ASPIS: A Holistic and Practical Mechanism for Efficient MTC Support over Mobile Networks

ASPIS: A Holistic and Practical Mechanism for Efficient MTC Support over Mobile Networks

Galini Tsoukaneri
School of Informatics
The University of Edinburgh
Email: G.Tsoukaneri@ed.ac.uk

Xenofon Foukas
School of Informatics
The University of Edinburgh
Email: x.foukas@ed.ac.uk

Mahesh K. Marina
School of Informatics
The University of Edinburgh
Email: mahesh@ed.ac.uk

***Abstract*—Machine Type Communications (MTC) collectively refers to the exchange of data among devices that operate without human intervention. A significant number of such devices are currently served by cellular networks, and that number is expected to grow in the near future. However, cellular networks, including current fourth generation LTE networks, face several challenges when it comes to handling MTC traffic as they were primarily designed for Human Type Communications (HTC) which have very different traffic patterns. In this paper we focus on periodic MTC devices, such as sensors and meters, which cause significant signaling load and increased collisions over standard LTE networks. We propose ASPIS, a holistic mechanism designed to overcome these problems. ASPIS reduces the signaling load by partly preserving a device's connection to the network in conjunction with a new Random Access process and efficient support for short message transmissions. In addition, it uses a proactive preamble split scheme to alleviate collisions. ASPIS is easy to implement without requiring hardware changes while at the same time maintains security and can be incrementally deployed alongside legacy devices/infrastructure. We showcase the practicality of ASPIS by implementing it on the OpenAirInterface platform. Further, we demonstrate its effectiveness through extensive evaluations via a combination of small-scale experimental evaluation and large-scale, realistic simulations.**

I. Introduction

Machine type communication (MTC) refers to the automated generation, processing and exchange of data by devices that operate without human supervision, such as meters and sensors. MTC is a key enabler for the Internet of Things (IoT), with a large number of MTC devices currently being served over cellular networks, while that number is expected to grow significantly in the near future [1], [2]. Forecasts of MTC growth in the coming years indicate that LTE will be the dominant access technology by 2020 [3], even though it does not natively support MTC devices.

However, supporting this growing number of MTC devices is challenging for current cellular networks, as they are designed for Human-Type Communications (HTC). HTC traffic is quite different from MTC, as it is mainly characterized by infrequent connections, large packets (e.g. downloading files) and is mostly downlink (i.e. from the network to the device). MTC traffic on the other hand, is mostly uplink, with short packets (e.g. temperature readings) and frequent connections [4]. More importantly, these connections are often not random, but periodic [5].

The frequent connections of MTC devices is the major reason that pushes LTE networks to their limits [5]. Typically, MTC devices pass through the Random Access (RA) and Attach procedures to get into a connected state, send a short message and are then switched back to the idle state by the eNodeB (base station in LTE), tearing down any established communication bearers. This procedure needs to be repeated each time a device wants to send data, introducing a large signaling load in both the Radio Access Network (RAN) and the core network (EPC) of LTE networks. This signaling overhead is often disproportional to the size of the actual message being transmitted [4], [5]. To make matters worse, these connections may be synchronized between similar devices (e.g. measurements at pre-specified times), which can cause large number of collisions on the Random Access Channel (RACH) during the RA procedure. These inefficiencies are not only problematic in theory, but have been shown to cause problems in existing LTE networks [3], [4]. Given the expected growth of MTC devices in the future, 3GPP has recognized this as a potential threat [6]. An alternative approach would be to allow devices to have perpetual connections and avoid setting up/tearing down their bearers on each state switch. However, this is highly inefficient as it wastes precious resources in the RAN, and requires devices to remain active at all times, negatively affecting their battery life. In this paper, we focus on efficiently supporting MTC traffic, especially MTC devices that require frequent and periodic connections (e.g., meters, sensors), over LTE networks.

Existing approaches for MTC support over LTE networks focus on three distinct issues: (i) reducing the number of collisions in the RACH; (ii) reducing the signaling load in the EPC; and (iii) reducing the signaling load in the RAN. Currently, to avoid collisions in the RACH, 3GPP [6] and other works [7]–[11] propose the use of Access Class Barring and Enhanced Access Barring, which reduce the number of collisions at the expense of increased access delay. A few works [12]–[15] reduce the signaling load in the RAN, but forego security. Some works [16]–[20] reduce the signaling load in the EPC, but only for very small messages.

In this paper we present a holistic mechanism, called ASPIS, that simultaneously addresses all three aforementioned issues (signaling load in the EPC, the RAN and the number of collisions in the RACH) via a suite of underlying techniques.

Firstly, it incorporates a new Radio Resource Control (RRC) state that allows MTC devices to partly retain their communication bearers between transmissions, resulting in reduced signaling overhead in the EPC. Secondly, it uses a modified RA process for MTC devices, with fewer messages that reduces the signaling load in the RAN and total access delay. ASPIS also efficiently supports short packet transmissions ($<$80 bytes) for which RAN bearers may not be needed. Finally, it uses a proactive, dynamic preamble splitting scheme that finds an optimal split by exploiting the periodicity of MTC devices to predict future congestion before it happens, thereby lowering RACH collisions; note that preamble splitting is aligned with the direction being taken by 3GPP [19] but existing preamble splitting schemes are either static or reactive.

ASPIS requires no hardware changes and can be implemented as a software update. To demonstrate the ease of its implementation, we develop a prototype implementation of ASPIS (sec. V) over the widely used OpenAirInterface (OAI) [21] platform. Another noteworthy feature of ASPIS is that it enables these capabilities without compromising the security, as all messages sent are being encrypted. Furthermore, ASPIS can be incrementally deployed alongside legacy MTC/HTC devices and eNodeBs. We evaluate ASPIS via a combination of small-scale evaluations with the prototype implementation (sec. V) and large-scale evaluations (sec. VI) with thousands of devices, using a custom simulator based on realistic traffic patterns [4]. Our results show that ASPIS significantly outperforms standard LTE and recent proposals in terms of signaling load and collisions.

