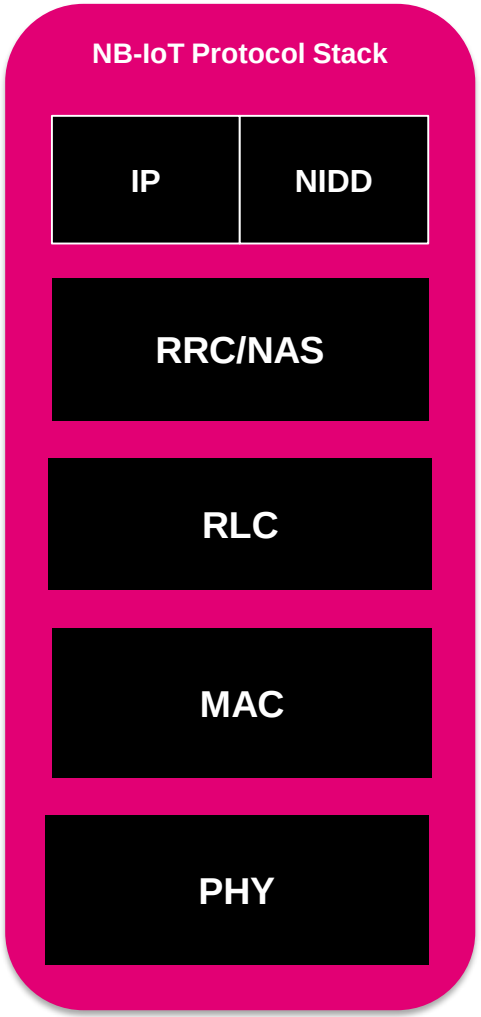
NB-IoT Data Delivery Study

Jeff Ahmet & Anokhi Shah

Overview and Objective – Quantify Data Delivery for NB-IoT

- **IP vs Non-IP Data delivery**
  - IP: TCP, UDP, CoAP and MQTT
  - Non-IP: SMS and Non-IP (NIDD)
- **Quantify Transaction and Payload Delta for each protocol at varying user data message sizes**
  - Single packet transmission
  - 80 bytes and 5 bytes
  - Quantify the Physical Layer (L1) payload and transaction overhead
  - Quantify the application layer payload and transaction counts for rate plan verification

Summary of Results



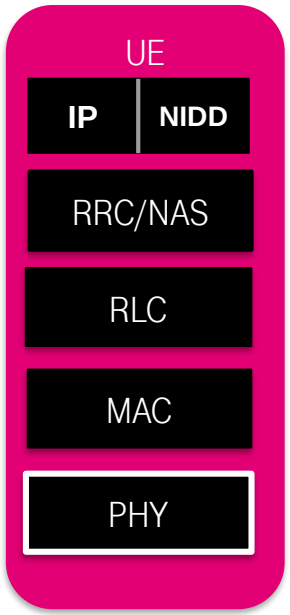
IP Data Delivery

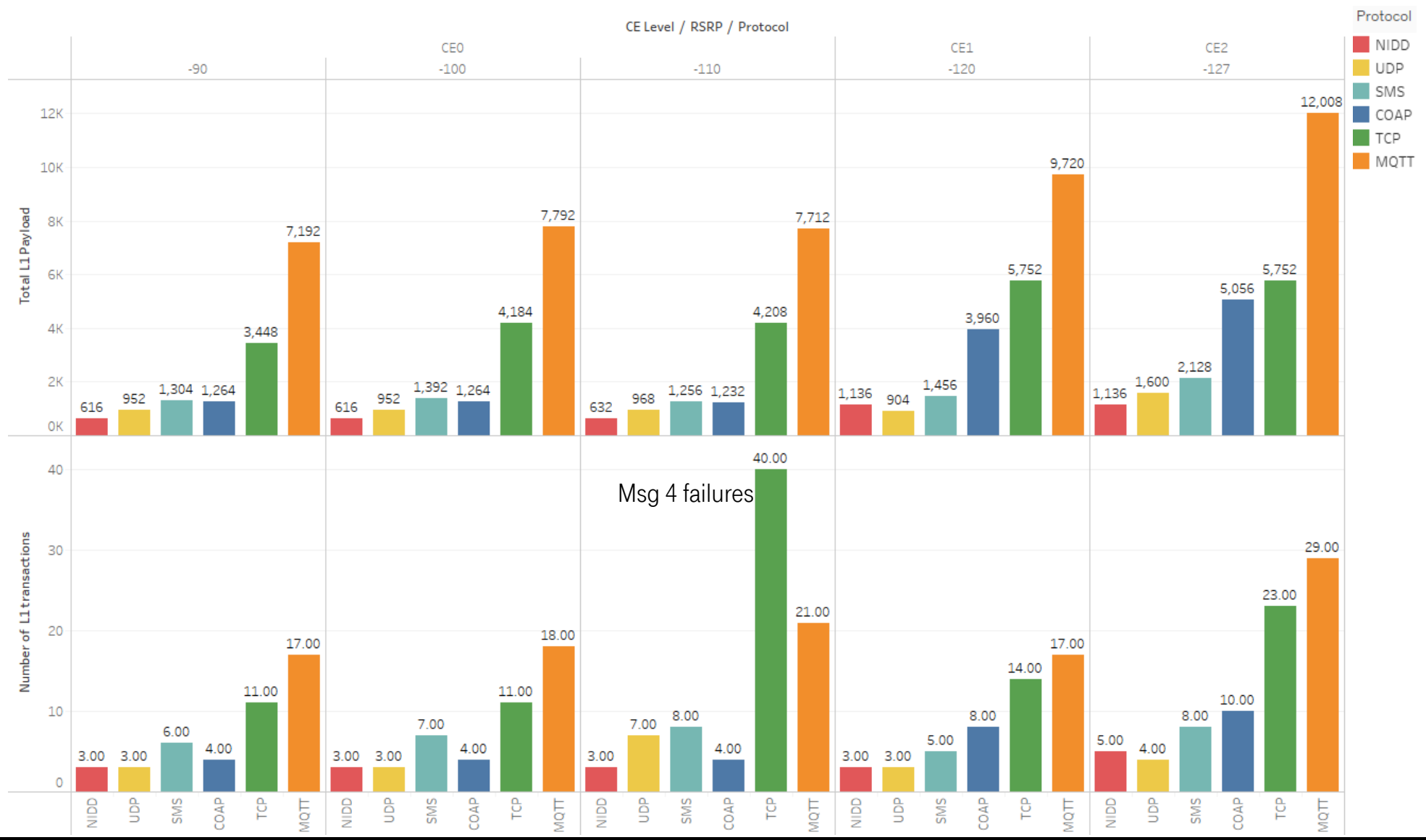
- **MQTT and TCP are chatty**
  - Air Interface overhead
    - *5x the transactions*
    - *10x payload increase*
  - Application Layer
    - *10x the transactions over Non-IP*
    - *100x the payload for 5 byte tx*
- **UDP is more efficient but less robust**
  - No Retransmission to ensure data went out successfully over the air
- **CoAP is the happy medium**
  - Built in retransmission mechanism in the app layer

