LTE-Xtend: Scalable Support of M2M Devices in Cellular Packet Core

Vasudevan Nagendra, Himanshu Sharma, Ayon Chakraborty and Samir R. Das
Department of Computer Science, Stony Brook University, Stony Brook NY 11794, U.S.A.
{ vnagendra, hisharma, aychakrabort, samir }@cs.stonybrook.edu

ABSTRACT

M2M (machine-to-machine) communications are bringing new challenges in the cellular networks as billions of such devices will need to be supported but at a fraction of the cost of today's smartphones. Analysis shows that while many of these devices generate little data load on the network, their control signaling load and memory/CPU resource consumption due to tunnel maintenance in the cellular core could still be significant. To address this scalability issue, we propose a modified packet core architecture for LTE networks called LTE-Xtend that customizes control message handling and tunnel management for M2M traffic. Evaluations using OpenAirInterface demonstrates significant scalability benefits in using LTE-Xtend.

CCS Concepts

•**Networks → Network performance analysis; Wireless access points, base stations and infrastructure;**

1. INTRODUCTION

Scalable and cost-effective M2M communication is central for future success of emerging technology areas such as Internet of Things and cyber-physical systems. Since M2M devices are expected to be ubiquitous there is a strong interest in supporting them directly in the cellular networks. Cellular connectivity also helps in better management and security of such devices as all supported devices can be under the direct control of the same network operator. While at the current time the number of M2M devices on cellular network is only modest relative to phones, industry trends indicate a significant rise [2]. For example, according to analysis [3, 4] there will be around 26 to 50 billion such devices on the network by year 2020.

M2M devices are heterogeneous as their applications are diverse, ranging from smart grid to medical telemetry. Many of them are also very limited in communication and thus require very little support from the network. For example, many M2M devices may not be mobile, may not receive incoming calls, or may generate very little data. This is in contrast to smart phones that are all functionally very similar, highly mobile and often very data intensive. The current generation cellular networks are fundamentally geared towards handling these homogeneous set of smart phones effectively. The pricing model also supports smart phone centric operation that may not apply to M2M devices. For example, a subscriber may not mind paying US$25-50 per month for a data plan of smartphone, but may not be willing to pay more than a few cents to support a smart light bulb. While such expectations may be justified, in the current generation 3GPP LTE networks, the packet core (evolved packet core or EPC) does not treat smart phones and M2M devices differently so as to reduce the resource cost for the latter. While M2M devices may offer much less data load, they still utilize much of the similar functionalities as the phone. Thus, significant re-architecting of the packet core must happen before billions of such devices are on the network without unduly expensive additional infrastructure.

In this work we propose LTE-Xtend, an extended LTE EPC architecture that adds a set of modest customization of standard EPC mechanisms to efficiently support various M2M devices. The idea is to provide scalable support without significant investment in new infrastructure. The broad idea is to match the resource provisioning to the actual services needed by specific M2M devices. We investigate two types of customizations:

1. Implementing group-based centralized tunnel and policy management mechanisms to (i) optimize M2M device's 'attach/activation' time to the network, (ii) optimize the resource consumption by choosing right tunnel option and right gateway node combination specific to the M2M device, (iii) reduce the number of states and tunnels needed to handle M2M connections by properly grouping them together on basis of their device and traffic characteristics, and (iv) reduce the amount of control signal messages in the EPC among the grouped devices.
2. Building M2M specific EPC gateway components to specifically address the M2M traffic requirements for enhanced performance and optimum resource utilization.

The LTE-Xtend architecture consists of a simple set of extensions and customizations that are incrementally deployable in the current generation EPC. It does not need a disruptive redesign of the cellular core [10], nor requires any extra physical infrastructure support (such as relay nodes) for aggregating the uplink M2M traffic among several devices [11]. It can indeed benefit from virtualization and software-

AllThingsCellular'16, October 03-07, 2016, New York City, NY, USA
© 2016 ACM. ISBN 978-1-4503-4249-0/16/10...$15.00
DOI: http://dx.doi.org/10.1145/2980055.2980062

defined controls [12, 8] though we do not explore this here. Similarly, the 3GPP study on low-cost provisioning of machine type communication (MTC) over LTE mainly focus on access network optimizations [5] neglecting the core network optimizations, which we deal here in our paper.

