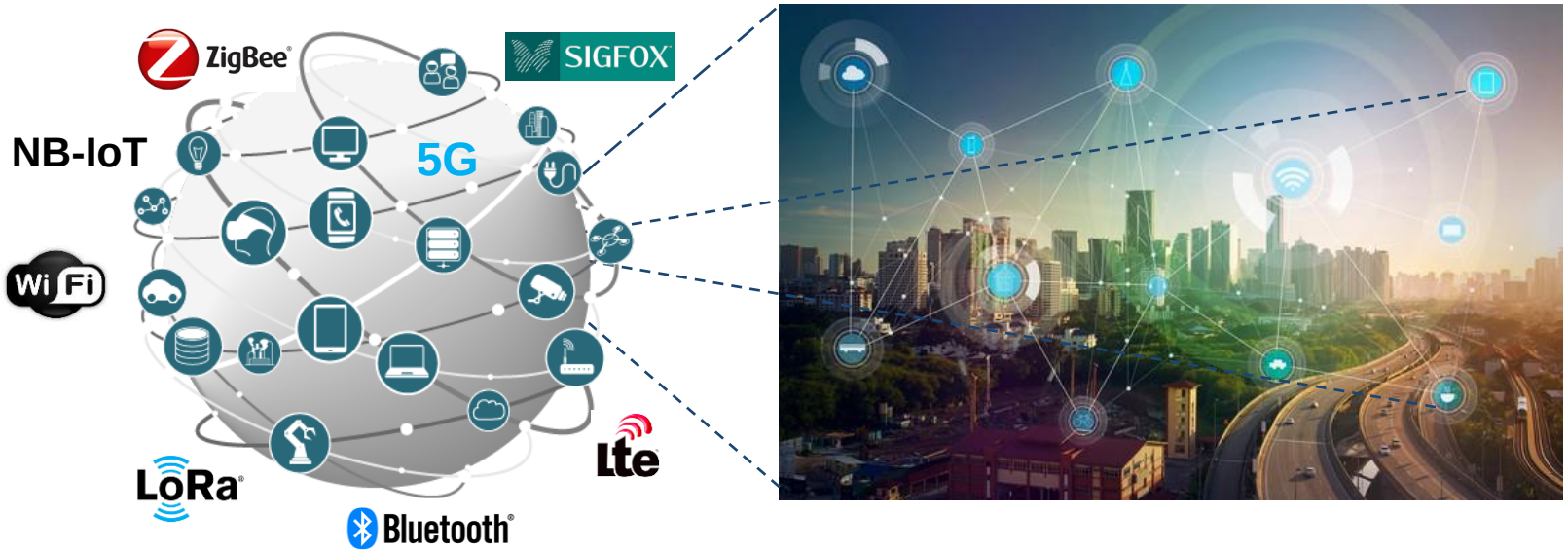
- Efficient IoT-Satellite Architecture
- MQTT-MFA: Aggregation of MQTT Topics

4- MQTT Protocol Optimization

- Efficient IoT-Satellite Architecture
- Aggregation of unicast CoAP responses to Group Communication Request

Satellites for IoT: Potential and Challenges

The internet of Things: a world of connected devices



ERICSSON

The number of IoT connected devices will surpass mobile devices in 2018

28 billion of connected devices worldwide, by 2021

CISCO

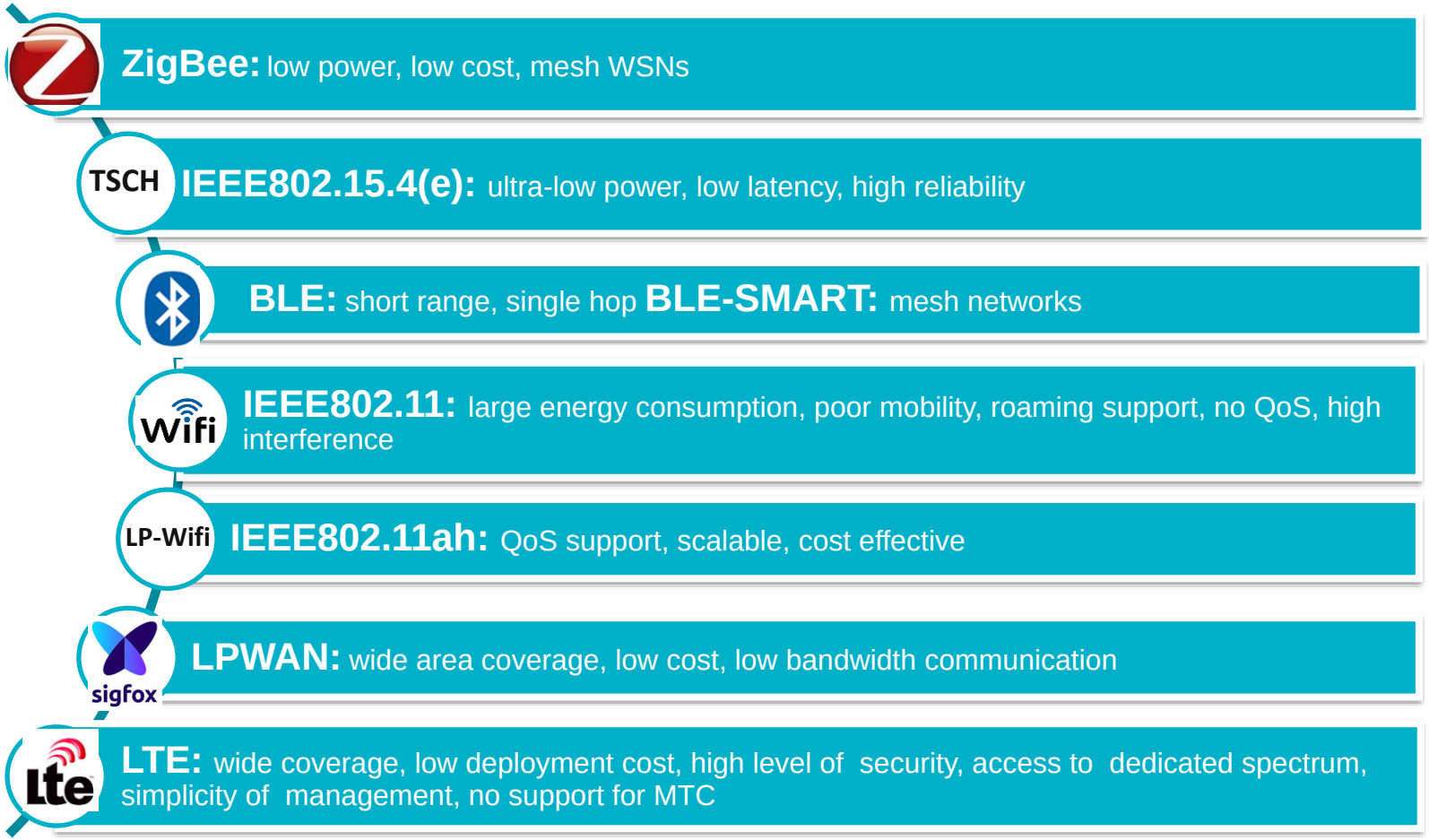
3.4 connected devices per person by 2020

IDC Analyze the Future

LoRaWAN & NB-IoT competing with Wi-Fi and BLE vendors

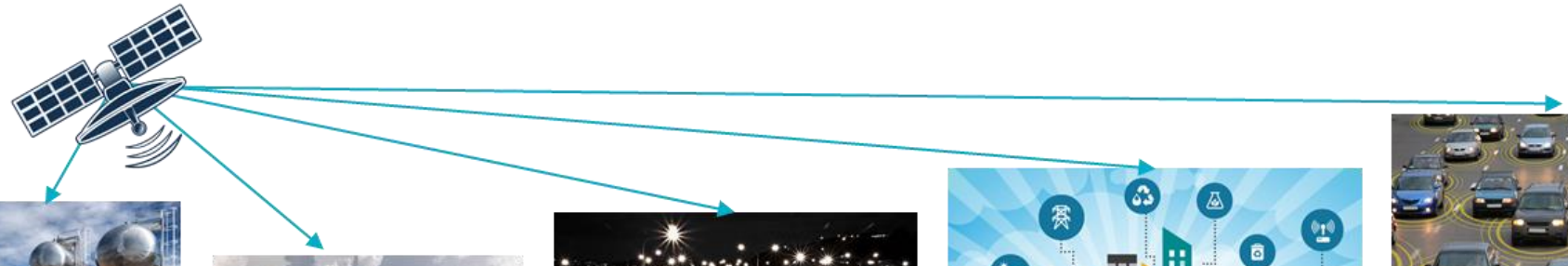
Satellites for IoT: Potential and Challenges

IoT Connectivity Landscape



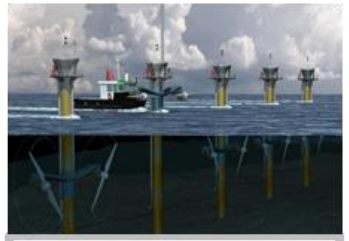
Satellites for IoT: Potential and Challenges

Transforming IoT connectivity via satellite: enabling new services



Connecting remote assets

Provide the critical communication means to conduct remote facility monitoring, real-time asset management at unmanned sites and offshore platforms.