Non- IP Data Delivery

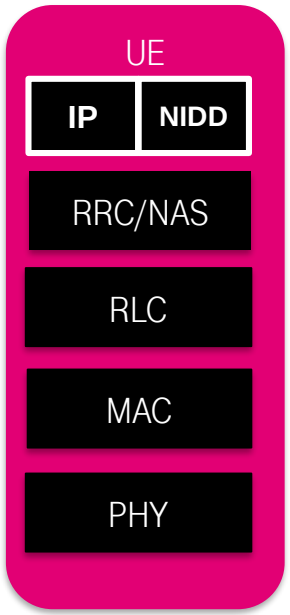
- **SMS**
  - Lacks control from the DDI
  - Retransmission at the NAS layer – CP Ack
  - Lower transmission overhead
- **NIDD is the most efficient for low data**
  - No Retransmission mechanism at the protocol layer
  - Similar to UDP

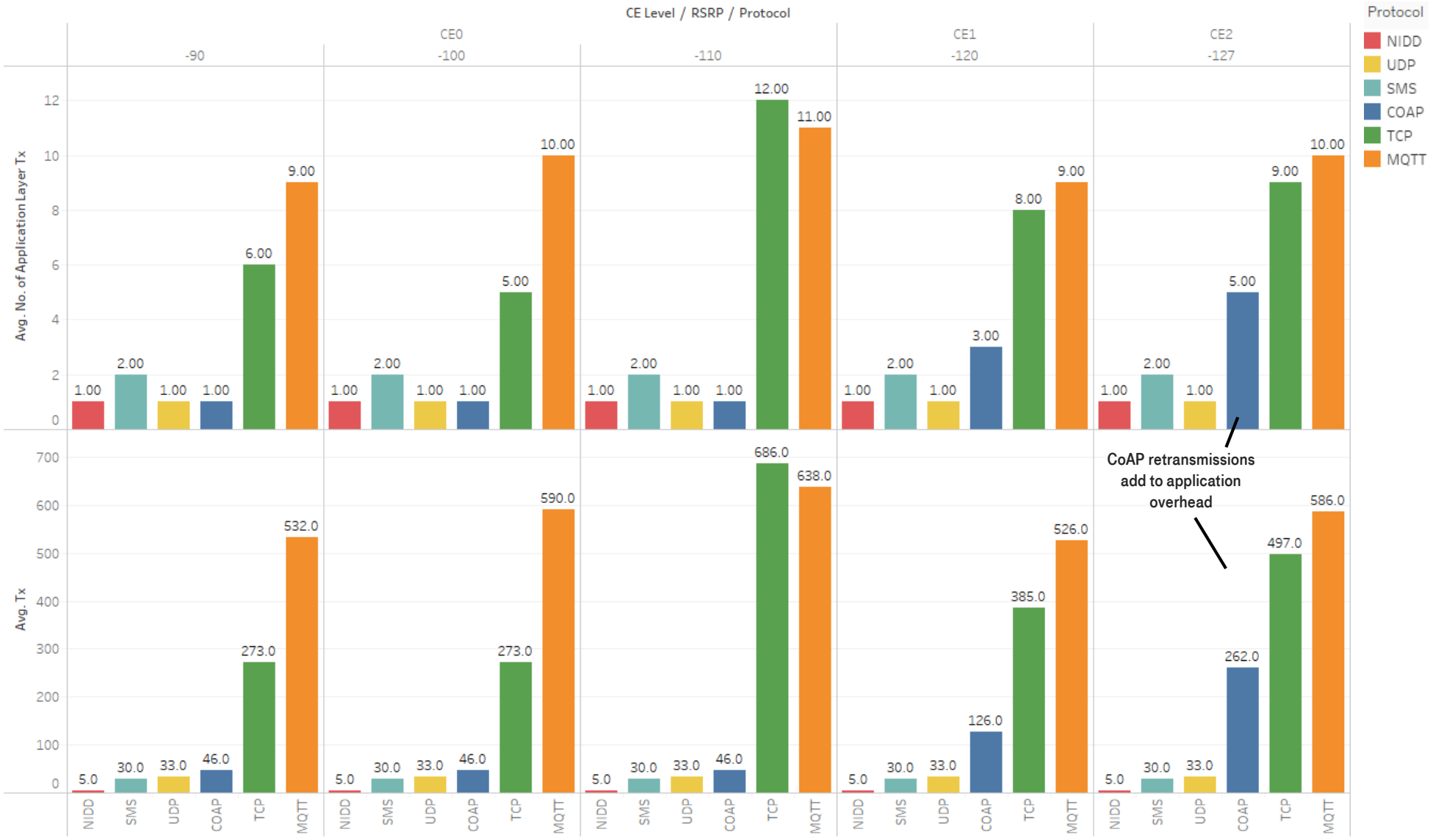
5 Byte - Air Interface Payload and Transaction Overhead