In the rest of the paper, we present the motivation behind EPC customization for M2Ms in §2 and present the modified architecture in §3. Then, we present preliminary performance evaluations in §4.

2. MOTIVATION FOR LTE-XTEND

In this section, we first provide an overview of the key functional components of LTE architecture. Then, we highlight the limitations of the LTE core for M2M traffic, followed by details on LTE extensions proposed in this paper.

2.1 Overview of the LTE architecture

As shown in Figure 1, conventional LTE cellular network consists of two main components: LTE Radio Access Network (RAN) and EPC. The User Equipment (UE) communicates to the Internet through eNodeB (enhanced NodeB) of RAN via EPC. For routing the data traffic of each user between the User Equipment (UE) and Internet, DRB (Data Radio Bearer) and GTP (GPRS Tunneling Protocol) based tunnels are used. DRB is established between the UE and eNodeB over a radio channel. The GTP based tunnels (named S1-u and S5) are created between eNodeB, Serving Gateway (SGW) and Packet data network Gateway (PGW) nodes in the uplink (UL) and the downlink (DL) directions. Each end of the GTP tunnel is identified using a tunnel endpoint identifier (TEID). The Mobility Management Entity (MME) stands responsible for the control plane functionality. MME verifies the subscription details of a user for authentication with Home Subscriber Server (HSS). The Policy and Charging Rules Function (PCRF) makes policy decision for each subscriber useful in controlling data-flow through PGW with specified QoS and bills the data usage respectively.

The MME interacts with SGW for data session establishment and release procedures. The `Attach request` message is used by a UE to register itself with the EPC for obtaining connectivity to the internet. Similarly, the `service request` (or device activation) is performed when an inactive UE in Idle state wishes to get activated to send or receive the data from/to UE. These two control procedures triggers a sequence of control messages to complete the attach/activation procedures inside EPC such as `create-session request (response)`, `delete-session request (response)`, `initial-context request (response)` and `modify-bearer request (response)`. For convenience, we name these messages as `EPC control messages`.

2.2 Limitations of LTE Core for M2M Traffic

Analysis of M2M traffic clearly indicates that M2M devices generate only modest data load on the cellular network, often only little data (bytes) at long periodic or semi-periodic intervals (minutes) [13, 9]. Overall this may account for few MBs per month of data versus GBs per month in the case of average smartphone users [2]. Simple back-of-the-envelop calculations based on this data shows that more than 70% of the M2M devices generate data less than 1/100$^{\text{th}}$ of the average data traffic sent by a smartphone user per month. While they carry much less data the M2M devices retain the same control signaling architecture as the smartphones. Thus, control load in M2M traffic is relatively higher perhaps much higher when normalized to data load. The control signal load can easily create bottlenecks when one or two orders of magnitude more M2M devices are on the network relative to the phones.

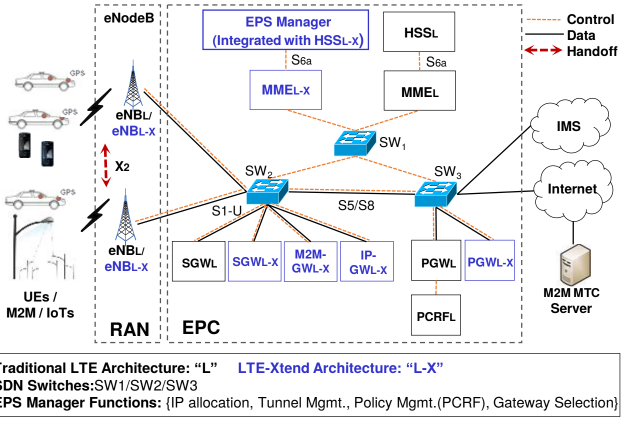


Figure 1: Conventional LTE and LTE-Xtend system architectures. LTE-Xtend components are shown in a different color. L and L-X subscript indicates components for conventional LTE and LTE-Xtend, respectively.

