

特別研究報告

題目

仮想化モバイルコアネットワークの機能配置のための
シグナリングプロトコル処理遅延の実験的評価

指導教員

松岡 茂登 教授

報告者

上野 真生

平成 29 年 2 月 14 日

大阪大学 基礎工学部 情報科学科

仮想化モバイルコアネットワークの機能配置のための
シグナリングプロトコル処理遅延の実験的評価

上野 真生

内容梗概

近年におけるスマートフォン等の携帯端末利用者の増加，及び端末の高機能化によって，Long Term Evolution (LTE) や第 5 世代移動通信システムなどのモバイルネットワークにおける，輻輳への対応が課題となっている．また，特に最近では，モバイルネットワークに接続される端末として Machine-to-Machine (M2M) や Internet of Things (IoT) 端末が注目されており，トラフィック量が増加することが指摘されている．M2M/IoT 通信を行う端末は，従来の携帯端末とは異なり，通信データ量や通信頻度は少ないものの，それらの端末数が膨大になるという通信特性を持つことがある．そのため，M2M/IoT 端末による輻輳の問題に対応するには，従来の携帯端末とは異なる接続方式を用いることが，標準化団体などで検討されている．

我々の研究グループでは，この問題に対し，M2M/IoT 端末を収容することに適したモバイルコアネットワークについての研究を進めている．特に，端末収容能力を向上させるためのコアノード機能の配置方法や，通信負荷を軽減するための手法を検討し，その有効性を性能評価を行うことで示してきた．しかし，これらの性能評価は待ち行列理論に基づく数学的解析や，ソースコードの命令文数を基にしている．より現実的な評価を行うためには，モバイルコアネットワークノードの実ソフトウェアを用いた実験が必要と考えられる．

そこで本報告では，モバイルコアネットワークを構築するための実ソフトウェアを用い，端末が通信を行う際に必要となるシグナリングプロトコルの処理遅延時間を評価した．具体的には，複数の仮想マシン上でモバイルコアネットワークノードを稼働させ，特に多くのシグナリングメッセージ処理を行う MME，及び S-GW/P-GW ノードに対して，CPU 負荷を与えた環境において，1 台の UE がネットワークに接続されるまでに要する時間を計測した．その結果，EPC ノードに対して高い CPU 負荷を与えた場合において，1%程度の確率で，平均値よりも 10 倍から 100 倍の大きな値が観測されることが明らかになった．これらの結果を用いることで，仮想化されたモバイルコアネットワークのノード機能の配置や，必要な資源量に関する知見が得られた．

主な用語

モバイルコアネットワーク, M2M/IoT 通信, Long Term Evolution (LTE), シグナリング処理, virtualized Evolved Packet Core (vEPC)

目次

1	はじめに	6
2	モバイルコアネットワーク	8
2.1	ネットワーク構成	8
2.2	シグナリング処理手順	9
3	実験方法	13
3.1	実験環境	13
3.1.1	モバイルコアネットワークソフトウェア	13
3.1.2	実験ネットワーク構成	13
3.2	実験手順	15
4	実験結果と考察	19
4.1	ベアラ確立時間	19
4.2	MMEにCPU負荷を与える場合	19
4.3	SP-GWにCPU負荷を与える場合	21
4.4	MMEとSP-GWの両方にCPU負荷を与える場合	28
4.5	考察	28
5	まとめと今後の課題	31
	謝辞	32
	参考文献	33

目 次

1	評価対象とするモバイルコアネットワークアーキテクチャ	11
2	シグナリング処理手順	12
3	実験ネットワーク環境	14
4	ベアラ確立時間の変化	20
5	MME のみに負荷を与えた場合のシグナリング処理時間	22
6	MME のみに負荷を与えた場合の総シグナリング処理時間	23
7	負荷プロセス数を用いた場合のシグナリング処理時間	24
8	負荷プロセス数を用いた場合の総シグナリング処理時間	25
9	SP-GW のみに負荷を与えた場合のシグナリング処理時間	26
10	SP-GW のみに負荷を与えた場合の総シグナリング処理時間	27
11	MME と SP-GW に負荷を与えた場合のシグナリング処理時間	29
12	MME と SP-GW に負荷を与えた場合の総シグナリング処理時間	30

表 目 次

1	各ノードを実装したサーバの仕様	14
2	MME/SP-GW に与える CPU 負荷の組み合わせ	16

1 はじめに

近年におけるスマートフォン等携帯端末利用者の増加, 及び端末の高機能化によって, Long Term Evolution (LTE) [1] や第5世代移動通信システム [2] などのモバイルネットワークにおける, 輻輳への対応が課題となっている. また, 特に最近では, モバイルネットワークに接続される端末として Machine-to-Machine (M2M) [3] や Internet of Things (IoT) [4] 端末が注目されている. 文献 [4] によると, 2020 年までに, インターネットに接続される IoT 端末の数は 2,120 億台になり, 2022 年までには, M2M 通信を行う端末のトラフィックは, ネットワーク全体の 45% を占めるようになるとされている.