II. Background

On the air interface, the communication between the device and the base station (eNB) is done via signaling messages of the RRC protocol. To comprehend the signaling load introduced when a device connects to an LTE network, we present an overview of the standardized connection procedure for LTE networks. We first outline the two existing RRC states (sec. II-1) and then describe the procedures to transition from one state to the other (sec. II-2).

1) Radio Resource Control States: In LTE networks, a device can be in one of two RRC states: *RRC_Connected* or *RRC_Idle* (fig. 1). The state of the device indicates its connection status. In the connected state, a device has an allocated serving eNB, Mobility Management Entity (MME) and a Serving Gateway (S-GW). Signaling Radio Bearers (SRBs), Data Radio Bearers (DRBs), as well as S1 and S5/S8 bearers[1], are all active and any communication from or to the network is possible. In the idle state, a device is only able to send connection requests and receive paging messages. Although it is still registered with an MME and a S-GW (and retain their S5/S8 bearers), but does not have a serving eNB.

2) State Transition Procedures: In LTE, when an idle device has data to send, it needs to switch to the connected state

[1]S1 and S5/S8 bearers provide connectivity between the eNB and different modules of the EPC.

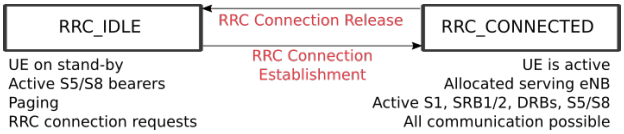

Fig. 1. LTE radio resource control states and their features.

using the RRC Connection Establishment process. Likewise, devices in the connected state that are inactive for a period of time (inactivity timer) are switched back to the idle state by the eNB using the RRC Connection Release process. We now briefly describe these two processes.

*a) **RRC Connection Establishment**:* The RRC Connection Establishment process (fig. 2) is used by a device in the idle state to request access to the network. It involves the *Random Access (RA)* process followed by the *Attach* process. At the end of the *Attach* process, the device is connected and able to exchange data.

The RA process begins with the transmission of a preamble value in a RACH slot (*msg1*). There are two different RA processes: the contention-free and the contention-based. The former is used when a device has dedicated preambles, whereas the latter is used when devices need to compete for channel access. In this work we focus on the contention-based RA process, as dedicating a preamble to each MTC device would be infeasible. Note that collisions can happen if multiple devices transmit the same preamble in the same slot. Upon receiving the preamble, the eNB replies with a *Random Access Response (RAR)* message (*msg2*) for each successfully decoded preamble. When a device receives a RAR associated with its transmitted preamble, it replies with a *RRC Connection Request* message (*msg3*). Collided devices on *msg1* will receive the same RAR, and thus also collide when transmitting *msg3*. To resolve the collision, the eNB selects one of the collided devices and sends it a *Contention Resolution (msg4)* message that echoes its *msg3*. Collided devices compare the echoed message with the one they transmitted. If it does not match, they terminate the process and retry again later; otherwise they send an acknowledgement to the eNB. The eNB then replies with a *RRC Connection Setup* message to configure the PHY and MAC protocols of the device. Finally, the device replies with an *RRC Connection Setup Complete* message to complete the RA process.

After the RA process, the device follows the *Attach* process, which involves a large number of signaling messages between the different entities of the EPC and the eNB [22]. The Attach process includes the *Authentication & Key Agreement (AKA)* process, the EPC bearer creation and the PDN connectivity request. Additionally, it allocates an MME and a S-GW to the device. The allocation of the MME and the S-GW as well as the AKA process are done when a device is switched on, and only need to be repeated if the device moves to an area administered by another MME. On the other hand, the EPC bearer needs to be set up every time the device switches from the idle to the connected state.

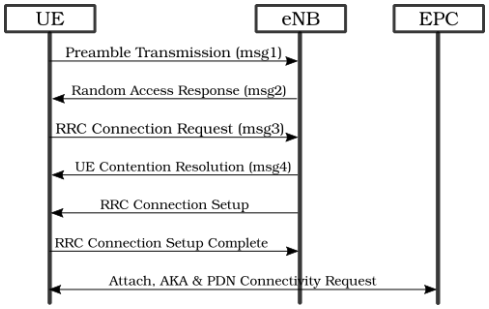

Fig. 2. RRC connection establishment process illustrated.

*b) **RRC Connection Release**:* After a period of inactivity, the eNB moves a device to the idle state using the RRC Connection Release procedure to free radio resources allocated to it. Specifically, this procedure is triggered after the expiration of an inactivity timer associated with the device, which is specified by the network operator. Commonly, the inactivity timer is set to 10 seconds.

The procedure begins with the eNB sending a *Context Release Request* message to the MME which forwards it to the S-GW. Upon receiving this message, both the MME and the S-GW tear down the S1 bearer of the device, but keep the S5/S8 bearers intact. The MME then replies back to the eNB and tells it to release the device's connection. The eNB sends an *RRC Connection Release* message to the device and tears down its SRBs, the DRBs and the S1 bearer. This puts the device to the idle state, and it will now need to repeat the RRC Connection Establishment process to transmit new data.

III. Related Work

There exist several works, that aim to handle MTC traffic more efficiently over LTE networks, by focusing on one of the underlying issues (signaling load reduction in EPC/RAN and reduction of collisions in the RACH). In contrast, with ASPIS, we address all of them simultaneously.

(a) Reduction of signaling load in the EPC: Several works (e.g., [19], [20], [23]) attempt to reduce the signaling load in the EPC by using existing signaling messages to transmit data. Although these schemes avoid setting up bearers in the EPC, they are only suitable for short packet transmissions and revert back to the standard process (sec. II) for larger messages. Additionally, they use the current RA process, that is not well suited for periodic transmissions.

Unlike these works, ASPIS makes the devices appear as perpetually connected via an intermediate RRC state that is transparent to the EPC while at the same time mimicking the idle state from the device/eNB perspective. Thus, signaling load in the EPC is avoided when device transitions between different states. A recent work [24], takes a similar perspective, but is inherently insecure. Specifically, to transition from the intermediate to the connected state, the device needs to send its *Connection Context ID* using the *RRC Connection Resume Request* message. At that point the eNB does not know which device it is communicating with, so it cannot activate its security context and the information needs to be sent in plain text, rendering the approach prone to sniffing attacks. Moreover, [24] is not compliant with LTE as the *Context ID* requires more than the 40 bits available at the LTE *RRC Connection Request* message. Finally, it is purely a design, lacking discussion for many implementation details.