A second issue is related to inefficient resource usage. Consider two types of M2M devices, a medical device which is of `Device-Originated Only data` (device only originates but does not receive data) type and an industrial IoT device which is of `No Mobility` type. When these device types are attached to the LTE network, it is provisioned with full-fledged tunnels (i.e., S1-u(UL/DL) and S5(UL/DL)) in the EPC. Clearly a subset of these tunnels are unnecessary, e.g., DL tunnels for `‘Device-Originated Only data’` or both S5 UL/DL tunnels for `‘No Mobility’`. Maintaining GTP tunnels unnecessarily is a significant waste of resources. Maintaining tunnel state requires non-negligible amount of memory (§4) and also consumes CPU resources for tunnel routing lookup.

2.3 Extending LTE Core for M2Ms

The limitations mentioned in §2.2 can be effectively addressed by customizing the EPC, by taking into consideration the device type and its traffic requirements while provisioning the resources. We will explore several customizations highlighted below.

Tunnel Option Customization. Instead of treating all devices uniformly regarding tunnel maintenance, we categorize M2M devices into five different categories and provision them with different tunnel options (T2 through T6, with T1 indicating unmodified LTE). In each of the tunnel option the number of tunnels and the type of tunnel (i.e., uplink or downlink) provisioned depend on the device's traffic and mobility characteristics as described in §3.2.2.

Group Tunneling. The categorized M2M devices are then grouped i.e., shares the same state space, data structure and tunnels (§3.2.3).

Control Message Customization. As the attach and service request messages constitutes more than 60% of the total control messages [6], we focus on these messages and optimize them by parallelizing the sequence of attach and service request procedures and effectively reducing the number of control messages with the help of centralized tunnel and policy management approach (§3.2.1).

Control Message Grouping. The control message grouping is associated with group tunneling discussed above, i.e., the control messages associated with the devices belonging to the same group tunnels are grouped together as described in §3.2.4. We will later see that this benefits in terms of control messages reduction and CPU load optimization (Figure 3).

Also, to handle different types of M2M devices, the existing gateway nodes (SGW and PGW) are customized and a special gateway node (M2MGW) is designed to cater the specific needs of M2M devices. In the following section, we present our approach describing the EPC customizations.

3. LTE-XTEND SYSTEM DESIGN

The main goal of our architecture design is to provide scalable and efficient solution to handle M2Ms in the LTE core. Figure 1 illustrates the comparative functional block diagrams of basic LTE and LTE-Xtend architectures. The following functional and infrastructure components described in §3.1 are built on top of the existing LTE architecture to support the customization mechanisms described in §3.2.

3.1 Functional Blocks of LTE-Xtend

Evolved Packet System (EPS) Manager. The EPS manager is a centralized tunnel and policy management module designed to optimize resource utilization and reduce control message load in the EPC of LTE-Xtend. The following existing functions of basic LTE access and core networks are migrated to centralized EPS Manager to accommodate M2Ms efficiently at scale, (i) `IP allocation` module, for providing centrally the M2M device with an IP address, (ii) `Policy and charging Rules Function (PCRF)`, to specify the policies to be enforced for M2M traffic inside the core network, and (iii) `TEID creation` module to provide unique tunnel identifiers for establishing GTP tunnels between eNodeB, SGW, and PGW nodes.

The following two new control functions are integrated into EPS Manager; (i) `Gateway selection` module, to centrally select the gateway nodes (SGW and PGW) required for a specific M2M connection to establish the data tunnels, (ii) `tunnel option selector` module to provide a specific tunnel option (§3.2.2) for a M2M connection considering the device's traffic and QoS requirements.