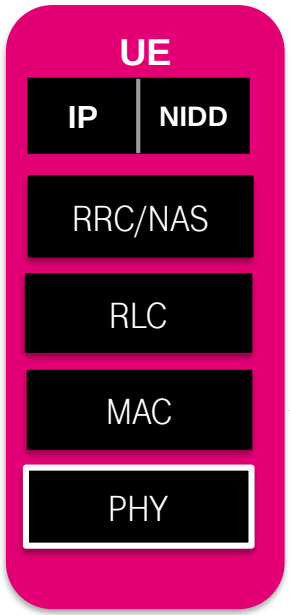


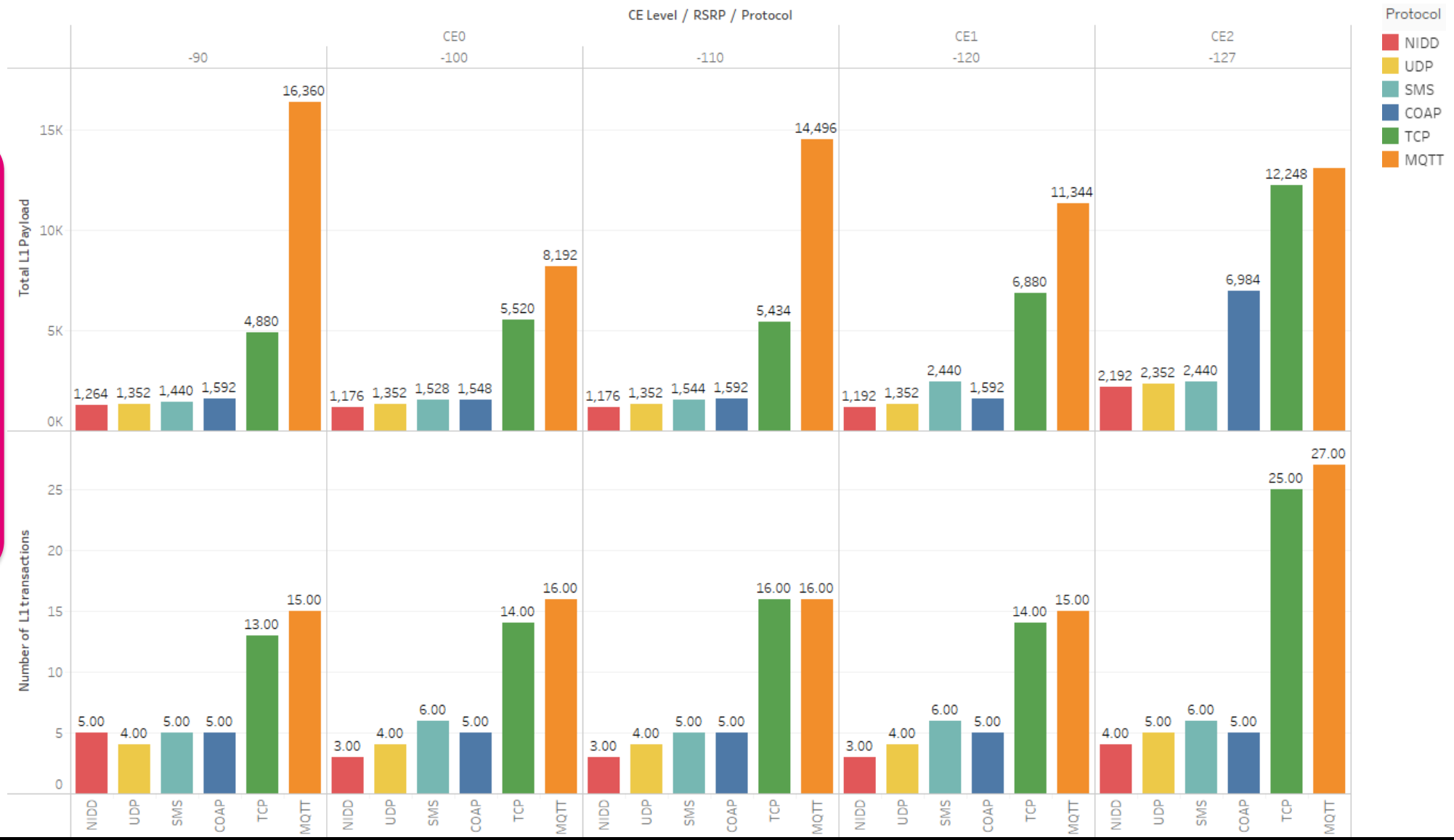
5 Byte - Application Layer Payload and Transaction Summary