(b) Reduction of signaling load in the RAN: To reduce the signaling load in the RAN, some works (e.g., [12]–[15]) propose using messages of the RA process to transmit short messages. However, [12], [13] are inherently insecure as no security context would have been established at the time of transmission of those messages. Furthermore, [14], [15] are only suitable for infrequent and non-periodic transmissions, since they would lead to an excessive number of collisions otherwise. Similar to these works, ASPIS supports the transmission of small packets without the need for an RA process, but can do so securely (sec. IV-B3). At the same time, ASPIS employs a modified RA process using fewer messages, reducing the signaling load in the RAN for larger packets.

(c) Collision reduction in the RACH: Access Class Barring (ACB) and Enhanced Access Barring (EAB) schemes [7]–[10], [19], [25] decrease the collisions in the RACH at the expense of higher access delay, by placing devices in different classes with different connection priorities, controlled by different access probabilities (APs). Network access is then controlled by adapting the APs. These schemes are complimentary to our work and can in fact be implemented on top of ASPIS.

Other works (e.g., [11], [19], [26], [27]) split the RA resources between the HTC and MTC devices to minimize the impact on HTC traffic. However, these splits are typically static irrespective of the changing network conditions and can limit the access opportunities for MTC device by misallocating the available resources, thus reducing the performance of the system under load. A few existing dynamic schemes [28], [29] *reactively* allocate additional RA preambles to the MTC devices based on the observed congestion. In contrast, ASPIS features a *proactive* preamble splitting technique which can predict congestion before it occurs and optimally allocate the preambles between the two groups of devices.

IV. ASPIS

In this section we present ASPIS, our proposed mechanism that aims to overcome the problems the large numbers of MTC devices cause to LTE networks, i.e., increased signaling load in the EPC, the RAN and increased number of collisions in the RACH. The core of ASPIS is the intermediate RRC state (sec. IV-A) that MTC devices transition to, instead of idle (sec. IV-B1), after a period of inactivity. The intermediate state preserves part of the device's previously established connection and reuses it each time that device wishes to transmit new data, thus reducing the signaling overhead in the EPC, as well as the access delay.

Furthermore, we introduce a new RA procedure for the intermediate state, to reduce the signaling load in the RAN. The new RA process (sec. IV-B3) requires fewer messages

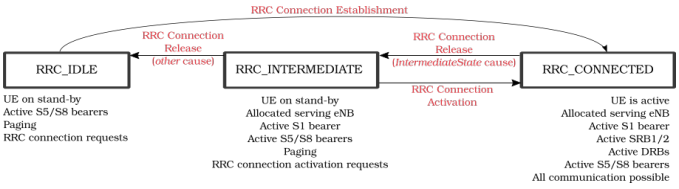

Fig. 3. ***Proposed RRC state diagram for MTC devices:*** *The figure depicts the new RRC state diagram for the MTC devices, with the existing states and their characteristics.*

than the standard RA process and also provisions for short packet transmissions (<80 bytes) for which SRBs and DRBs are not needed.

Finally, following 3GGP [19], we split the preambles between MTC and HTC devices. However, unlike a static split (as proposed in [19]) we use a novel proactive dynamic preamble split scheme, that exploits the periodic nature of MTC devices to predict future increases in the network access requests and minimize the number of collisions before they happen. ASPIS can be supported by newer devices without affecting the operation of the existing systems. An eNB is informed about a device capable of supporting ASPIS via a new UE-Category (13) (sec. IV-C).

We will now describe in detail the intermediate state, the procedures to switch from it to the existing states and the proactive preamble split.

A. RRC Intermediate State

We propose a new intermediate state for MTC devices in the RRC layer, which combines some characteristics of the two existing states and allows the MTC devices to be in a semi-connected mode. Similar to the idle state, devices in this mode do not have dedicated RAN resources and cannot transmit scheduling requests or periodic measurement reports. However, they can listen for paging messages, receive connection reconfiguration messages and MAC alignment control commands (to remain synchronized). Devices in the intermediate state continue to be registered with an MME and a S-GW. The intermediate state is only visible to the eNB and the EPC is unaware of it. All previously established EPC bearers still exist and can be used immediately after the device switches back to the connected state, resulting in decreased signaling load in the EPC. In this work, we assume that MTC devices have only one EPC bearer, which is a realistic assumption for many MTC deployment scenarios.

In order to facilitate transitions of MTC devices between the different RRC states (fig. 3), we use two additional inactivity timers. The first one is used to move the device from the connected to the intermediate state and the second one switches the device from the intermediate to the idle state. Similarly to the inactivity timer in standard LTE, these timers are set by the network operator and can be chosen to strike a balance between required resources and signaling load (sec. VI-A).

B. RRC State Transition Procedures

In this section we describe the procedures that the MTC devices need to follow to transition between the three RRC states: connected, intermediate and idle (fig. 3). We also discuss how our proposed procedures manage to reduce the signaling load both in the EPC and the RAN. The complete procedure is depicted in fig. 4. As for the standard LTE, the communication in the air interface is done using RRC signaling messages.

1) Transition from CONNECTED to INTERMEDIATE: After a period of inactivity in the connected state (controlled by our first inactivity timer), the eNB releases the connection, frees its allocated resources, and moves the device to the intermediate state. To do so, the eNB sends a *RRC Connection Release* message with the release cause set to the new value of "IntermediateState". This instructs the device to tear down the DRB and all its SRB bearers and sets it to the intermediate state. The use of any other release cause indicates that the eNB does not support ASPIS, and thus the default RA procedure needs to be followed the next time the device has data to transmit. In this way, ASPIS capable devices can be supported by legacy eNBs. The eNB preserves all the bearers of the device in the core network (S1, S5/S8). Note that the MME and S-GW are never informed about the new state of the device, thus avoiding unnecessary load in the core network.

When switched to the intermediate state, the device keeps its personal keys that were established during the initial AKA procedure, which will be reused once it switches back to the connected state. Therefore, the AKA process happens only once and does not incur any additional overhead for the subsequent transitions to the connected state.