The EPS manager module is integrated with HSS to handle new M2M connections and assigns the necessary QoS policies, tunnel identifiers and selects respective gateway nodes for establishing the necessary data tunnels (S1/S5) between them.

Customized eNodeB, MME, S/PGW Nodes. The eNodeB and gateway nodes are customized to handle the extensions proposed in §2.3 pertaining to `control message customization`, `control message grouping`, `tunnel customization` and `group tunneling`.

3.2 Customizations Supported in LTE-Xtend

The high level details of the extensions and customizations done to the basic LTE architectural components (eNodeB and EPC) for supporting M2M devices are described below.

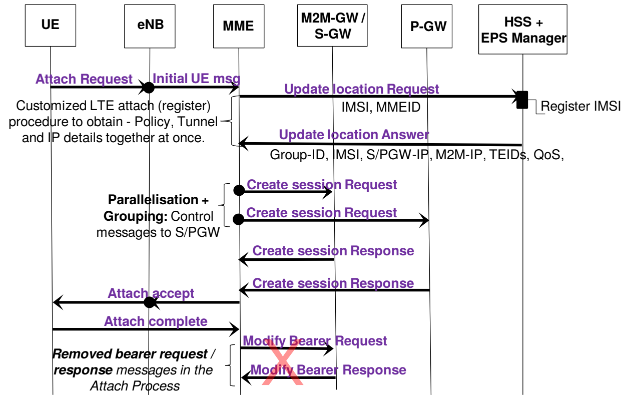


Figure 2: LTE-Xtend M2M/UE attach procedure control flow diagram.

3.2.1 EPC Control Message Customization

In the current LTE architecture, MME lacks a direct control channel interface to manage the PGW and PCRF modules. Hence, the vital control functionalities such as `M2M/UE IP allocation`, `TEID creation`, `Gateway selection` and `Policy specification` are carried out in a distributed fashion across eNodeB, SGW, PGW, and PCRF modules. This distributed approach increases both latency and the number of control messages required to complete the device attach/activation procedures, thereby increasing the control message load on the gateway nodes (SGW and PGW).

In LTE-Xtend, we address this limitation and optimize the device attach/activation procedures by separating only the control plane functions specific to tunnel and policy management modules and centralizing them. The following functions `IP allocation`, `Policy specification`, `Gateway selection` and `Tunnel ID creation` are separated from the data plane (eNodeB, SGW and PGW) and PCRF modules for centralizing them with the `EPS Manager`. This centralized architecture helps us to optimize the number of control messages required to complete the attach/activation procedures and allows control message parallelization (between the MME and gateway nodes) thereby reducing the time required to complete these two control procedures.

Attach request customization. In the traditional LTE architecture, when a device (UE/M2M) attaches to the core network, the `create-session` request message is generated by the MME and exchanged between the SGW, PGW and PCRF modules performing specific set of control functions to complete the attach procedure. As part of the session creation procedure following control messages are also exchanged inside EPC for (i) assigning IP address to the UE or M2M device, (ii) provisioning QoS to gateway nodes, (iii) SGW and PGW selection mechanism and (iv) exchanging the end point tunnel identifiers of S1-u and S5 tunnels between the eNodeB and S/PGW nodes. With the centralized LTE-Xtend architecture, the above mentioned control messages are composed together into a single control message and the control messages exchanged between MME and gateway nodes are parallelized as shown in Figure 2. The optimizations described above reduces both the attach time and number of messages required to complete the attach procedure (Table 1).

Also, in the basic LTE attach procedure, the `modify-`

bearer request and response messages are used by the MME to update the SGW with TEID details of eNodeB. In the LTE-Xtend, Since the tunnel identifier (TEID) creation and distribution happens from the centralized EPS manager, it removes the need for these two messages in the attach procedure as shown in Figure 2.

Service request customization. Similar benefits can be obtained for the service request procedures with centralization approach. The control messages exchanged between the MME, SGW and PGW (i.e., `context-setup request (response)` and `modify-bearer request (response)` messages) are effectively optimized and parallelized to reduce the time required to complete the device activation procedure (Table 1).