Driving the use of sensor networks

Satellite-based sensor networks for energy and mining applications

Transforming transportation infrastructure

Enabling the next generation of mobility services (broadband connectivity on trains, cargo vehicles and maritime vessels)

Developing sustainable cities

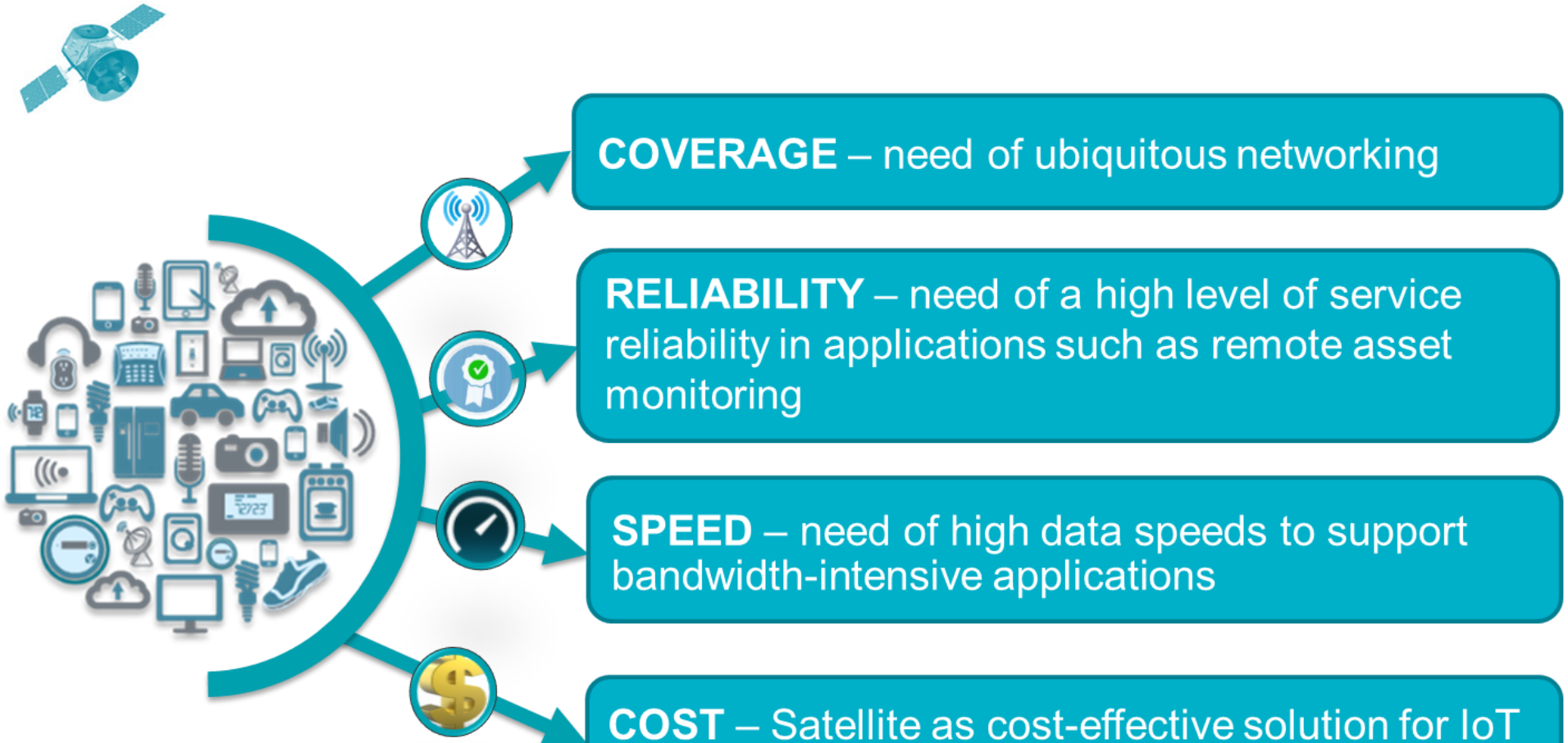
Satellite services can facilitate the deployment of smart grids in remote regions where terrestrial networks fall short and act as a reliable backup network for critical services such as safety and security

Connected Cars

Guarantee availability of safety services (emergency calling and airbag deployment notifications) in connected vehicles

Satellites for IoT: Potential and Challenges

Satellite PotentialS for IoT



COVERAGE – need of ubiquitous networking

RELIABILITY – need of a high level of service reliability in applications such as remote asset monitoring

SPEED – need of high data speeds to support bandwidth-intensive applications

COST – Satellite as cost-effective solution for IoT connectivity needs

Satellites for IoT: Potential and Challenges

Challenges to Face for INTEGRATING IoT and SATELLITE

Integration - need to meet different networks' requirements

Interoperability - ability to communicate despite different data format and protocols

Optimization – need to design modification to improve QoS and system performance

Outline

1- Satellites for IoT: potential and challenges

2- ESA M2MSAT: Context & Motivation

- **IoT-Satellite Hybrid Networks**
- **IoT Application Protocol: MQTT and CoAP**

3- MQTT Protocol Optimization

- Efficient IoT-Satellite Architecture
- MQTT-MFA: Aggregation of MQTT Topics

4- CoAP Protocol Optimization

- Efficient IoT-Satellite Architecture
- Aggregation of unicast CoAP responses to Group Communication Request

M2MSAT Project

https://artes.esa.int/projects/m2msat

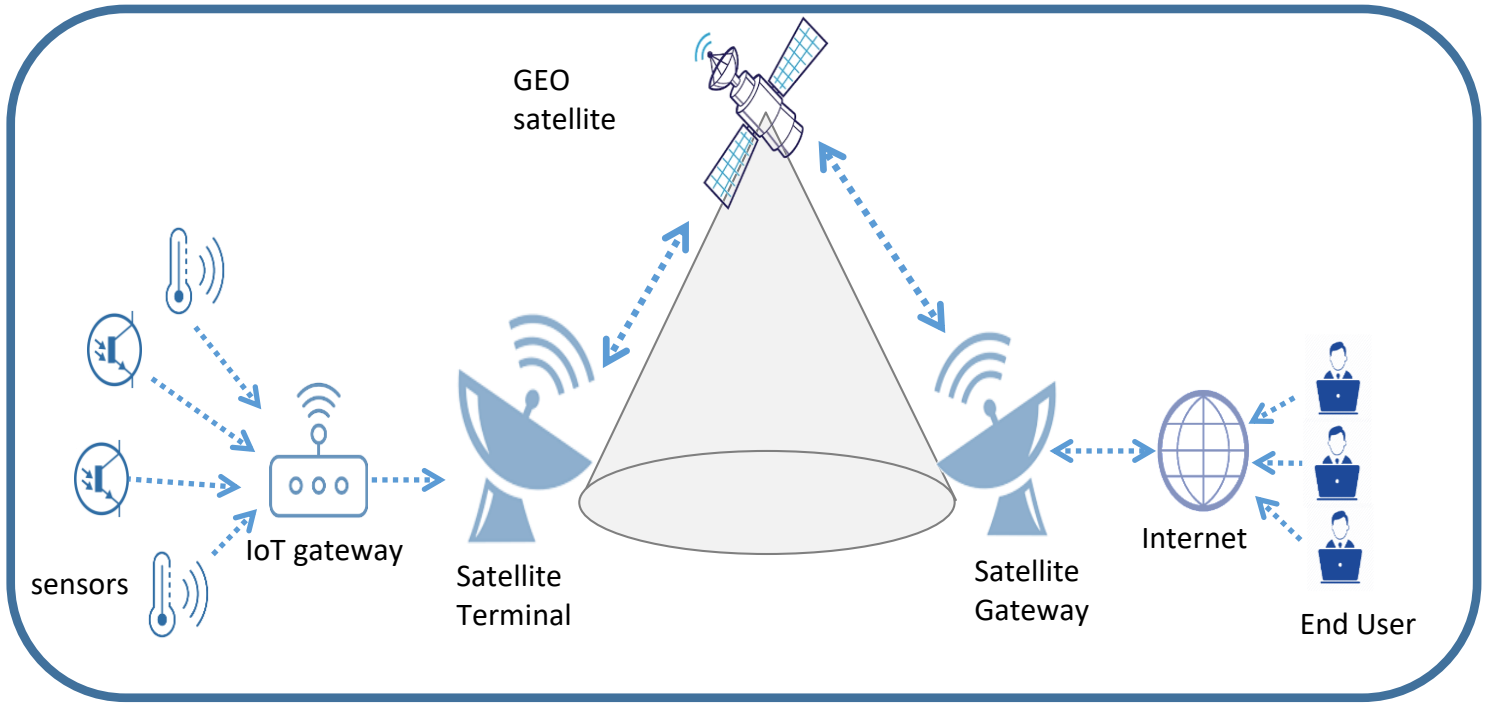
❑ Funded by ESA (ITT AO8550)

- Light-Weight Application and Transport Protocols For Future M2M Applications”
- Partners

- **Objectives**
  - ➢ Define references M2M/IoT satellite network scenarios
  - ➢ Identify M2M/IoT protocols suitable for future services
  - ➢ Design protocols optimizations, and integrated network functions
  - ➢ Demonstrate, via implementations in a real testbed, the achievable enhanced performance