2) Transition from INTERMEDIATE to IDLE: When a device remains inactive in the intermediate state for a period of time specified by our second inactivity timer, the eNB releases its EPC connection, in order to reduce the number of unused resources and avoid potential DoS attacks. To accomplish that, the eNB pages the device and then initiates the standard LTE *S1 Release Procedure* (sec. II).

3) Transition from INTERMEDIATE to CONNECTED: If an MTC device in the intermediate state wishes to transmit new data, it needs to transition to the connected state following our new RA process. In the current procedure there are two different groups of preamble values (groupA, groupB) for contention-based RA [30]. Here, we introduce a new preamble group (groupC) that is only used by MTC devices in the intermediate state. Being in the intermediate state indicates that the device has previously registered and connected to the network, and has an established EPC bearer. The allocation of preambles in groupC is done by the eNB (sec. IV-C).

Initially, the MTC device chooses a random preamble value from groupC and transmits it in the RACH. Similarly to the standard LTE procedure (sec. II), the eNB replies with a RAR message, and the device then replies with a *RRC Connection Activation* message. This is a newly introduced message that contains the KSI (Key Set Identifier) of the device-specific security context (that was established during

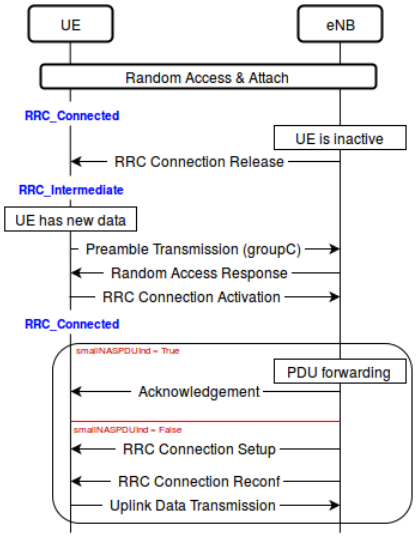

Fig. 4. ***ASPIS*** *The figure depicts the complete procedure of ASPIS with all the different cases it handles.*

the initial AKA procedure), the S-TMSI of the device, a NAS PDU container, an indicator of whether a small NAS PDU message is encapsulated in the NAS PDU container, the APN of the PDU's destination and the device's EPC bearer-ID. The NAS PDU indicator, NAS PDU, EPC bearer-ID and APN are encrypted with the device's specific security keys. The eNB knows that the connection activation message was transmitted by a device in the intermediate state based on the preamble value used. Therefore it uses the 1-1 map between a device and its KSI along with the S-TMSI to uniquely identify the requesting device and authenticate the received message. In case of authentication failure, the eNB discards the packet.

The NAS PDU indicator informs the eNB whether a short packet ($<$80 bytes) is encapsulated in the NAS PDU container. If the NAS PDU indicator is false (i.e., the device wants to transmit a message larger than 80 bytes), the eNB needs to set up RAN bearers. This is done with a *RRC Connection Setup* message followed by *RRC Connection Reconfiguration* to set up the the DRB and SRBs for the device. The device can then use its existing EPC bearer and start sending data. If the indicator is true, the eNB does not need to set up RAN bearers for the device. Instead, the eNB first needs to communicate with the MME to identify the device's allocated S-GW. It then extracts the PDU, encapsulates it in a GTP packet and forwards it to the S-GW using the device's existing bearer.

Having to set up RAN bearers for small packets introduces a signaling load disproportional to the actual size of the message. Therefore, our provision for short data transmissions further decreases the signaling load in the RAN. The inspiration for this enhancement is based on the Small Data Transmission (SDT) [31] procedure of 3GPP. However, unlike ASPIS, the SDT procedure requires the establishment of an RRC connection before any data transmission occurs.

Since all MTC devices in the intermediate state share the same preambles (groupC), collisions may occur, and the eNB does not send an acknowledgement to any of the colliding devices. When a device does not receive an acknowledgement within a certain number of frames it backs off and tries again at a later time. If the device fails to re-activate its connection for a specified number of times, it switches back to the idle state and attempts the conventional RA process. However, in this case the device needs to inform the eNB about its active EPC bearer when it manages to connect again, to prevent the eNB and EPC from creating a new bearer and also avoid the existence of stalling EPC bearers. Since the MTC device is now in the idle state, it uses the standard RA process and informs the eNB about its existing bearer using the *RRC Connection Setup Complete* message.

C. Proactive dynamic preamble split

So far we have discussed our new intermediate state and RA process, which reduce the signaling load in the EPC and RAN. Here, we present our proactive preamble split mechanism that exploits the periodicity of MTC devices to reduce the number of collisions in the RACH. Intuitively, our proactive preamble split mechanism predicts the number of MTC devices that are likely to transmit in each frame and proactively adapts the number of preambles allocated between MTC in the intermediate state (groupC) and any devices in the idle state (groupA and groupB), in order to minimize the *overall* number of collisions.

The idea behind this enhancement is that MTC devices send data on predefined intervals specified by their periodicity. Based on the time of their last transmission and the periodicity of each device, the eNB can predict how many devices are expected to transmit in any given frame. This allows us to optimally split the preambles in advance so that we do not needlessly over-allocate preambles to any group. The dynamic preamble split is broadcasted in the SIB2 and indicates the new number of preambles in the different groups.

In order to learn the periodicity of a device, the eNB can use the Capabilities Enquiry message. To indicate that the device supports ASPIS, we introduce a new UE-Category (13). The device indicates its periodicity in milliseconds in the Indicator-31 of the *featureGroupIndicators* field in the capabilities message.