As most of the control functions required to complete the attach/service request messages are handled together in the EPS manager, this reduces the number of control messages exchanged inside the EPC as shown in Figure 2. Approximately 40% improvement to the attach/activation time is obtained through this process as shown in Table 1.

3.2.2 Tunnel Option Customization

In the current LTE architecture, GTP based data tunnels are used for supporting QoS, traffic aggregation and mobility needs of the user's data traffic. By default, the EPC provisions S1-u and S5 data tunnels in both the directions (uplink and downlink) to all the devices irrespective of their needs. But, with complete shift of traffic characteristics with M2M communications, the fundamental question arises about the need of GTP for M2M devices and the cost involved with it as highlighted in the §2.2. With increase in the number of GTP tunnels in the system, the GTP TEID look up cost as well as tunnel en-/decapsulation overhead and tunnel state maintenance cost increases.

In LTE-Xtend, we address this by optimizing the existing GTP tunnel management mechanism to allocate the necessary GTP tunnel considering the traffic characteristics in any specific direction (uplink or downlink or both). The EPS manager plays a key role in choosing the gateway nodes needed for a specific tunnel option and deciding on the type of tunnels configured for any M2M device.

In LTE-Xtend different tunnel options are proposed by categorizing the M2M connections according to their traffic requirements, devising new tunnel option for each category (i.e., T2 through T6). **T1** represents the traditional LTE architecture, which uses both SGW and PGW with S1 and S5 tunnels.

(i) **T2** is designated to M2M devices that requires the QoS for their traffic and also have mobility needs, e.g., Intelligent transport systems, smart cars, asset tracking system and so on. This tunnel option uses both SGW and PGW nodes along with S1 and S5 data tunnels.

(ii) **T3** is used for M2M devices that require QoS, but no mobility e.g., Industrial/Smart City IoTs, and so on. T3 uses customized SGW module with S1 tunnel only.

(iii) **T4** is used for M2M devices that need only transmitting capabilities with QoS and mobility e.g., Medical, public safety IoTs. T4 uses both S/PGW nodes along with only uplink (S1, S5) tunnels.

(iv) **T5** is used for stationary M2M devices with only data transmission capabilities e.g., public safety, home automation IoTs. T5 uses customized SGW with only S1(UL) tunnel. For both the tunnel options T4 and T5, the Downlink traffic (if any) is handled by simple IP based routing.

(v) **T6** is used for devices that needs mere IP connectivity to update their liveliness. e.g., smart city IoTs. T6 uses simple IP forwarding without a need for GTP.

3.2.3 Group Tunneling

Considering M2M traffic characteristics [13], it is evident that more than 50% of M2M devices generate flows that are much less than 5 minutes per hour or generates only few bytes of data per transaction. But, the actual duration for which the data tunnels and their states are maintained ranges on an average from 20 to 30 minutes. Hence, we take this as an opportunity to group the data tunnels (and their states) of different M2M connections having similar data traffic characteristics 'or' same QoS requirements for optimizing the resource utilization.

This is accomplished during the `attach` process i.e., when a M2M device attaches to the LTE network, the EPS manager verifies the device properties and its traffic requirements to select a respective tunnel option. The M2M connection is either assigned to an existing connection having same tunnel option or a new set of data tunnels are created (if no tunnels exist with the same tunnel option). The new M2M connection details, QoS values and other tunnel related parameters are reconfigured to group tunnel data structure accordingly. The group tunneling mechanism is implemented on top of tunnel option customization suggested in §3.2.2.