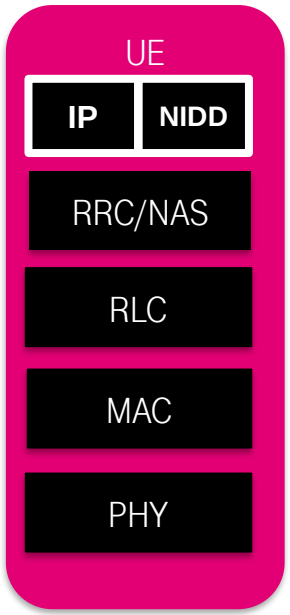


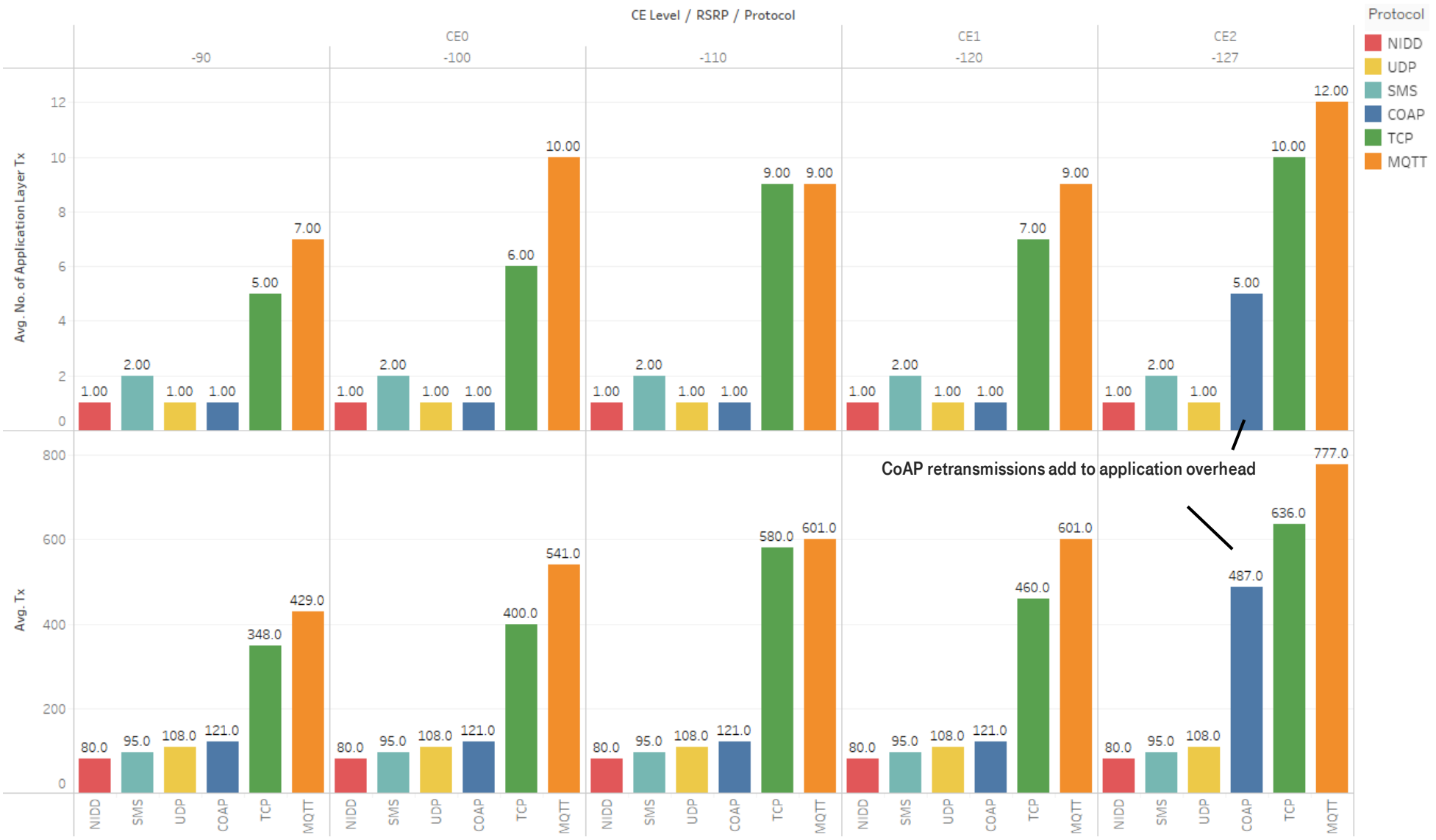
80 Byte - Air Interface Utilization





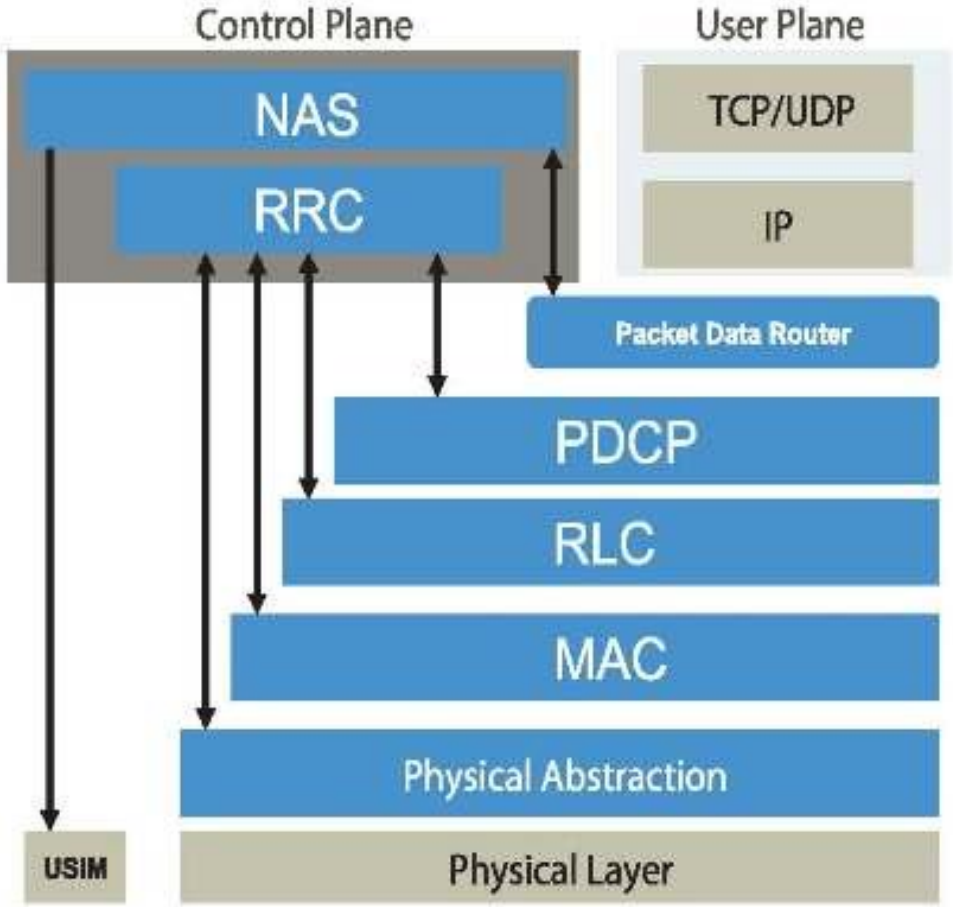
80 Byte - Application Layer Payload and Transaction Summary