We pose our proactive preamble allocation mechanism as an optimization problem. Let R be the total number of preambles available for contention-based access, r_M be the number of preambles values allocated for MTC devices in the intermediate state, and n_t be the number of such devices expected to transmit in frame t. The expected number of colliding MTC devices (in the intermediate state) is then:

$$E(n_t, r_M) = n_t(1 - (1 - \frac{1}{r_M})^{n_t - 1}) \quad (1)$$

Similarly, let n'_t be the number of idle devices that transmit in frame t. Since we cannot predict when these devices will transmit we can approximate n'_t as the running average over a number of previous frames. The expected number of collisions of these devices can be expressed as:

$$E(n'_t, R - r_M) = n'_t(1 - (1 - \frac{1}{R - r_M})^{n'_t - 1}) \quad (2)$$

As the total number of contention-based preambles is fixed, increasing the preambles for MTC devices in the intermediate state decreases the available preambles for other devices. Ideally, we would like to minimize the total number of collisions for all devices. Additionally, we would like to guarantee a minimum number r_{min} of preambles given to HTC devices, as their expected transmissions are only an approximation based on past frames. In other words, we want to minimize:

$$\arg\min_{r}(E(n_t, r_M) + E(n'_t, R - r_M)), \text{ s.t. } r_M \leq R - r_{min} \quad (3)$$

Note that $E(n_t, r_M)$ is monotonically increasing while $E(n'_t, R - r_M)$ is monotonically decreasing as r_M increases, so eq. 3 is convex. As such, minimizing eq. 3 can be done efficiently with a simple modification of the binary search algorithm. Furthermore, the total number of preambles R is typically very small so the overall computation time is negligible for the eNB.

Due to the use of the proactive preamble allocation, ASPIS is able to scale gracefully. If there are no devices connected, the system will not waste resources on them, but it will naturally fall back to the current behavior of allocating all preambles to devices in the idle state.

V. Prototype Implementation and Experimental Results

An attractive aspect of ASPIS is the ease with which it can be realized in the context of current and emerging cellular network standards. It can be implemented via software updates to eNB and devices, and does not require hardware changes. More specifically, implementing ASPIS involves changes to the RRC and PHY layers at the eNB and the device. In the RRC layer, ASPIS requires the introduction of the new intermediate state, two additional inactivity timers and the *RRC Connection Activation* message. Minimal changes at the PHY layer are required for the proactive preamble splitting technique so that the eNB can update the SIB2 about the allocation of the preambles to the different groups. A noteworthy aspect of ASPIS is that it can be incrementally deployed, i.e. legacy devices can connect to ASPIS enabled eNBs and ASPIS enabled MTC devices can still connect to legacy eNBs (sec. IV-B1).

We have developed a prototype implementation of ASPIS over the OpenAirInterface (OAI) platform [21], a well-known open source software implementation of the LTE RAN and EPC components. Doing so was straightforward and essentially involved making the changes outlined above. We intend to make this implementation publicly available in the near future.

We have used this prototype implementation of ASPIS for experimental evaluation in small-scale settings (10 ASPIS enabled MTC devices), considering various different metrics. As signaling load results are similar to those obtained via simulations (sec. VI), we only present here a sample experimental result based on the implementation focusing on access delay. 3GPP [32] defines access delay as the time elapsed from the transmission of the first preamble until the receipt of the *msg4* (fig. 2). As we alter the existing RA process, we instead measure the access delay as the time passed from the first preamble transmission until the start of actual data transmission (fig. 5). We used 10 ASPIS-enabled MTC devices associated with an eNB and implemented over OAI.

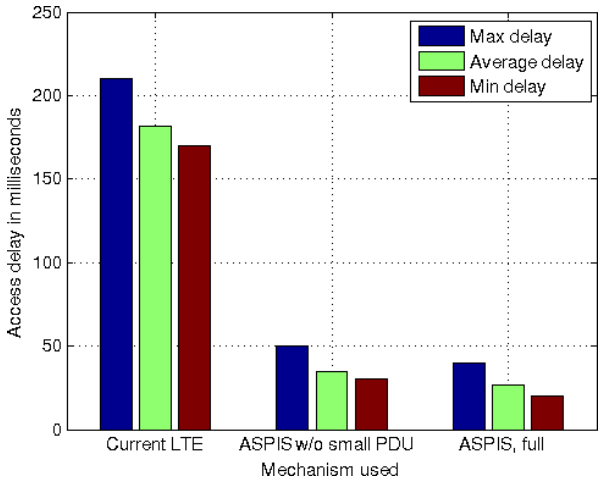

Fig. 5. ***Access delay:*** *comparison between variants of ASPIS and standard LTE in a setting with 10 MTC devices implemented over OAI.*

Fig. 5 shows the average access delay of ASPIS in comparison with standard LTE specifications. ASPIS enabled MTC devices show clearly faster network access with and without optimising for small PDUs. This is expected as the device requires fewer messages to switch to the connected state from the intermediate state rather than the idle.

VI. Large Scale Evaluations via Simulations

A. Setup

While the ASPIS implementation over OAI demonstrates its practicality, OAI cannot currently support large number of devices. To assess ASPIS with respect to the key metrics of interest (signaling load in the EPC, in the RAN and the number of collisions in the RACH during the preamble transmission), in scenarios with thousands of MTC devices, we developed a custom simulator in MATLAB that models both standard LTE procedures as well as ASPIS. We validated this simulator against the ASPIS OAI implementation in small-scale settings in terms of the signaling load for the same number of devices and obtained similar results.

We use realistic traffic models based on [4], which is the largest publicly available study on MTC traffic patterns. We simulated at least 18000 frames, with 1000 HTC devices per cell and a varying number of MTC devices, ranging from 500 up to 4000. All simulations include different types of MTC devices that transmitted small PDUs (up to 80 bytes) as well as larger packets. In our simulation 50% of the MTC devices transmit small PDUs, which accounts for the 41% of

the total MTC traffic. For ASPIS, the inactivity timer 1 for the intermediate state was set to 2.8 seconds, while the inactivity timer 2 was set to 90 seconds. We set the value of inactivity timer 1 to 2.8 seconds based on the average session interarrival and session length of MTC devices of [4]. Each data point in the plots is an average obtained from 10 repetitions for that data point.

For the RA process, we used 56 preambles in total. For the standard LTE [19], we allocate 12 preambles for MTC devices and 44 preambles for HTC devices. The split between the HTC and the MTC devices for ASPIS is done using the proactive split mechanism (sec. IV-C). In addition, the maximum number of collisions a device can experience was set to 3, as in standard LTE. We compare against the current specifications [33] which we use as our baseline, the SDT [31] proposal of 3GGP, and the recent works of [14], [20]. The SDT proposal [31] avoids setting up EPC bearers for small packets but still requires the establishment of an RRC connection in the RAN. Maldonado et al. [14] present an alternative RA process for small packet transmission; however, for larger packets the device still has to go through the complete RA and Attach procedure. In [20] the authors use the authentication messages of the AKA procedure to securely transmit short messages. This approach uses the existing RA without any modifications, so it requires the Attach process when larger packets need to be transmitted. We do not compare against the works like [12], [13] as these use *msg1* or *msg3* of the RA process to transmit data, which is inherently insecure (sec. III)).