Mapping M2M devices to Group tunnels. In the traditional LTE architecture, the eNodeB manages a UE contextual information for each of the connections maintaining mapping between the radio bearer and the GTP data tunnel. This helps the packet received at the eNodeB to be forwarded to and forth between the radio bearer and the GTP data tunnel on the eNodeB. But, the same is not possible in the case of group tunneling as the user data from a group tunnel need to be properly mapped back on to the respective radio tunnel. Hence, to handle the data traffic across different radio bearers and the GTP tunnels at eNodeB a unique identifier is required. In our case, M2M IP address is used as the unique identifier to perform this mapping. At eNodeB, the following piece of data-structure is used to map radio bearer with the GTP tunnel which is commonly maintained in the datastructure of both the radio and gtp tunnels at eNodeB. i.e., '`rnti`, `RABID (Radio Access Bearer Identifier)`, `M2M IP address` and `TEID`'. We use hash of this mapping detail mentioned above for faster look-up of Group TEID to respective RAB-ID for forwarding the reply packets back to the respective M2M device through radio bearer.

3.2.4 EPC Control Message Grouping

The centralized management infrastructure and `Group Tunneling` mechanism provides an opportunity to further optimize the resource utilization and reduce the control message load in the EPC. This is achieved by grouping the control messages of M2M devices in the EPC belonging to the same group tunnel. This is handled by customizing the gateway nodes and MME to capably group all the control message that arrives either at MME or gateway nodes. The control messages are grouped on the basis of the their tunnel identifiers i.e., a group tunnel identifier that is maintained in common for both group tunneling and control message grouping approaches.

Table 1: Average memory utilized per M2M connection (states and tunnels), latency in-cured by data packet in EPC core and their attach/activation times with OAI[7].

Tunnel Schemes	Memory (KB)	Latency (ms)		
		Latency in EPC	Service Request	Attach Request
T1	93.15	4.2	1030	1503
T2	83.8	4.1	685	995
T3	49.1	2.25	593	954
T4	62.4	3.2	615	987
T5	38.8	1.6	557	948
T6	18.45	1.1	471	796

The LTE-Xtend provides the provision to custom specify the control message grouping `buffer wait times` i.e., duration for which a specific message can be queued for grouping purposes. This provides the user an option to configure the control message buffer wait time depending on the type of the M2M devices or their characteristics. For example, M2M devices that are delay-insensitive, the buffer time can be configured as high as in seconds. Also, the messages like delete-bearer request messages are buffered much longer as they impose very little impact on the performance.

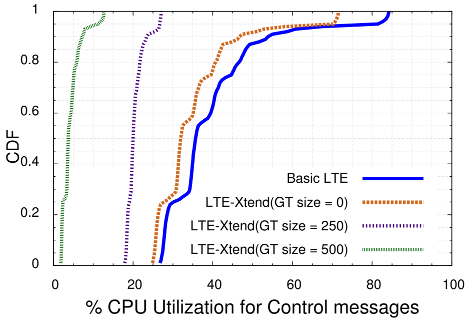


Figure 3: CPU utilization of control messages for different group tunnel sizes.

4. PERFORMANCE EVALUATION

In this section we discuss implementation details of LTE-Xtend followed by description of our test-bed and experimental results. We showcase how LTE-Xtend can scale better than traditional LTE architecture under different resource constraints (e.g., CPU utilization, memory footprint, latency) in the core network.

4.1 LTE-Xtend Testbed

We build LTE-Xtend on top of OpenAirInterface (OAI) [7], an open source LTE framework running on the Linux platform. OAI provides implementation of the LTE air interface/client, eNodeB and the EPC modules. We modify the eNodeB and EPC modules to implement the custom tunneling options T2 through T6, as discussed in §3.2.2. Our testbed consists of 6 systems that house different components of LTE-Xtend (i.e., M2M test clients, OAISIM as eNodeB, OAI SGW + M2MGW, nwEPC as PGW, OAI MME and M2M Test Server).