M2MSAT Project

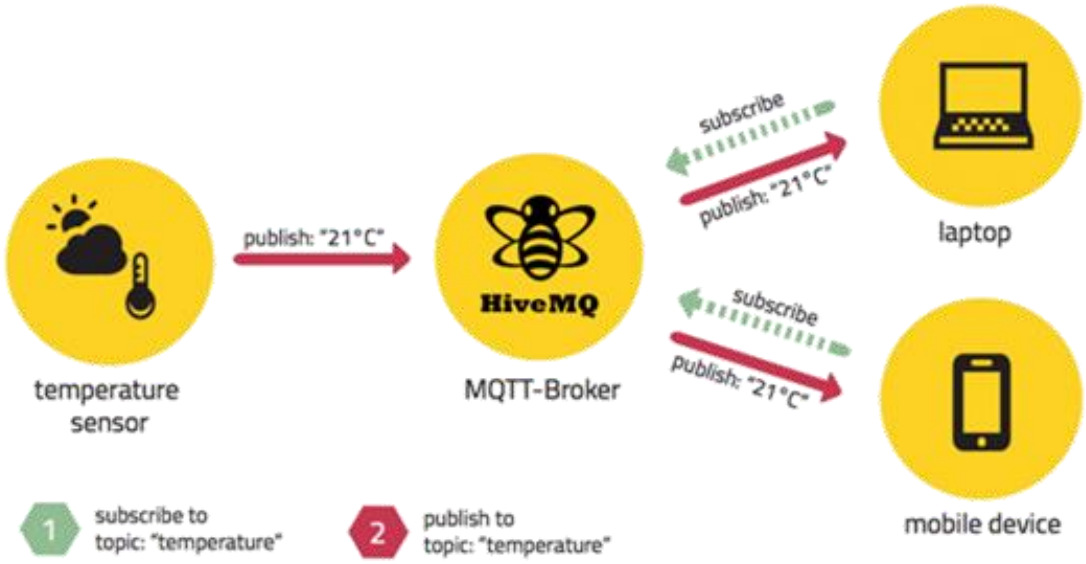
Integrated Satellite-Terrestrial Network: GEO-Based Scenario



- Satellite user terminal provides local area network (WLAN, WPAN) connectivity to IoT devices and other network equipment
- IoT Devices: COTS devices with WPAN or WLAN connectivity
- Space Segment: GEO/MEO
- Frequency Band: FSS
- Relevant Satellite Systems:SES AstraConnect GEO, Eutelsat SmartLNB GEO, Intelsat GEO, Viasat Exede GEO, O3b/O3b NEXT MEO

MQTT: Message Queuing Telemetry Transport

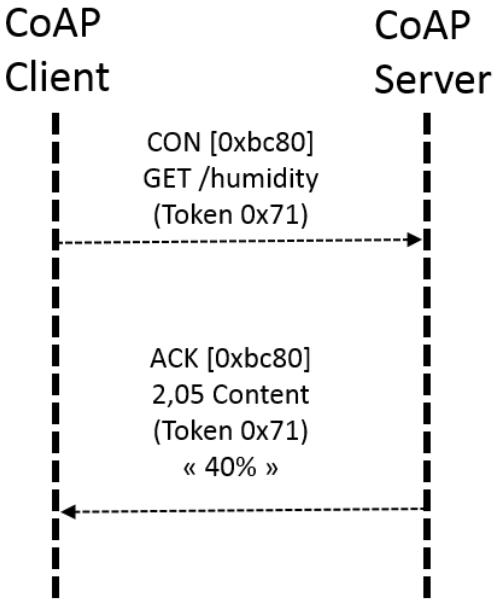
- Standardized at OASIS in 2014
- **Pub/Sub communication pattern**
  - One-to-one, one-to-many message distribution
  - Decoupling producers and consumers
- Lightweight
  - Low header overhead
  - Small client footprint
- Payload agnostic
- **TCP based**: *socket connection oriented*
- Three levels of QoS



Source: HiveMQ

❑ CoAP: Constrained Application Protocol

- ▲ Defined in RFC 7959 by IETF CORE WG
- ▲ Specialized web transfer protocol for constrained devices
- ▲ Main features:
  - Designed for machine- to-machine (M2M) applications
  - Low overhead limiting the need for fragmentation
  - **UDP** [RFC0768] binding
  - **Request/response interaction model**
    - Client & Server model
    - four methods (GET, POST, PUT, DEL)
    - four type messages (CON, NON, ACK, RST)



Outline

1- Satellites for IoT: potential and challenges

2- ESA M2MSAT: Context & Motivation

- IoT-Satellite Hybrid Networks
- IoT Application Protocol: MQTT and CoAP

3- MQTT Protocol Optimization

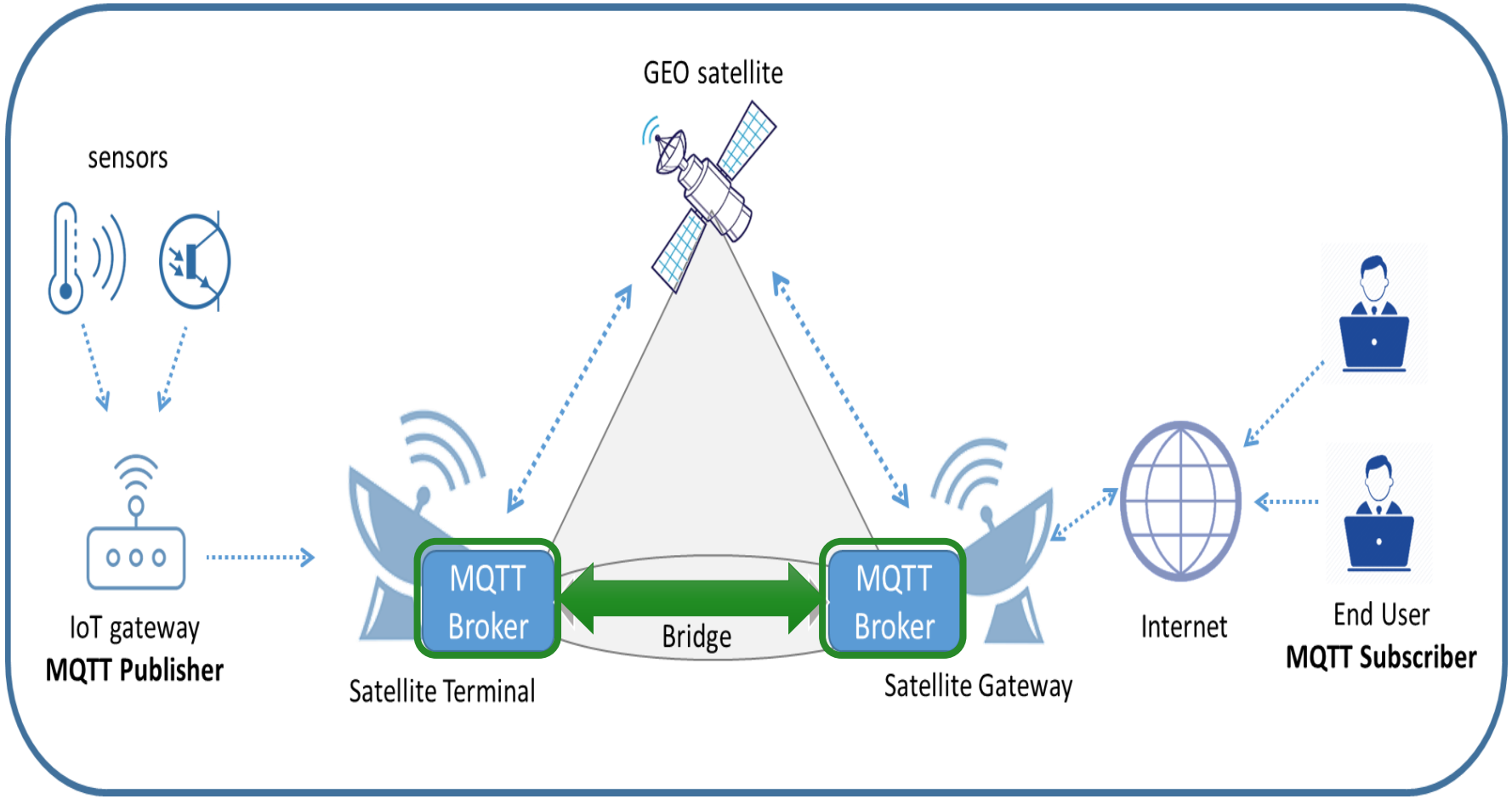
- **Efficient IoT-Satellite Architecture**
- **MQTT-MFA: Aggregation of MQTT Topics**

4- CoAP Protocol Optimization

- Efficient IoT-Satellite Architecture
- Aggregation of unicast CoAP responses to Group Communication Request

M2MSAT Project

MQTT EFFICIENT Configuration in the GEO-based scenario



M2MSAT Project

MQTT Optimization: Aggregation of MQTT Topics

- **Broker at the ST**
  - Support of aggregation / compression of IoT data
- **Broker at the GW**
  - Support of aggregation of multiple MQTT requests coming from several subscribers
  - Support of caching: the broker can make (cached) resources available to future subscribers interested in the same cached resource
    - Reduce delay in retrieving the data
    - Save bandwidth over the satellite return channel (no need to request and exchange the same data again over the satellite link)

M2MSAT Project

MQTT Optimization: Aggregation of MQTT Topics

- **Background**: MQTT standard supports `topic pattern` for the Bridge to request different topics within the same MQTT message

```
topic pattern [[[ out | in | both ] qos-level] local-prefix remote-prefix]
.
```