B. Results

*1) **Load in the EPC**:* Fig. 6 shows the results for signaling load in the EPC for the different approaches. Signaling load is normalized by the number of frames in the simulations. Clearly, ASPIS significantly outperforms the current connection establishment procedure [33], the SDT proposal [31] as well as the recently proposed schemes of [14], [20]. This is expected, as the focus of most of these alternative approaches is only on the transmission of small packets that can be contained in existing RRC or NAS messages, and they still have to follow the existing process for setting up bearers in the core network for larger messages. ASPIS provisions for both small (up to 80 bytes) and larger messages, and alleviates the signaling load in both cases. Key contributor to savings with ASPIS is the use of the intermediate RRC state that gives the illusion to the EPC that the device is in the connected state even though it may be dormant between periodic transmissions.

*2) **Load in the RAN**:* Here, we evaluate the signaling load of ASPIS in the RAN against the current RA procedure [33] and the work of [14]. Recall that [31] and [20] use the RA procedure of the current specifications, so the load in the RAN is identical to that. As shown in fig. 7, ASPIS performs significantly better compared to standard LTE and the recently proposed mechanism [14] due to the fewer messages that it requires. This is because the alternative mechanisms either do not account for MTC traffic (current cellular networks) or only alleviate load for small PDU messages [14]. In contrast to these schemes, ASPIS decreases the signaling load for all packet sizes.

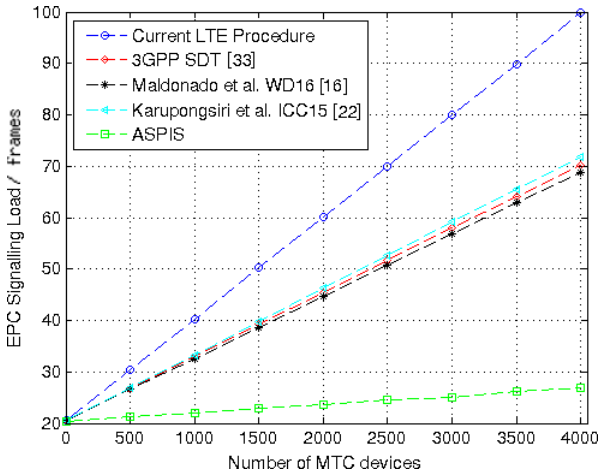

Fig. 6. ***Signaling load in the* EPC***: the figure depicts the EPC signaling load as a function of number of MTC devices with ASPIS and alternative approaches.*

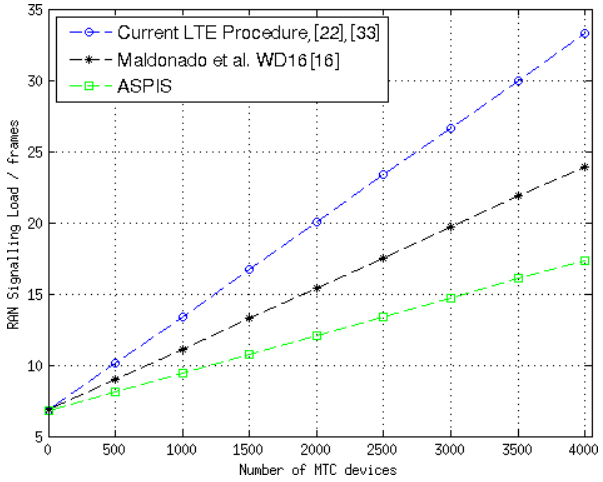

Fig. 7. ***Signaling load in the* RAN***: the figure depicts the RAN signaling load as a function of number of MTC devices with ASPIS and alternative approaches.*

a) Small PDUs vs Larger PDUs: To better appreciate the benefit of the provision for the small PDUs, we compare the signaling load performance of ASPIS (fig. 8) with the provision for the small PDUs transmission (shown as full) and without; current RA procedure is included in the comparison for reference. One interesting thing to notice is that in the EPC, the signaling load introduced when small packets are transmitted is marginally greater than when ASPIS does not provision for small PDUs. This is because the transmission of small PDUs uses an extra signaling message in the EPC to forward the PDU (sec. IV-B3). However, both our approaches result in significant gains in terms of the signaling load compared to standard LTE.

*3) **Collisions**:* In this section we evaluate the collisions of ASPIS in the RACH compared to other splitting schemes

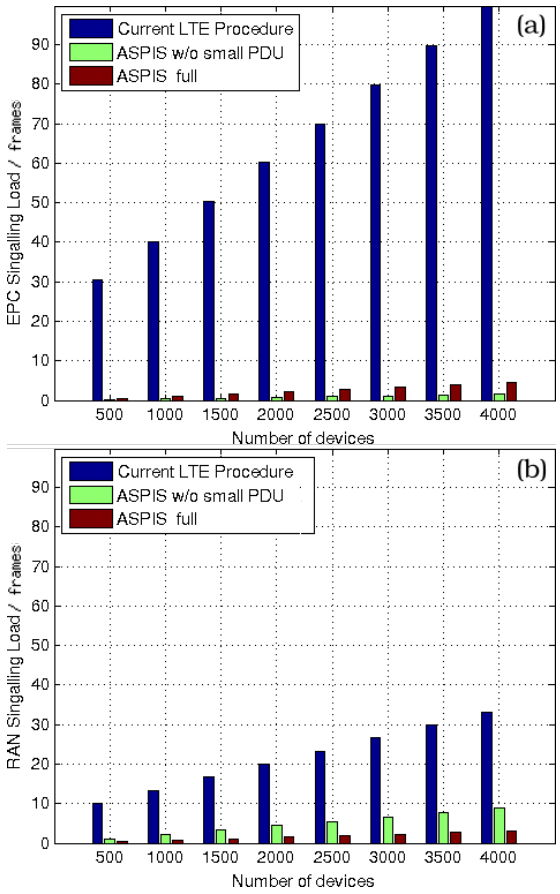

Fig. 8. ***Comparison of ASPIS with and without the use of small PDUs:*** *The figure depicts the normalized signaling load of ASPIS against the current RA procedures. Note that without the small PDUs, the MTC devices use ASPIS to set up RAN bearers for their transmission. (a) Signaling load in the EPC, (b) Signaling load in the RAN.*