Traffic Generator: To make our M2M traffic as realistic as possible we depend on [13] that characterizes M2M traffic on a large scale. We create `ON/OFF` traffic using both TCP and UDP connections carrying variably sized packets (128, 512 and 1024 Bytes) at different bitrates. About 90% of the devices maintain persistent TCP connections with (keep-alive) session lengths varying from ≈1 minute to 1 hour, while about 10% devices generate bursty traffic with each burst ranging from 1 second to 10 seconds.

eNodeB and EPC: All connections go through the eNodeB implemented using OAISIM that routes them to the serving gateway (SGW). The SGW is implemented using the OAI's *SGW-Lite* module. However, OAI does not provide an implementation of the packet gateway (PGW). We implement this using the *nwEPC* [1]. Other essential components (MME, HSS etc.) are implemented directly using the respective OAI modules. The PGW forwards the traffic to a sink server that hosts the other end point of the connection. The PGW and the server are both connected to our lab's high speed Ethernet network.

Logging Modules: We deploy Python-based logging modules across different components in our testbed to log various timing information and resource utilization (CPU, memory etc.) related attributes. We make sure that the logging process does not create significant resource overhead (CPU utilization less than 1%).

We provision our setup to maintain atleast 1000 simultaneous connections using the tunneling option T1. Using our testbed we can generate both data traffic as well as control traffic. To explicitly account for the control traffic we use the OAISIM module to trigger the attach and service request messages without generating any data traffic. The data traffic is generated from separate M2M clients to be sent through the eNodeB.

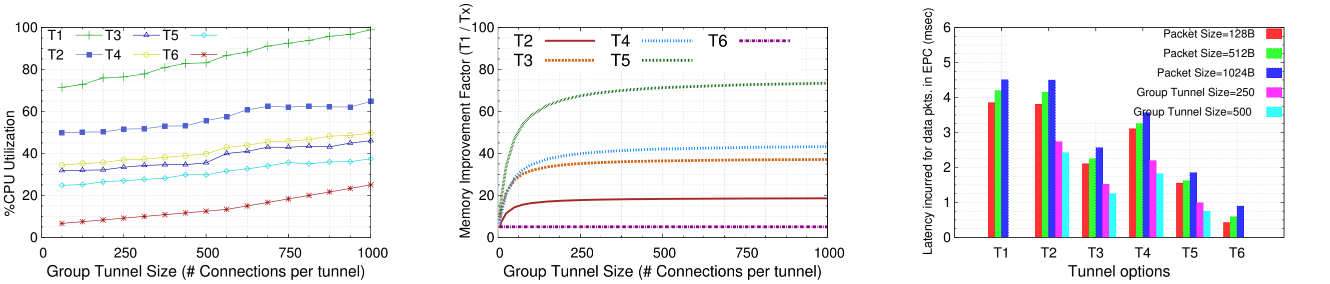
4.2 Experimental Results

We generate traffic corresponding to each of the tunneling options T1 through T6 lasting approximately for 8 hours. The experiments are run for a sufficiently long time to exhaustively emulate different patterns of traffic (varying packet sizes, bitrates etc.). Second it also helped us to weed out measurement noise. Our evaluation results are based on trace-logs collected from such experiments. In the following we present the benefits achieved in terms of resource overhead using the group tunneling mechanism with different tunnel options of LTE-Xtend across both control and data planes respectively.

Benefits in the Control Plane: We perform a comparative benchmark of CPU utilization for basic LTE and LTE-Xtend. For LTE-Xtend we create group sizes of 250 and 500 connections per tunnel. Figure 3 shows a CDF of CPU utilization across all such configurations. The results show significant savings in terms of CPU utilization (≈ 1.5× – 10× improvement in median). Given that control messages are prevalent in non-trivial proportions, the results demonstrate how LTE-Xtend can scale without overwhelming the infrastructure.

Benefits in the Data Plane: In this case we showcase the benefits in terms of CPU utilization, memory footprint and latency improvement in the core. We increase the group size from 50 till 1000 in steps of 25.

- **CPU Utilization:** Figure 4a shows the average CPU uti-



(a) CPU utilization for different tunneling options (T1 – T6).

(b) Memory footprint improvement for tunneling options (T2 – T6) over T1.

(c) Latency incurred in the core for different packet sizes/group sizes.