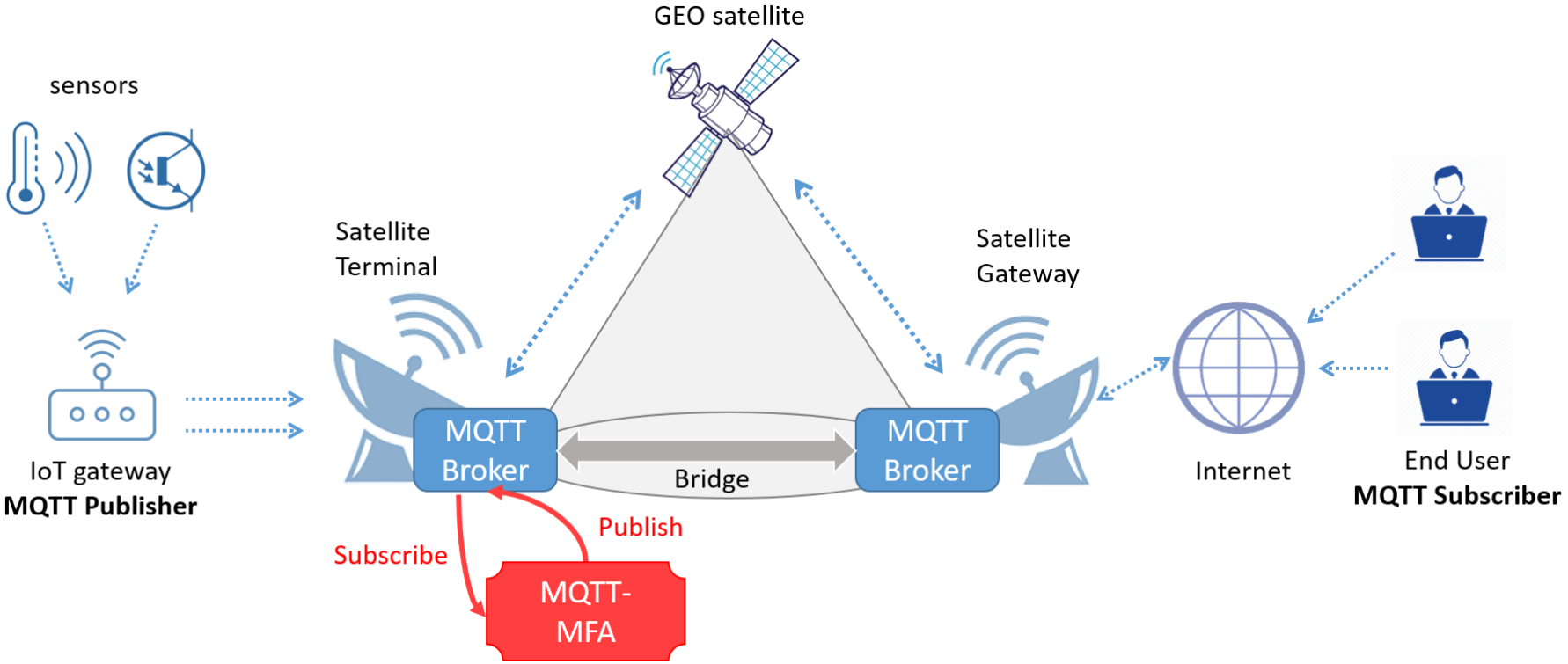
  - The second parameter defines the direction that the messages will be shared in, so it is possible to import messages from a remote broker using *in*, export messages to a remote broker using *out* or share messages in both directions. If this parameter is not defined, the default of *out* is used.
  - The qos-level defines the publish/subscribe QoS level used for this topic and defaults to 0.

- **Optimization for the satellite scenario:** aggregation of MQTT topics matching the pattern within the same response

- **Goal:** reduce the amount of traffic exchanged on the return link of the satellite segment

M2MSAT Project

MQTT MESSAGE FILTER AGGREGATOR (MQTT-MFA)

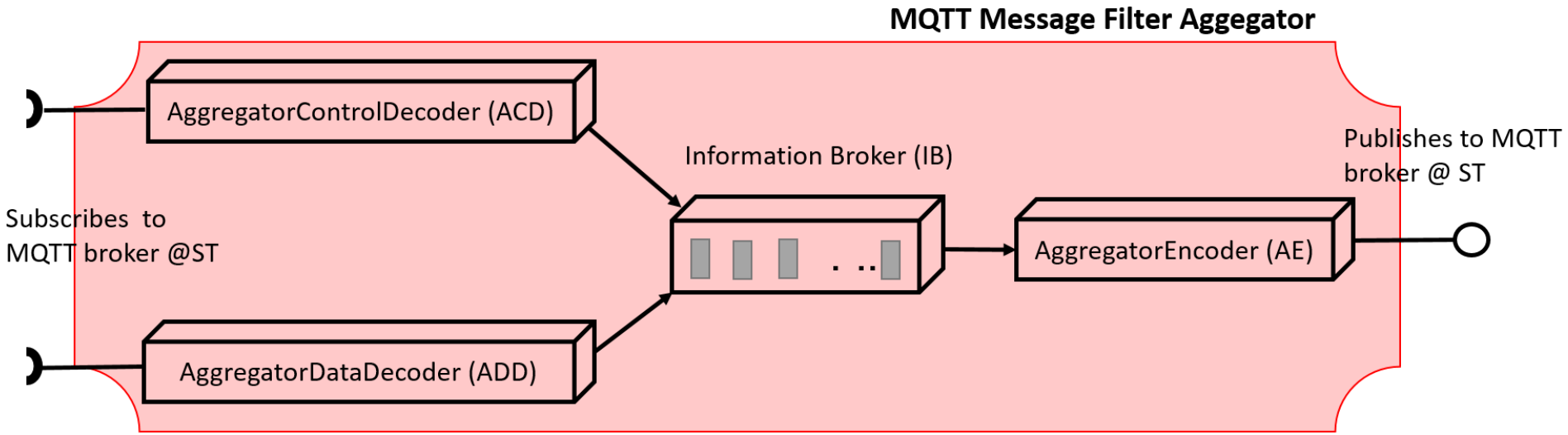


- **MQTT Message Filter:** node connected to the Broker
  - implementation **independent** by Mosquitto development
  - Compliant with the MQTT std: the broker should not modify the msg content (payload)
  - Flexibility: possibility of having multiple MQTT Msg filters

M2MSAT Project

MQTT MESSAGE FILTER AGGREGATOR (MQTT-MFA)

- an **AggregatorControlDecoder (ACD)**: subscribes to a MQTT-Broker. It decodes control messages (e.g. requests) and prepares a transaction for the response. It is responsible of the mapping between the request and the response.
- an **AggregatorDataDecoder (ADD)**: subscribes to a MQTT-Broker. Decodes the payload of incoming messages generated by the subscribed sources, and extracts the information to be kept.
- an **Information Broker (IB)**: a named message queue passing messages from the decoders to the endcoder .
- an **AggregatorEncoder (AE)**: caches all data and answers the requests. Publish the new encoded messages to the MQTT-Broker.



M2MSAT Project

MQTT MESSAGE FILTER AGGREGATOR (MQTT-MFA)

- MQTT-MFA SW Implementation
  - MQTT-MFA is written in Python 3.6
  - Tested under Linux (Ubuntu 18.04) and Window 10 with the following packages
    - MQTT Broker Mosquitto 1.5
    - MQTT Client Paho-mqtt >= 1.3.1

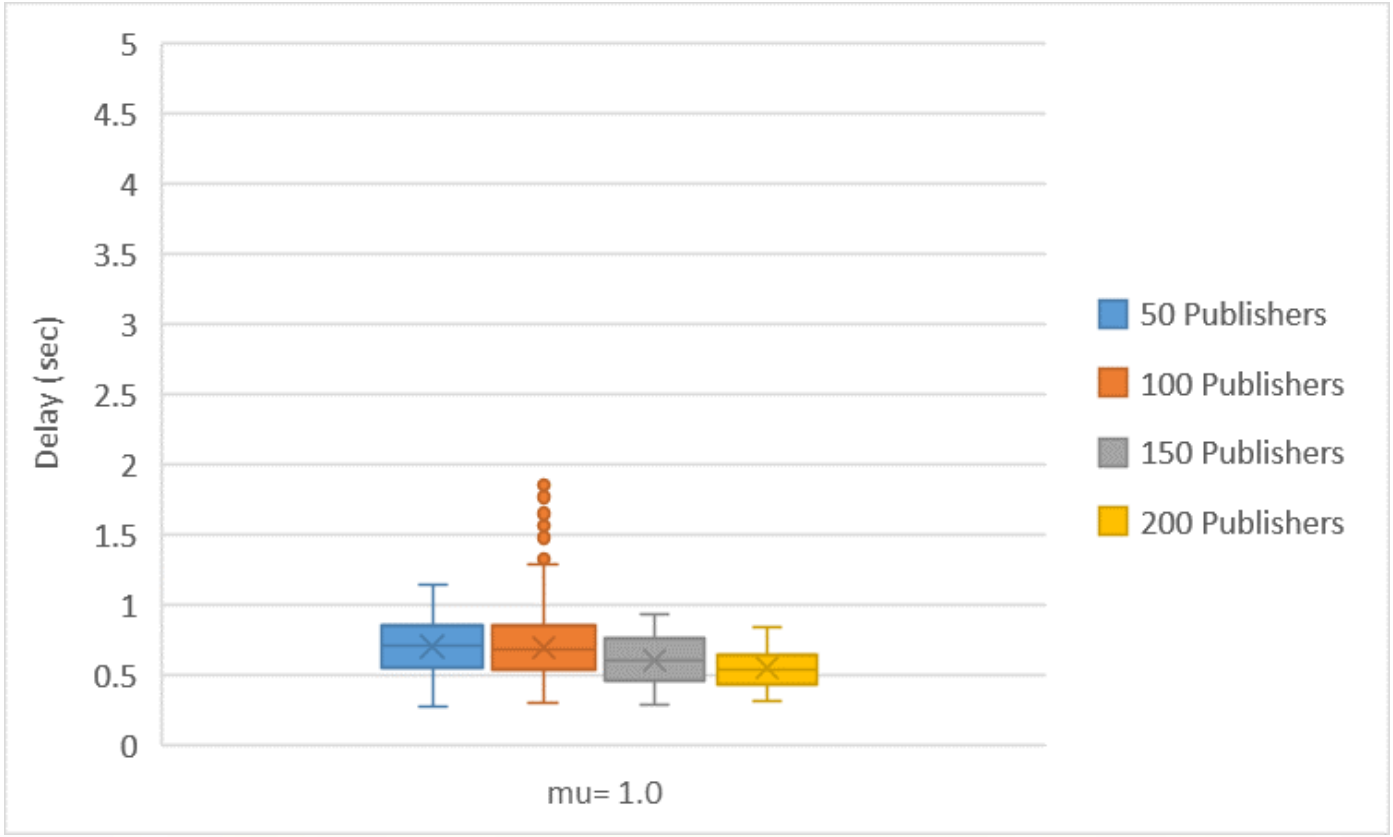
- MQTT-MFA is integrated in the Integrated satellite terrestrial testbed