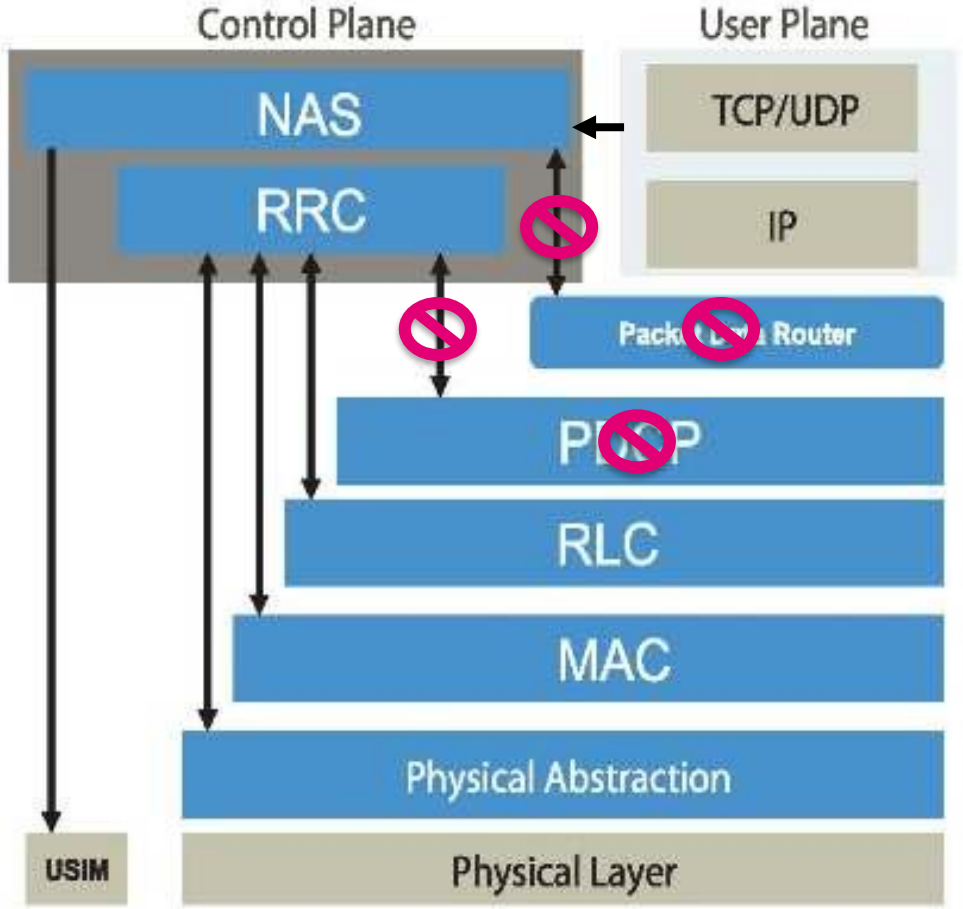


Protocol Stack

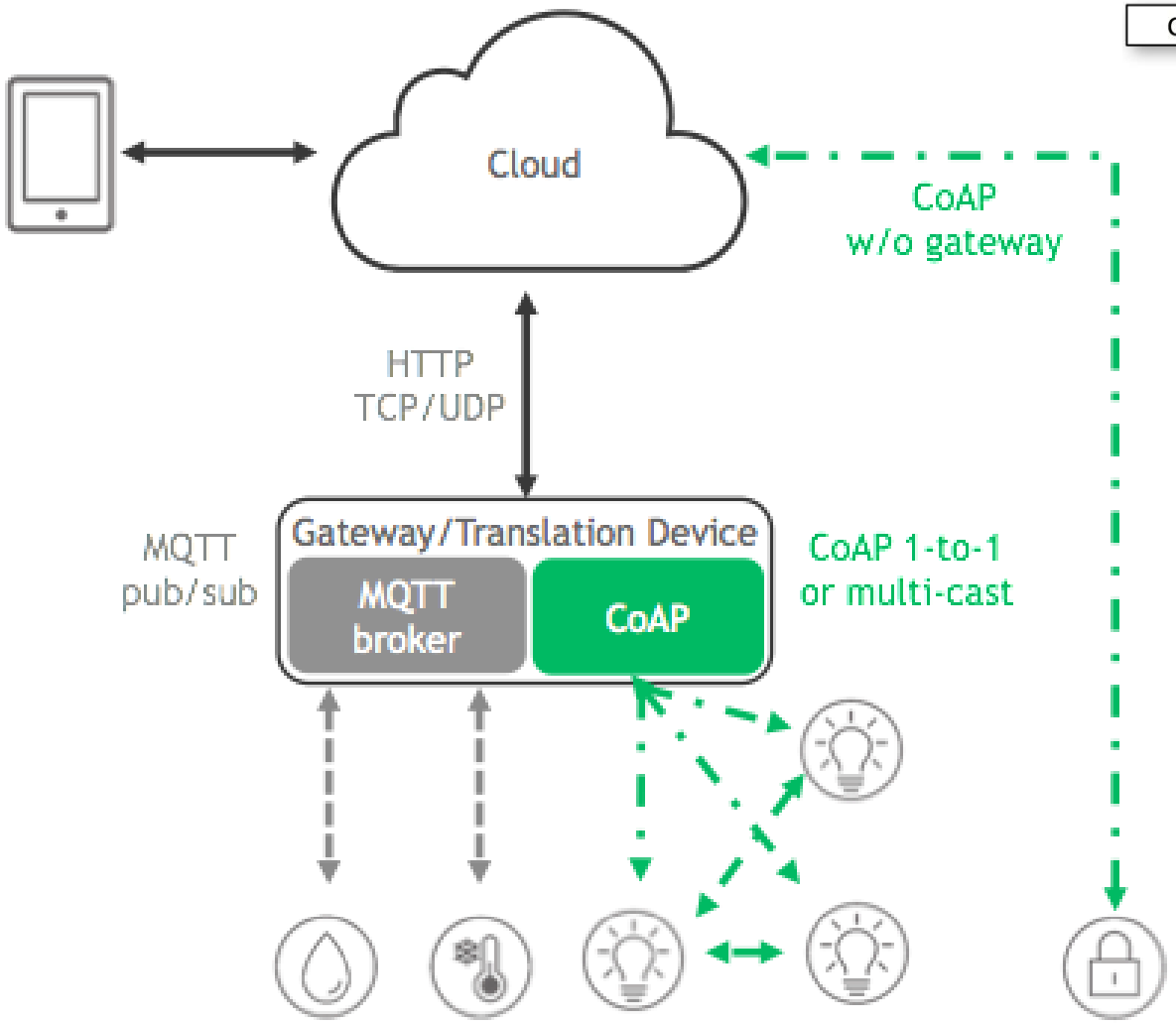
LTE Protocol Stack

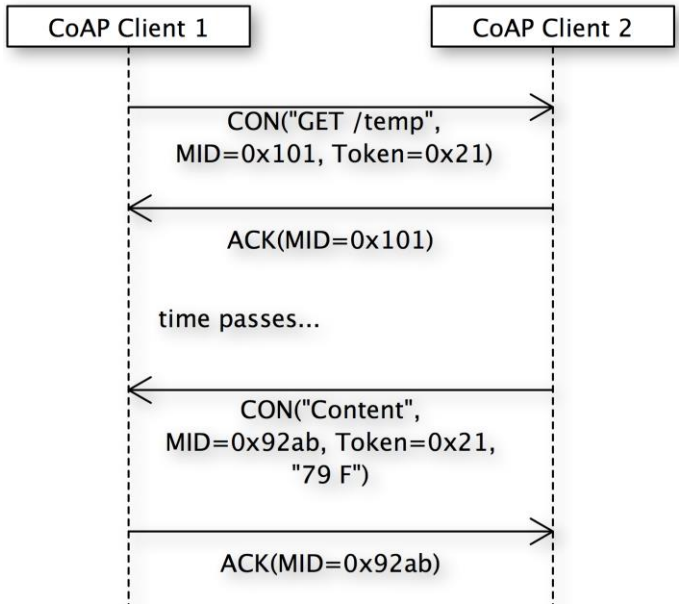


NB-IoT Protocol Stack



Constrained Application Protocol

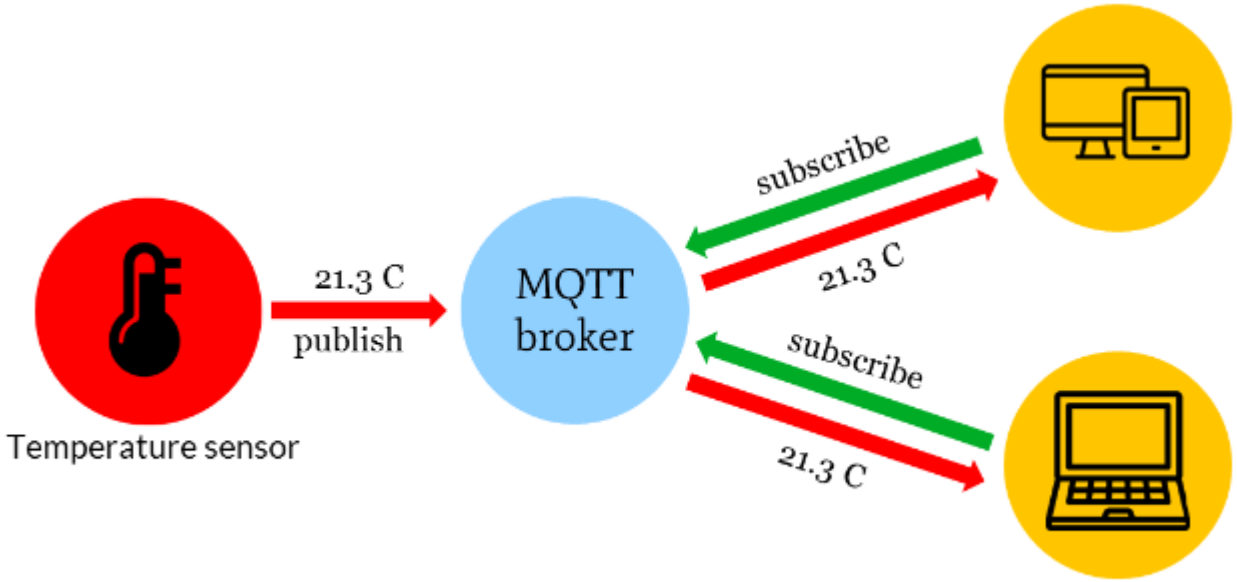