Figure 4: Improvement in resource utilization in data plane for LTE-Xtend compared to basic LTE.

lization for different tunnel options with increasing group size. The CPU utilization for different tunneling options in LTE-Xtend shows an improvement of 1.5 (in case of T2) to as high as 7 (in case of T6) over the basic LTE scheme (T1).

- **Memory Footprint:** With grouping enabled, LTE-Xtend offers a huge improvement in memory footprint for scaling the number of connections. Figure 4b demonstrates an improvement factor of 60-80 times when compared to T1 (basic LTE).
- **Latency Incurred at the Core:** We measure that each GTP encapsulation and decapsulation process costs around 80μs to 250μs. With the full-fledged basic LTE architecture the time spent in the GTP module is calculated to be around 640μs to 2ms (considering both the UL/DL directions). With the group tunneling option the TEID look-up cost at the eNodeB and SGW for routing the GTP tunnel is reduced considerably. The overall average performance improvement inside the EPC core is shown in the Figure 4c, with the results shown for various tunnel options and for group tunnel sizes of 250 and 500. The latency is cut down to half or more in tunnel options T5 and T6 compared with traditional LTE architecture (≈1 to 2ms vs. ≈4ms).

5. CONCLUSION

In this paper we have proposed a modified EPC architecture called LTE-Xtend that augments the standard 3GPP EPC architecture to handle M2M traffic in a scalable fashion. The basic idea is to improve resource efficiency significantly for M2M traffic. The system is implemented using the OAI [7] software stack and extensively benchmarked for M2M class traffic. Results show upto 10× improvement in CPU utilization and upto 80× improvement in memory overhead for both data traffic and control messages. Our work shows a good promise to support the ongoing trend of M2M devices while optimizing operational costs for the infrastructure.

6. ACKNOWLEDGMENTS

The work is partially supported by NSF grant AST-1443951.

7. REFERENCES

[1] EPC SAE Gateway.
https://sourceforge.net/projects/nwepc/files/.

[2] Ericsson Mobility Report, June 2016.
http://www.ericsson.com/res/docs/2016/
ericsson-mobility-report-2016.pdf.

[3] Gartner Says the Internet of Things Installed Base Will Grow to 26 Billion Units By 2020, December 2013. http://www.gartner.com/newsroom/id/2636073.

[4] Internet of Everything (IoE) connection counter, July 2013. https://newsroom.cisco.com/feature-content?articleId=1208342.

[5] Low cost M2M over LTE. http://www.3gpp.org/news-events/3gpp-news/1714-lc_mtc.

[6] Managing lte core network signaling traffic, (2015).
http://www.tmcnet.com/tmc/whitepapers/
documents/whitepapers/2015/
11134-reduce-core-network-signaling-with-field-proven-mme.pdf.

[7] OpenAirInterface (OAI): Towards Open Cellular Ecosystem.
http://www.openairinterface.org/?page_id=864.

[8] H. Baba et al. Lightweight virtualized evolved packet core architecture for future mobile communication. In *Proc. IEEE Wireless Communications and Networking Conference*, 2015.

[9] J. Jermyn et al. Scalability of machine to machine systems and the internet of things on lte mobile networks. In *Proc. IEEE World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2015.

[10] Jin Xin et al. SoftCell: Scalable and Flexible Cellular Core Network Architecture. In *Proc. ACM CoNEXT*, 2013.

[11] Marwat Safdar Nawaz Khan et al. Data aggregation of mobile M2M traffic in relay enhanced LTE-A networks. In *EURASIP Journal on Wireless Communications and Networking*, 2016.

[12] Qazi Zafar Ayyub et al. KLEIN: A Minimally Disruptive Design for an Elastic Cellular Core. In *Proc. ACM SOSR*, 2016.

[13] Shafiq M. Zubair et al. Large-scale Measurement and Characterization of Cellular Machine-to-machine Traffic. In *Proc. IEEE/ACM Transactions on Networking*, 2013.