	Forward band	Return band
Standard	DVB-S2	DVB-RCS2
Bandwidth	36 MHz	200 KHz
Roll-off	0,2	0,25
C/N	20 dB (UL) 17.5 dB (DL)	7 dB (UL) 20 dB (DL)

M2MSAT Project

MQTT MESSAGE FILTER AGGREGATOR (MQTT-MFA)

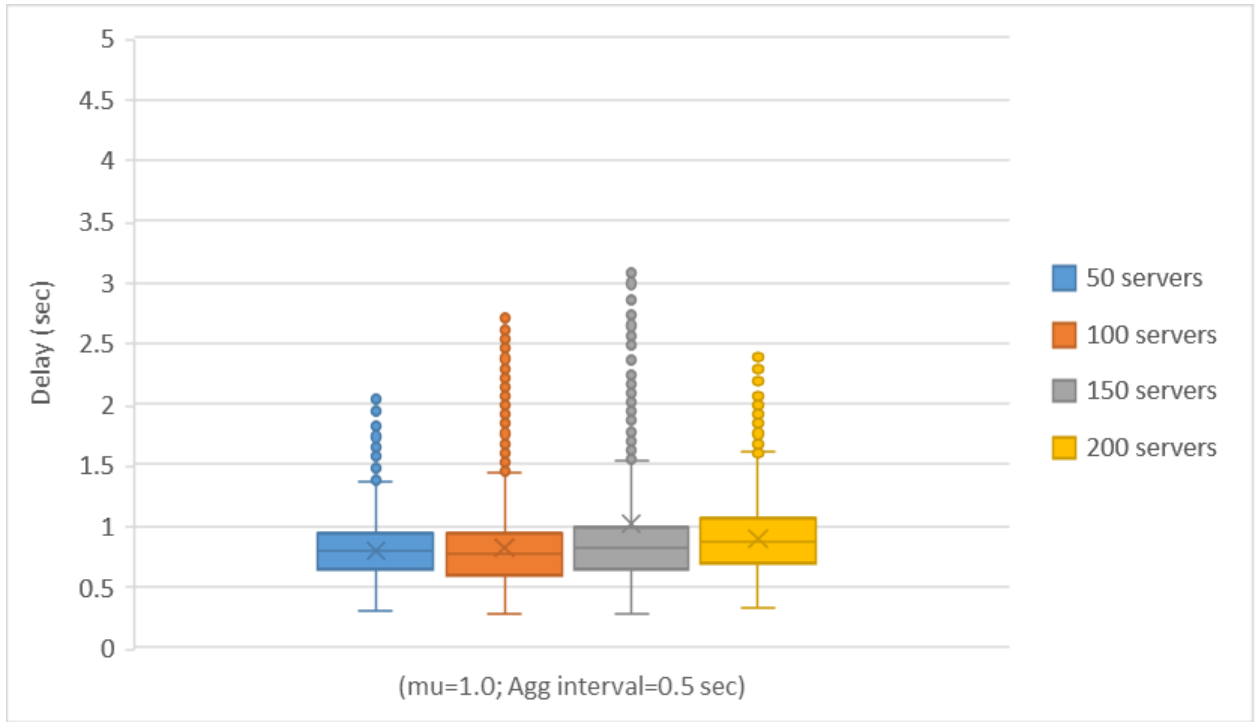
- MQTT-MFA: Performances Evaluation:
  - **Delay of MQTT topics collection Without Aggregation**

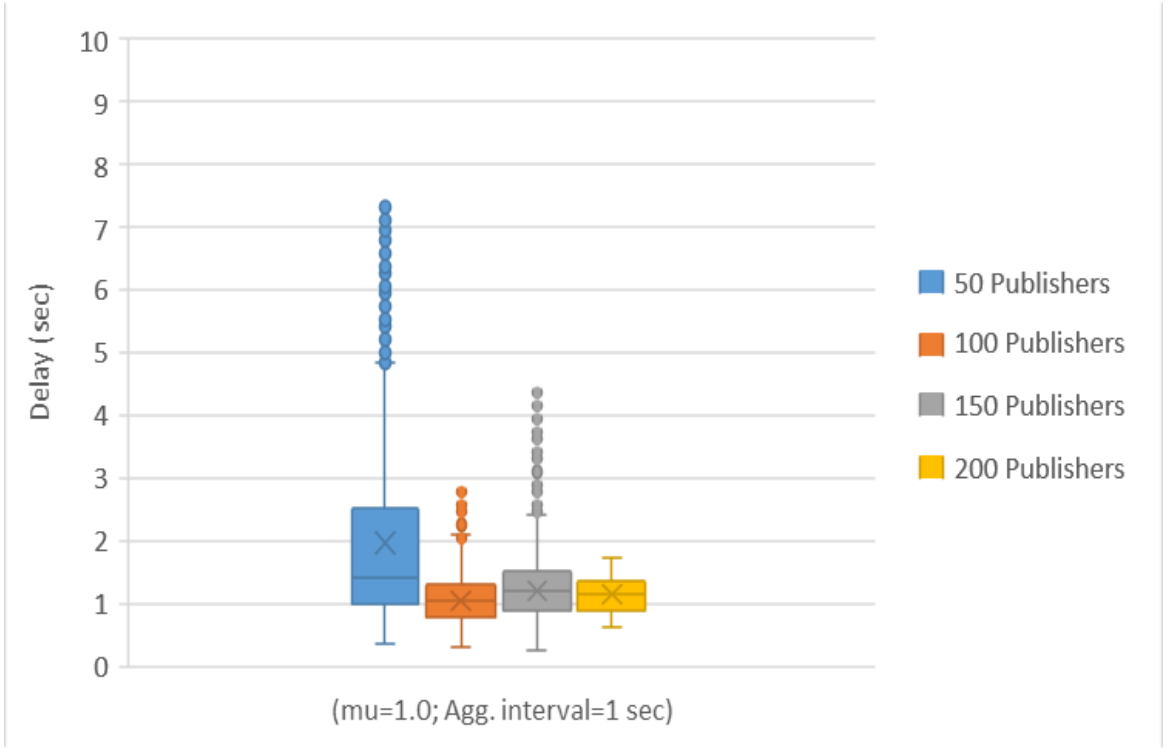


M2MSAT Project

MQTT MESSAGE FILTER AGGREGATOR (MQTT-MFA)

- MQTT-MFA: Performances Evaluation:
  - **Delay of MQTT topics collection With Aggregation**
    - Different Aggregation intervals