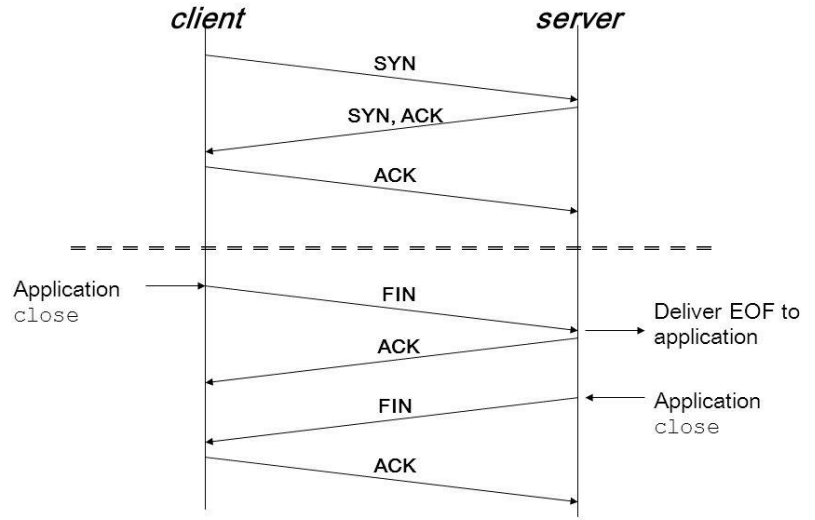
CoAP is a UDP based protocol

- Can be configured for acknowledgements at the app layer
  - Confirmable or non-confirmable packets
- Can be used in a client/server or client to client environment
- Better suited to NB-IoT technology