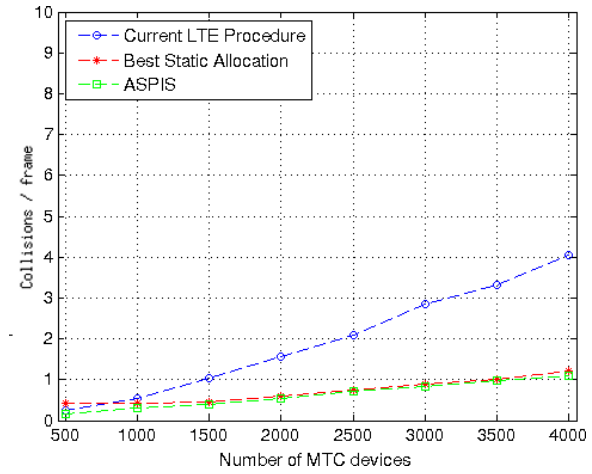

Fig. 9. ***Collisions*** *The figure depicts the collisions in the RACH normalized by the number of frames, with the current static scheme of cellular networks, the best performing static split, and our proactive preamble split scheme.*

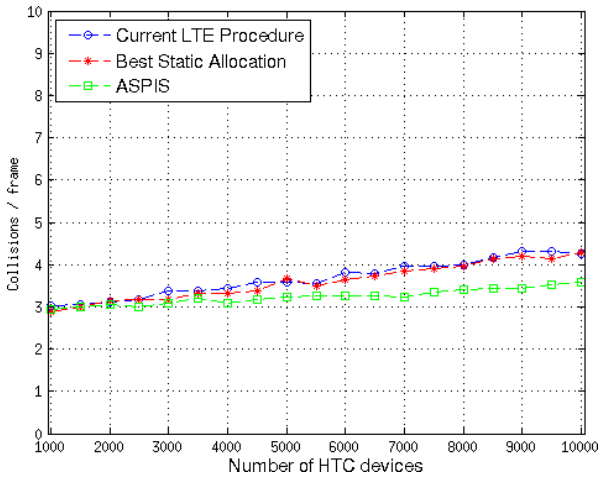

Fig. 10. ***Collisions:*** *The figure depicts the collisions in the RACH normalized by the number of frames, for varying number of HTC and a fixed number of MTC devices compared against the current cellular networks procedure and the best performing static split.*

(fig. 9), which is an important aspect of our method. Our proactive preamble split scheme exploits the periodicity of MTC devices to reduce the likelihood of collisions in the RACH. To evaluate our proactive scheme, we measure the collisions during the preamble transmission and compare it against standard LTE procedure with the static preamble split [19], which we use as our baseline. Furthermore, we consider every possible static preamble split to show that our proactive preamble split produces fewer collisions in all cases. For visual clarity, we only show the static split with the best performance, i.e. the static split that produced the least number of RACH collisions. Also note that all these splitting schemes use the intermediate state. We show that our proactive split performs comparably to the best possible static split, even though the latter is impossible to know in advance as future HTC traffic cannot be known a-priori.

We also evaluate the collisions in the RACH for a fixed number of MTC devices (4000) and a number of HTC devices varying from 1000 to 10000 (fig. 10). As before, we only show the standard LTE approach and the best performing static split for clarity. Notice how the differences between the different methods are smaller compared to fig. 9. As mentioned before, MTC devices have a significantly more negative effect on the number of collisions compared to the HTC devices, due to their frequent connections. As a result, increasing the nunber of HTC devices does not increase the number of collisons as much. However, the results show that ASPIS results in lower collisions in this scenario as well.

VII. Conclusions

In this paper we have focused on efficiently supporting large numbers of MTC devices that require periodic and repeated transmissions (e.g. sensors and meters) over LTE networks. Due to their traffic patterns, such devices introduce a considerable signaling load both in the EPC and the RAN, which is usually disproportional to their actual message size. Furthermore, such devices may often be synchronized resulting in increased collisions in the RACH. To address all these problems, we proposed ASPIS, a holistic mechanism that is easy to implement, and works with existing hardware.

ASPIS introduces an additional RRC intermediate state that partially preserves a device's connection which can be reused in future transmissions, alleviating the need to set up a new connection each time. It is noteworthy that our intermediate state idea bears similarity to the approach being considered for the upcoming 3GPP standards (Rel. 14) en route to 5G [34]. This is a positive sign as ASPIS blends this idea together with other techniques in a practical manner for widely deployed LTE networks that will be the dominant carrier of MTC traffic as we approach 2020. Besides, ASPIS features an improved RA process with fewer messages that also provisions for short packet transmissions. While we have not conducted energy consumption measurements, as these vary significantly between different devices, we note that reduced messaging would directly result in reduced energy consumption. Finally, ASPIS incorporates a proactive preamble split scheme that predicts future increases in the access requests and dynamically adapts the preamble split to alleviate collisions before they happen. We show the practicality of ASPIS through a prototype implementation over the OpenAirInterface platform. Our experimental and large-scale simulation results confirm that ASPIS outperforms the standard LTE as well as other recent proposals.

REFERENCES

[1] R. Behrends, L. K. Dillon, S. D. Fleming, and R. E. K. Stirewalt, "The zettabyte era—trends and analysis," Cisco, Tech. Rep. 1465272001628117, June 2016. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html

[2] Vodafone, "The M2M adoption barometer 2014," Vodafone, Tech. Rep., July 2014.

[3] Cisco, "White paper: Cisco visual networking index: Global mobile data traffic forecast update, 2016–2021," Cisco, Tech. Rep., March 2017. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html

[4] M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang, and J. Wang, "Large-scale measurement and characterization of cellular machine-to-machine traffic," *IEEE/ACM Transactions on Networking*, vol. 21, no. 6, Dec 2013.