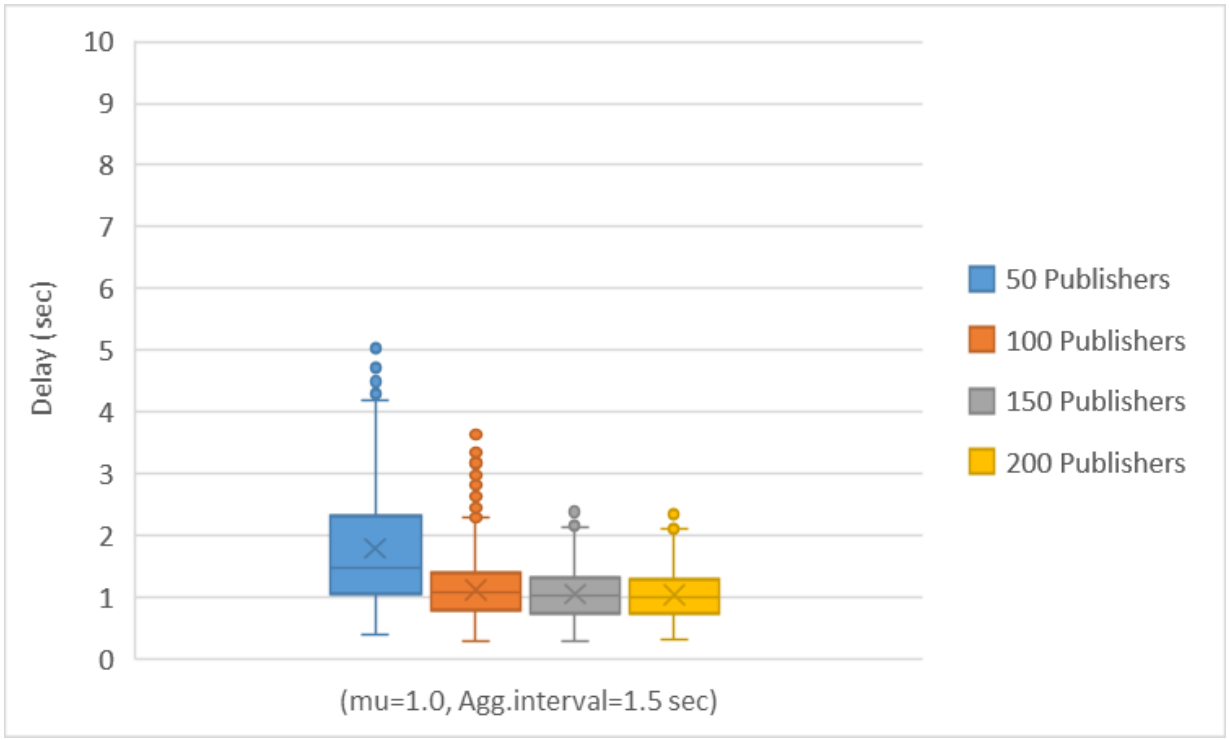


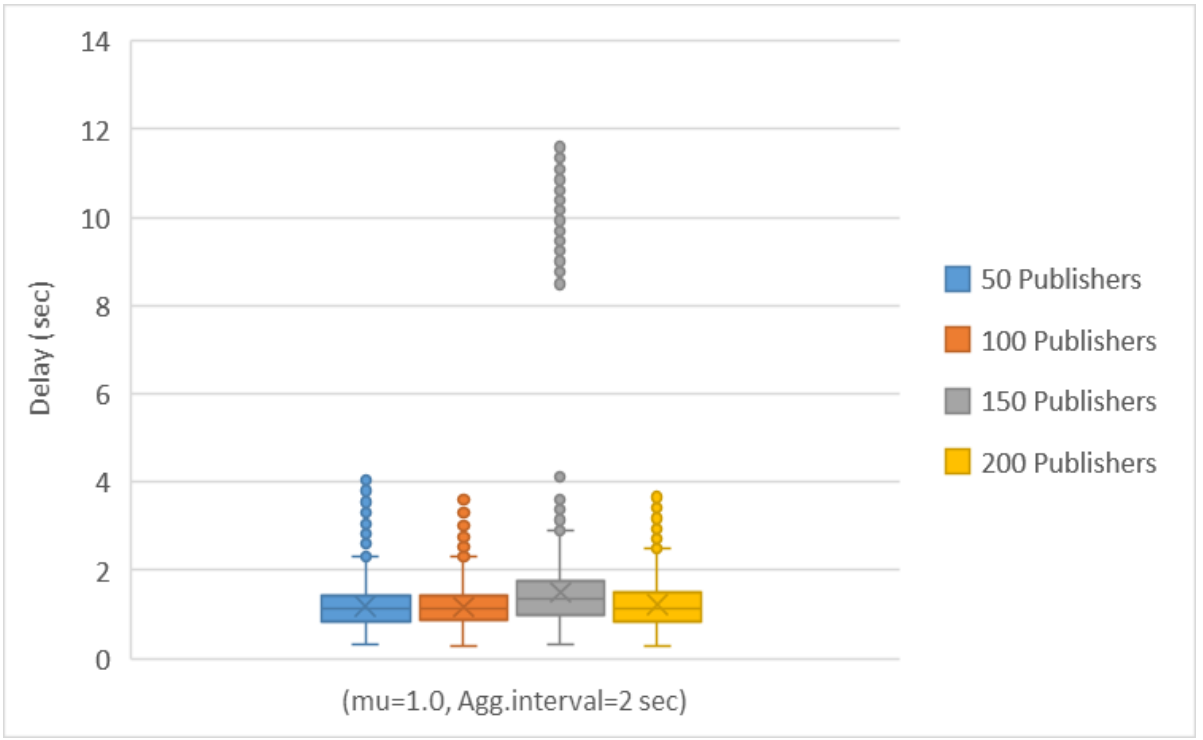


M2MSAT Project

MQTT MESSAGE FILTER AGGREGATOR (MQTT-MFA)

- MQTT-MFA: Performances Evaluation:
  - **Delay of MQTT topics collection With Aggregation**





M2MSAT Project

MQTT MESSAGE FILTER AGGREGATOR (MQTT-MFA)

❑ Conclusion & Lessons learnt

- ▲ **This optimization allows an end-user to receive multiple topics in one message sent through the satellite return link**

→ Less traffic load on the satellite return link for a SAT operator

→ Faster delivery time for an end-user

- ▲ **It reduces the overhead on the satellite return link for a SAT operator:**
  - ▲ No need to establish a connection for every requested topic since the bridge supports topic pattern and hence all topics that match the pattern are being bridged
  - ▲ The aggregation of several MQTT topics requested by the MQTT subscriber alleviates the high connection establishment overhead for the exchange of a small amount of data.

Outline

1- Satellites for IoT: potential and challenges

2- ESA M2MSAT: Context & Motivation

- IoT-Satellite Hybrid Networks
- IoT Application Protocol: MQTT and CoAP

3- MQTT Protocol Optimization

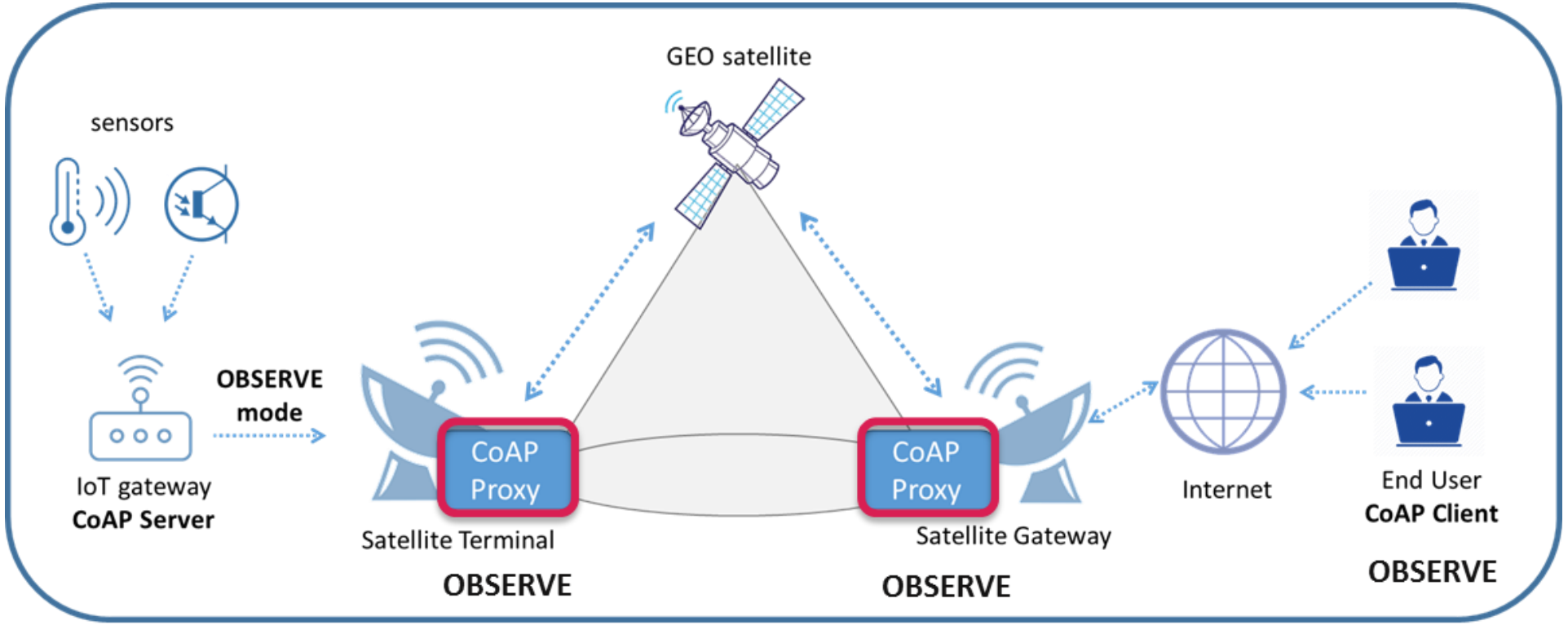
- Efficient IoT-Satellite Architecture
- MQTT-MFA: Aggregation of MQTT Topics

4- CoAP Protocol Optimization

- **Efficient IoT-Satellite Architecture**
- **Aggregation of unicast CoAP responses to Group Communication Request**

M2MSAT Project

CoAP Optimization: Aggregation of Unicast Responses to Group Communication Request