MQTT has built-in ‘convenience’ for app developers but not for NB-IoT



Schematic data flow from sensor (machine) to devise (machine)

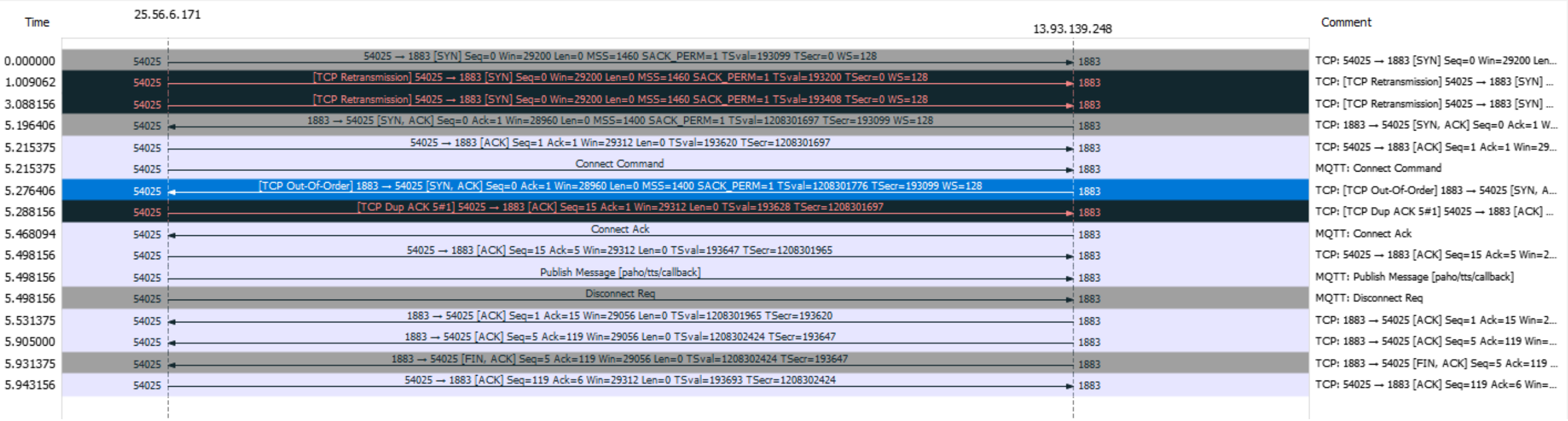


32

Time	25.56.6.171		13.93.139.248	Comment
0.000000	56911	SYN	1883	Seq = 0
0.768594	56911	SYN, ACK	1883	Seq = 0 Ack = 1
0.786563	56911	ACK	1883	Seq = 1 Ack = 1
0.787407	56911	PSH, ACK - Len: 14	1883	Seq = 1 Ack = 1
1.394157	56911	ACK	1883	Seq = 1 Ack = 15
1.418594	56911	PSH, ACK - Len: 4	1883	Seq = 1 Ack = 15
1.449344	56911	ACK	1883	Seq = 15 Ack = 5
1.449344	56911	PSH, ACK - Len: 101	1883	Seq = 15 Ack = 5
1.449344	56911	FIN, PSH, ACK - Len: 2	1883	Seq = 116 Ack = 5
2.073594	56911	ACK	1883	Seq = 5 Ack = 116
2.199313	56911	FIN, ACK	1883	Seq = 5 Ack = 119
2.210219	56911	ACK	1883	Seq = 119 Ack = 6

- 3 way TCP handshake increases transmission overhead
- Retransmissions built into TCP for robust transmission
- Publish/Subscriber broker model works for cloud based providers

TCP retransmissions eat into NB-IoT air interface resources



Time	25.56.6.171		13.93.139.248	Comment
0.000000	54025	54025 → 1883 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=193099 TSecr=0 WS=128	1883	TCP: 54025 → 1883 [SYN] Seq=0 Win=29200 Len...
1.009062	54025	[TCP Retransmission] 54025 → 1883 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=193200 TSecr=0 WS=128	1883	TCP: [TCP Retransmission] 54025 → 1883 [SYN] ...
3.088156	54025	[TCP Retransmission] 54025 → 1883 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=193408 TSecr=0 WS=128	1883	TCP: [TCP Retransmission] 54025 → 1883 [SYN] ...
5.196406	54025	1883 → 54025 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1400 SACK_PERM=1 TSval=1208301697 TSecr=193099 WS=128	1883	TCP: 1883 → 54025 [SYN, ACK] Seq=0 Ack=1 W...
5.215375	54025	54025 → 1883 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=193620 TSecr=1208301697	1883	TCP: 54025 → 1883 [ACK] Seq=1 Ack=1 Win=29...
5.215375	54025	Connect Command	1883	MQTT: Connect Command
5.276406	54025	[TCP Out-Of-Order] 1883 → 54025 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1400 SACK_PERM=1 TSval=1208301776 TSecr=193099 WS=128	1883	TCP: [TCP Out-Of-Order] 1883 → 54025 [SYN, A...
5.288156	54025	[TCP Dup ACK 5#1] 54025 → 1883 [ACK] Seq=15 Ack=1 Win=29312 Len=0 TSval=193628 TSecr=1208301697	1883	TCP: [TCP Dup ACK 5#1] 54025 → 1883 [ACK] ...
5.468094	54025	Connect Ack	1883	MQTT: Connect Ack
5.498156	54025	54025 → 1883 [ACK] Seq=15 Ack=5 Win=29312 Len=0 TSval=193647 TSecr=1208301965	1883	TCP: 54025 → 1883 [ACK] Seq=15 Ack=5 Win=2...
5.498156	54025	Publish Message [paho/tts/callback]	1883	MQTT: Publish Message [paho/tts/callback]
5.498156	54025	Disconnect Req	1883	MQTT: Disconnect Req
5.531375	54025	1883 → 54025 [ACK] Seq=1 Ack=15 Win=29056 Len=0 TSval=1208301965 TSecr=193620	1883	TCP: 1883 → 54025 [ACK] Seq=1 Ack=15 Win=2...
5.905000	54025	1883 → 54025 [ACK] Seq=5 Ack=119 Win=29056 Len=0 TSval=1208302424 TSecr=193647	1883	TCP: 1883 → 54025 [ACK] Seq=5 Ack=119 Win=...
5.931375	54025	1883 → 54025 [FIN, ACK] Seq=5 Ack=119 Win=29056 Len=0 TSval=1208302424 TSecr=193647	1883	TCP: 1883 → 54025 [FIN, ACK] Seq=5 Ack=119 ...
5.943156	54025	54025 → 1883 [ACK] Seq=119 Ack=6 Win=29312 Len=0 TSval=193693 TSecr=1208302424	1883	TCP: 54025 → 1883 [ACK] Seq=119 Ack=6 Win=...