[5] J. Jermyn, R. P. Jover, I. Murynets, M. Istomin, and S. Stolfo, "Scalability of machine to machine systems and the internet of things on LTE mobile networks," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2015 IEEE 16th International Symposium on a*, June 2015.

[6] 3GPP, "Access class barring and overload protection," 3rd Generation Partnership Project (3GPP), TR 23.898, March 2005.

[7] J. Moon and Y. Lim, "Adaptive access class barring for machine-type communications in LTE-A," in *2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*, July 2016.

[8] Z. Wang and V. W. S. Wong, "Optimal access class barring for stationary machine type communication devices with timing advance information," *IEEE Transactions on Wireless Communications*, vol. 14, no. 10, Oct 2015.

[9] S. Duan, V. Shah-Mansouri, Z. Wang, and V. Wong, "D-ACB: Adaptive congestion control algorithm for bursty M2M traffic in LTE networks," *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, 2016.

[10] M. Tavana, V. Shah-Mansouri, and V. W. S. Wong, "Congestion control for bursty M2M traffic in LTE networks," in *Communications (ICC), 2015 IEEE International Conference on*, June 2015.

[11] J.-P. Cheng, C. han Lee, and T.-M. Lin, "Prioritized random access with dynamic access barring for RAN overload in 3GPP LTE-A networks," in *GLOBECOM Workshops (GC Wkshps), 2011 IEEE*, Dec 2011.

[12] Y. Chen and W. Wang, "Machine-to-machine communication in LTE-A," in *Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd*, Sept 2010.

[13] D. T. Wiriaatmadja and K. W. Choi, "Hybrid random access and data transmission protocol for machine-to-machine communications in cellular networks," *IEEE Transactions on Wireless Communications*, vol. 14, no. 1, Jan 2015.

[14] P. Andres-Maldonado, P. Ameigeiras, J. Prados-Garzon, J. J. Ramos-Munoz, and J. M. Lopez-Soler, "Reduced M2M signaling communications in 3GPP LTE and future 5G cellular networks," in *2016 Wireless Days (WD)*, March 2016.

[15] S. Andreev, A. Larmo, M. Gerasimenko, V. Petrov, O. Galinina, T. Tirronen, J. Torsner, and Y. Koucheryavy, "Efficient small data access for machine-type communications in LTE," in *2013 IEEE International Conference on Communications (ICC)*, June 2013.

[16] G. Zhang, A. Li, K. Yang, L. Zhao, Y. Du, and D. Cheng, "Energy-efficient power and time-slot allocation for cellular-enabled machine type communications," *IEEE Communications Letters*, vol. 20, no. 2, Feb 2016.

[17] S. H. Wang, H. J. Su, H. Y. Hsieh, S. p. Yeh, and M. Ho, "Random access design for clustered wireless machine to machine networks," in *Communications and Networking (BlackSeaCom), 2013 First International Black Sea Conference on*, July 2013.

[18] G. Farhadi and A. Ito, "Group-based signaling and access control for cellular machine-to-machine communication," in *Vehicular Technology Conference (VTC Fall), 2013 IEEE 78th*, Sept 2013.

[19] 3GPP, "Study on RAN Improvements for Machine-Type Communcations," 3rd Generation Partnership Project (3GPP), TR 37.868, 2011.

[20] C. Karupongsiri, K. S. Munasinghe, and A. Jamalipour, "Smart meter packet transmission via the control signal of LTE networks," in *2015 IEEE International Conference on Communications (ICC)*, June 2015.

[21] "OpenAirInterface (OAI): Towards open cellular ecosystem." [Online]. Available: http://www.openairinterface.org/?page_id=864

[22] 3GPP, "General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access," 3rd Generation Partnership Project (3GPP), TS 23.401, Sep. 2013.

[23] S. T. Sheu, C. H. Chiu, S. Lu, and H. H. Lai, "Efficient data transmission scheme for MTC communications in LTE system," in *2011 11th International Conference on ITS Telecommunications*, Aug 2011.

[24] I. L. D. Silva, G. Mildh, M. Säily, and S. Hailu, "A novel state model for 5g radio access networks," in *2016 IEEE International Conference on Communications Workshops (ICC)*, May 2016.

[25] T. M. Lin, C. H. Lee, J. P. Cheng, and W. T. Chen, "PRADA: Prioritized random access with dynamic access barring for MTC in 3GPP LTE-A networks," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 5, Jun 2014.

[26] Y. Liu, F. Pingzhi, and H. Li, "A preamble allocation method for M2M traffics in 3GPP LTE-A networks," in *2015 Seventh International Workshop on Signal Design and its Applications in Communications (IWSDA)*, Sept 2015.

[27] D. Kim, W. Kim, and S. An, "Adaptive random access preamble split in LTE," in *2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*, July 2013.

[28] S. Duan, V. Shah-Mansouri, and V. W. S. Wong, "Dynamic access class barring for M2M communications in LTE networks," in *2013 IEEE Global Communications Conference (GLOBECOM)*, Dec 2013.

[29] W. Li, Q. Du, L. Liu, P. Ren, Y. Wang, and L. Sun, "Dynamic allocation of RACH resource for clustered M2M communications in LTE networks," in *2015 International Conference on Identification, Information, and Knowledge in the Internet of Things (IIKI)*, Oct 2015.

[30] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access (E-UTRAN); Overall description; Stage 2," 3rd Generation Partnership Project (3GPP), TS 36.300, Sep. 2008. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/36300.htm

[31] ——, "Study on machine type communicaions and other mobile data applications communications enhancements," 3rd Generation Partnership Project (3GPP), TR Rel 12, 2013.

[32] ——, "Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification," 3rd Generation Partnership Project (3GPP), TS 36.331, Sep. 2013.

[33] ——, "Medium access control (MAC) protocol specification," 3rd Generation Partnership Project (3GPP), TS 36.321, July 2013.

[34] ——, "Study on Architecture for Next Generation System," 3rd Generation Partnership Project (3GPP), TS 23.799, Dec. 2016.