- CoAP Observe
- Proxy to Proxy over the satellite segment

M2MSAT Project

CoAP Optimization: Aggregation of Unicast Responses to Group Communication Request

- **Reverse Proxy at the GW:** it registers to the forward proxy on behalf of the client.
  - This registration can be re-used for multiple clients, asking for the same resource.
- **Forward Proxy at the ST:** For each resource updated, only one notification is sent from the ST to the GW and distributed to all registered clients
- **Caching** can be enabled at the reverse proxy:
  - Reduce delay in retrieving the data
  - Save bandwidth over the satellite return channel (no need to request and exchange the same data again over the satellite link)

M2MSAT Project

CoAP Optimization: Aggregation of Unicast Responses to Group Communication Request

- **Background**: group communication (RFC7390) allows to request the different resources within a single CoAP request
- **Problem statement**: CoAP responses are sent in separated unicast messages from each single CoAP server to the CoAP Client
- **Optimization for the satellite scenario:** aggregation of unicast responses in a single message
- **Goal:** reduce the amount of traffic on the return link of the satellite segment

M2MSAT Project

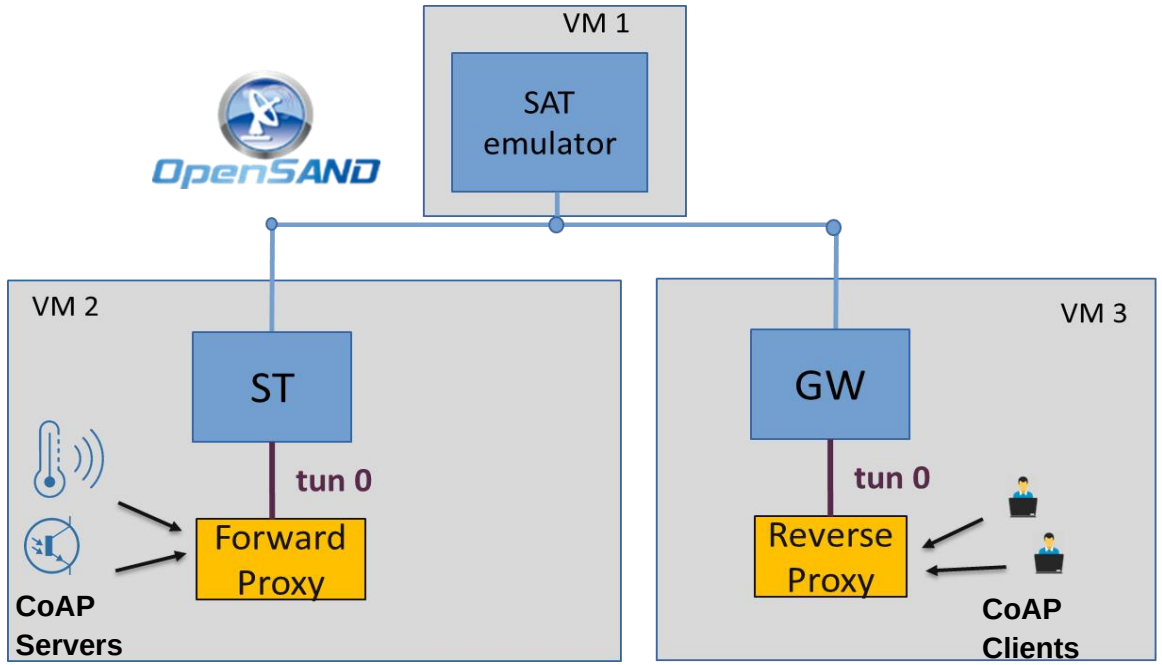
CoAP Optimization: Aggregation of Unicast Responses to Group Communication Request

- **Two steps solutions** is proposed:
  - **First phase***: servers discovery with a multicast GET request*
    - The list of discovered servers is cached to reduce the discovery overhead
  - **Second phase**: *unicast request to every server and data aggregation*
    - Aggregation parameters: `maximum number of servers` and `waiting time_window`
    - Aggregated responses can be cached as normal CoAP messages
    - Compression is supported
- Different groups can be distinguished by different multicast addresses and ports

M2MSAT Project

CoAP Optimization: Aggregation of Unicast Responses to Group Communication Request

- **SW implementation**
  - CoAPthon v4.01
  - CoAPthon adopts the default values of the CoAP standard.
  - Integrated satellite terrestrial testbed

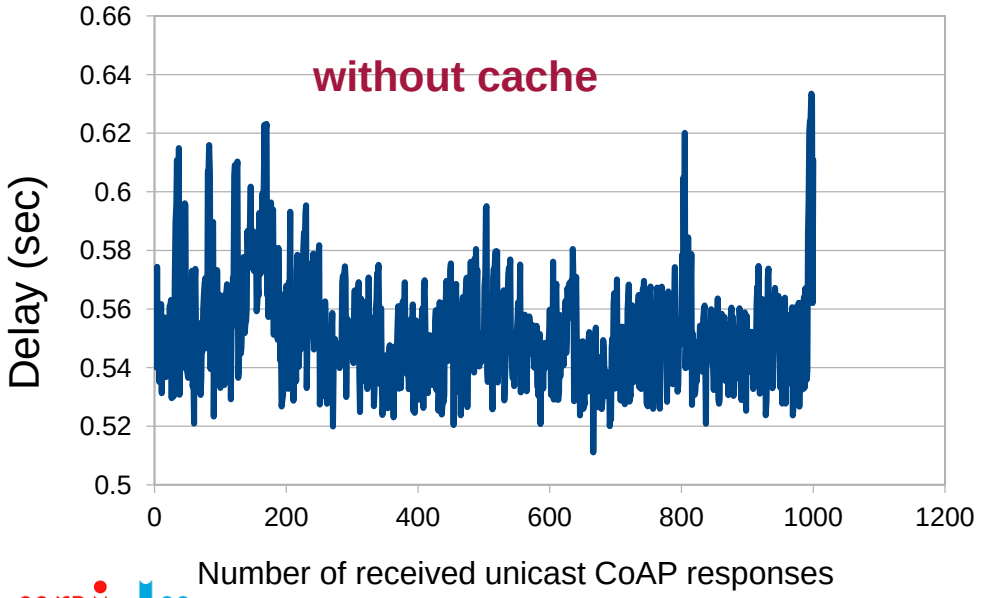


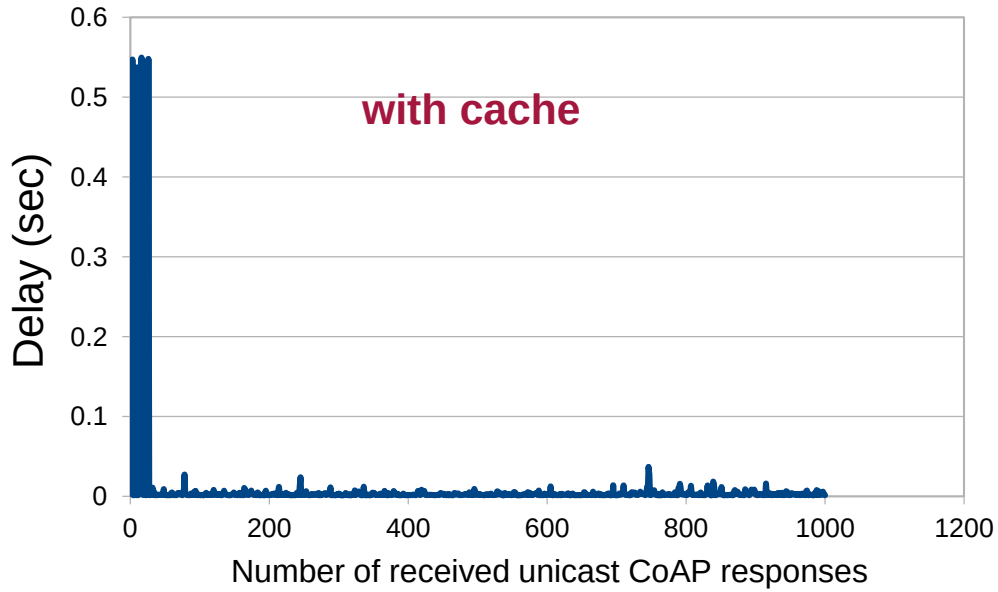
M2MSAT Project

CoAP Optimization: Aggregation of Unicast Responses to Group Communication Request