- Retransmissions can impact any packet of the TCP process causing significant delays in transmission and air interface overhead

SCEF Improves Data Delivery

Non-IP Data Delivery

Overhead Reduction

- TCP/IP data overhead 20 to 60 bytes for IPv4
- For small data transmission the payload is reduced significantly by removing the IP overhead

Security

- No device IP address
- Devices can only be reached by authorized application servers

SCEF Efficiency Improvements

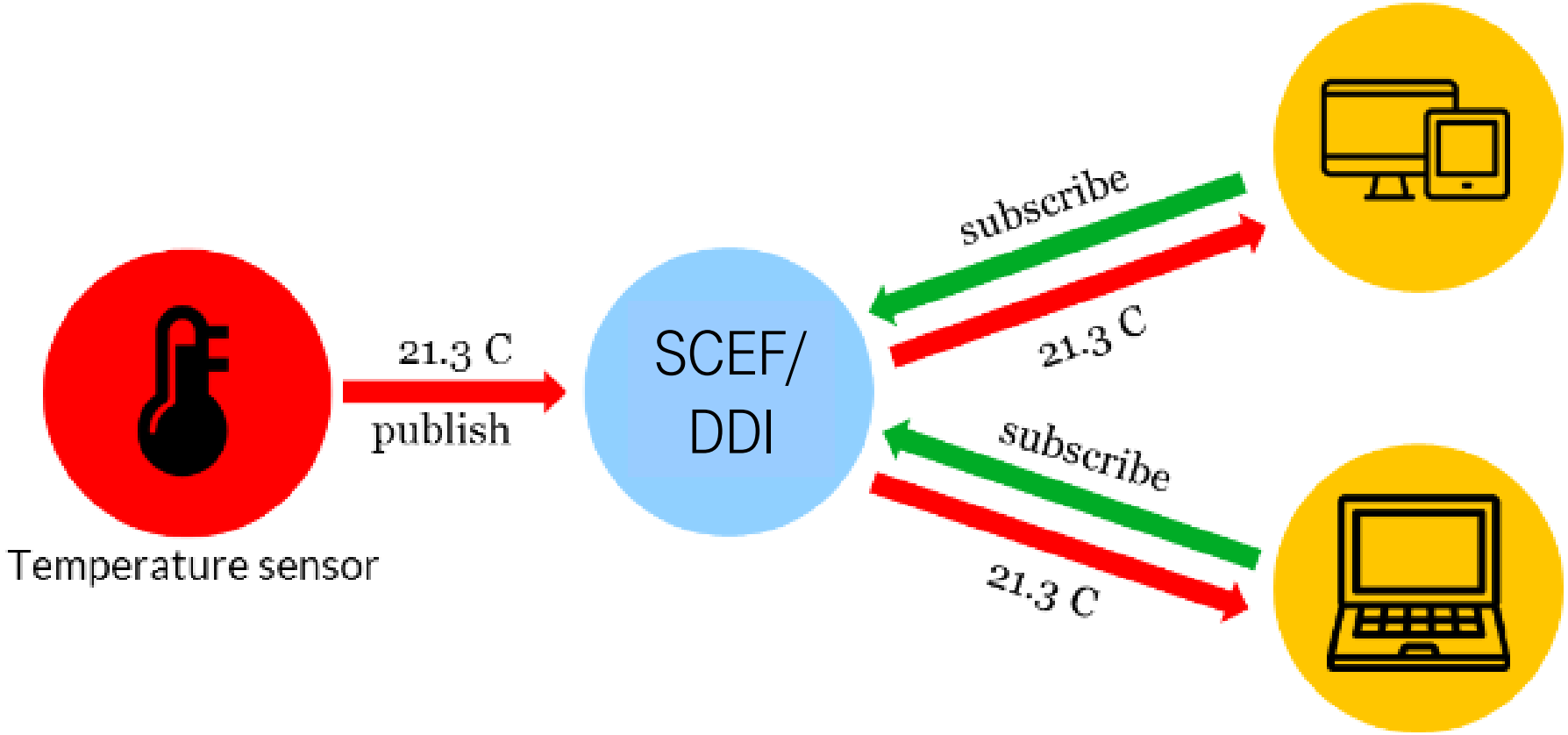
Multiple Recipients

- The Application Server needs to send only one message
- Message forwarded to all or some of the devices the SCEF has onboarded

Downlink Data Efficiency

- Network Data Buffering
- Data sent only when device is awake

SCEF provides similar functionality to MQTT broker



Schematic data flow from sensor (machine) to devise (machine)

Recommendations

- **CoAP is a preferred method for IP data transmission**
  - Requirements for retransmission timers should be created and shared with app developers
- **Non-IP data delivery can offer superior transmission efficiency for both network capacity and product partners**
  - Significant product development required on both application and module sides
- **Developers need to be incentivized toward a protocol to better protect network assets**