- Performance Evaluation
  - **GET Request:**
    - KPI: Delay

- 10 Clients
- 100 CoAP requests/client
- **With Std. group** Communication



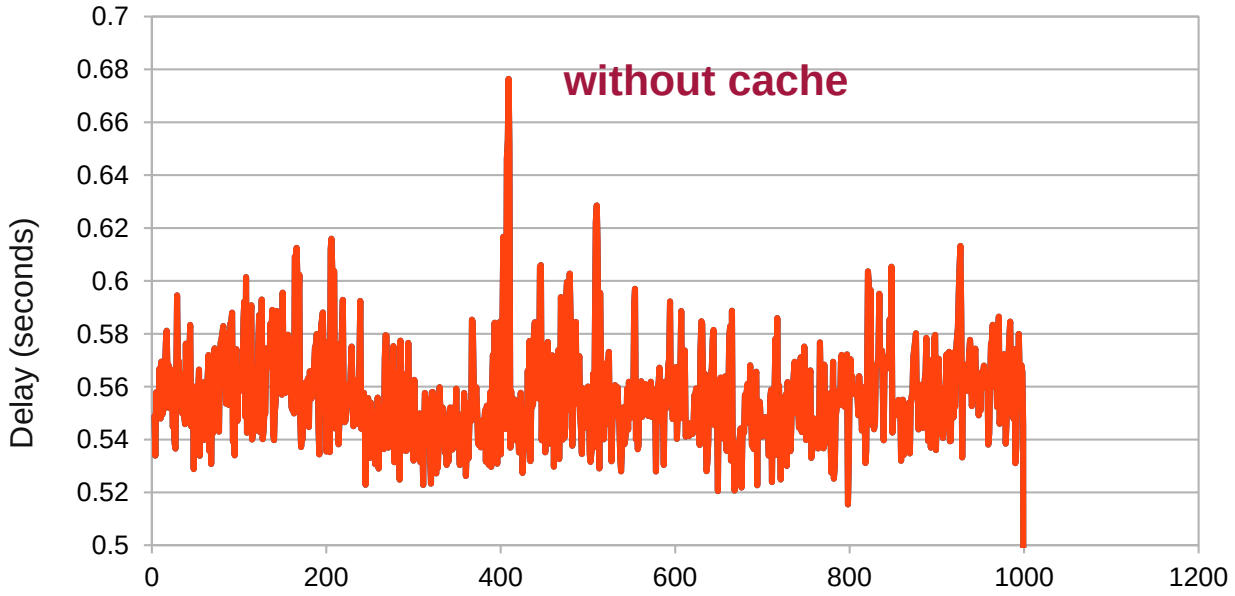


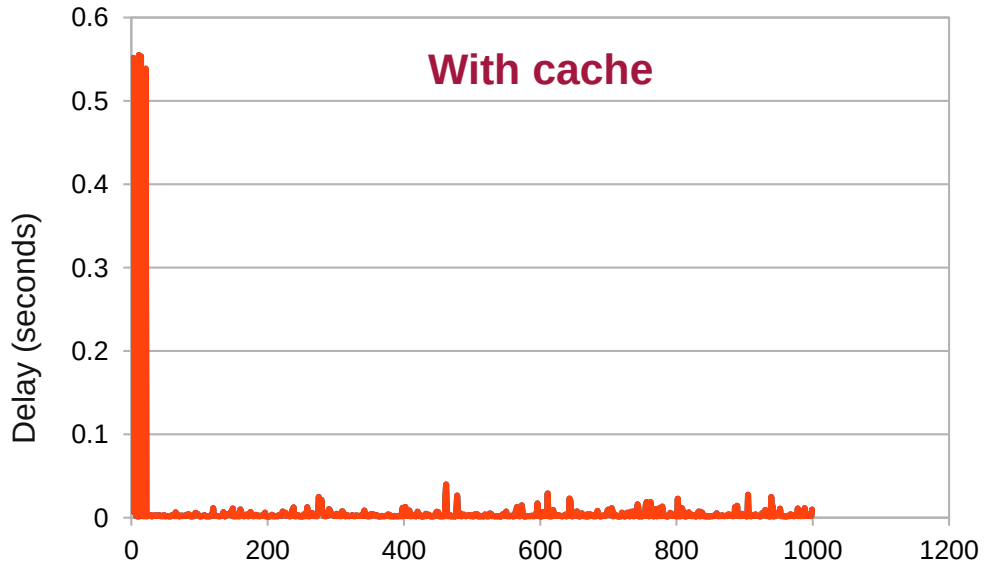
M2MSAT Project

CoAP Optimization: Aggregation of Unicast Responses to Group Communication Request

- Performance Evaluation
  - **GET Request:**
    - KPI: Delay

- 10 Clients
- 100 CoAP requests/client
- **With Aggregation group** Communication

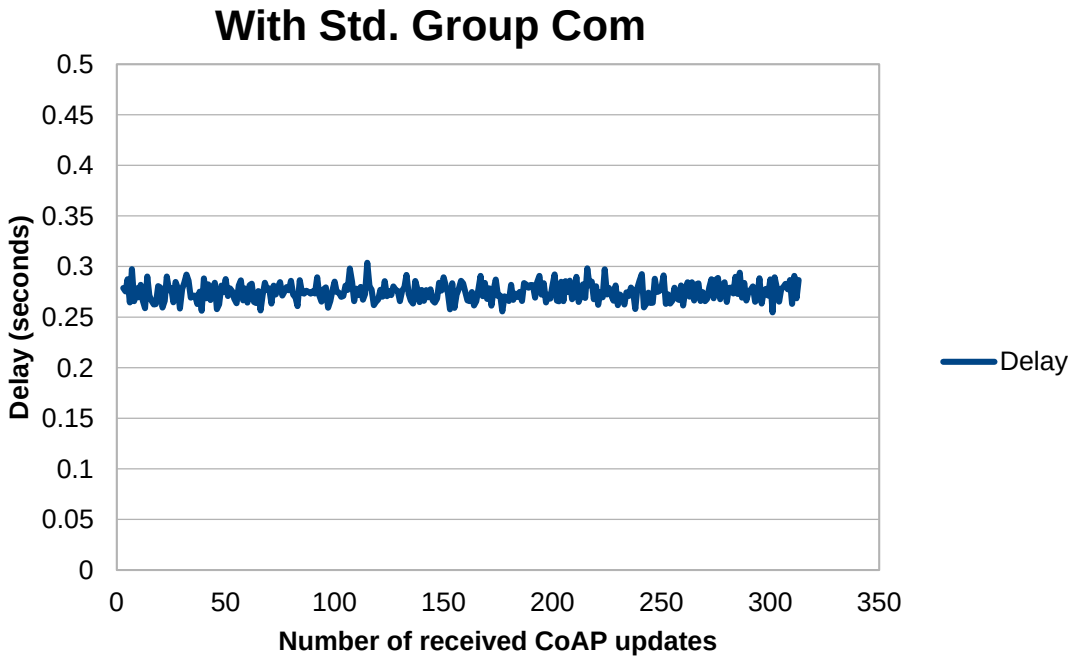


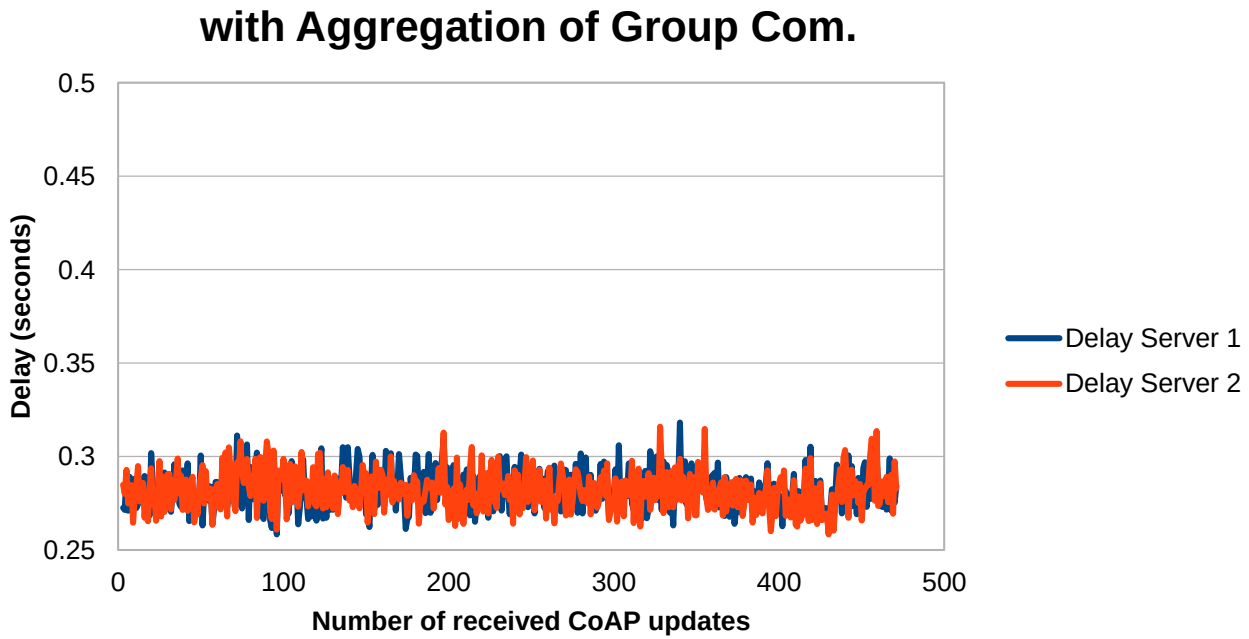


M2MSAT Project

CoAP Optimization: Aggregation of Unicast Responses to Group Communication Request

- Performance Evaluation
  - **OBSERVE Request:**
    - KPI: Delay
  - 10 Clients
  - 100 CoAP requests/client





M2MSAT Project

Aggregation of CoAP unicast responses

❑ Conclusion & Lessons learnt

- The use of proxies combined with caching reduces the delay needed to collect aggregated data by factor of 10 when several clients are requesting different topics.

- An end-user may use this optimization to query in an efficient way a single subnet or multi-subnet of constrained devices.

- The discovery of CoAP servers introduces an additional delay to CoAP responses gathering.

  - ➢ End users can tune aggregation parameters: maximum number of servers and waiting time_window, based on the requirements of their application