M2M/IoT 通信を行う端末は, 従来の携帯端末とは異なり, 通信データ量や通信頻度は少ないものの, それらの端末数が膨大になるという通信特性を持つことがある. このような特徴を持つ M2M/IoT 端末をネットワークに接続する一つの方法として, 従来のセルラ無線とは異なる, Low Power, Wide Area (LPWA) [5] と呼ばれる, 省電力でありかつ広域をカバーすることが可能な無線ネットワークが提案されている. 日本国内においても, 2016 年 9 月にソフトバンク株式会社が LPWA ネットワークの一つである Long Range WAN (LoRaWAN) [6] の提供を発表し [7], 同年 11 月には株式会社 NTT ドコモも LPWA 通信の実証実験を開始することを発表した [8]. また同月には, 京セラコミュニケーションシステム株式会社が, LPWA ネットワークの一つである SIGFOX [9] を日本で展開することを発表するなど [10], 非セルラ無線ネットワークによる M2M/IoT 通信収容の広がりが見られている. しかしながら, LPWA を実現するためには, 新たなインフラ網を構築する必要があるなど, コスト面における課題が存在している. これに対し, 従来のセルラ無線に M2M/IoT 端末を収容する方式では, 既存のインフラ網を有効に活用できるという利点がある. その反面, 多数の M2M/IoT 端末を収容すると, それらの端末の通信特性に起因する輻輳の問題が発生すると考えられている. そのため, 従来の端末とは異なる接続方式を用いることが, 3GPP 等の標準化団体などで検討されている [11].

我々の研究グループでは, この問題に対し, M2M/IoT 端末を収容することに適したモバイルコアネットワークについての研究を進めている. 文献 [12] [13] では, M2M/IoT 通信において, 従来の携帯電話やスマートフォンと異なる通信の特徴を踏まえ, LTE/EPC ネットワークの実アプリケーションソフト OpenAirInterface (OAI) [14] を用いてモバイルコアネットワークの性能評価を行っている. 具体的には, 複数のモバイルコアネットワークアーキテクチャのモデルを検討し, 各モデルについて待ち行列理論に基づく数学的解析を行い, 数値例を挙げることによって性能評価を行っている. 評価の際に必要なシグナリング処理負荷の重みは, LENA および OAI における実装コードの命令文数から決定している. しかし, 実際のシグナリング処理はコンパイル後に得られる実行コードによって行われるた

め、ソースコードの命令文数が実際のシグナリング処理負荷量に対して相関を持つとは限らない。そのため、モバイルコアネットワークを構成するノードの C/U 分離や、各ノードへの資源配分方法を検討するためには、シグナリング処理負荷を実装システム上で観測する必要がある。

そこで本報告では、モバイルコアネットワークを構築するための実ソフトウェアを用い、実験を通じて、端末が通信を行う際に必要となるシグナリングプロトコルの処理遅延時間を評価する。具体的には、まず OAI を用いて複数の仮想マシン上でモバイルコアネットワークノードを構築した。次に、特に多くのシグナリングメッセージ処理を行う MME 及び S-GW/P-GW ノードに対して、人工的な CPU 負荷を与えた環境において、1 台の UE がネットワークに接続されるまでに要する時間を計測する。それにより、モバイルコアネットワークを構成するノード負荷が、シグナリング処理遅延時間に与える影響を明らかにし、多数の M2M/IoT 端末を収容するためのネットワークアーキテクチャを検討するための知見を得る。

本報告の構成は以下のとおりである。2 章では、本報告で評価対象とするモバイルコアネットワークアーキテクチャ、及びシグナリング処理手順について述べる。3 章では、実験で使用する OAI の仕様や、モバイルコアネットワークが稼働する実験環境における各ノードのネットワーク構成、設定内容、及び実験手法や計測手法について説明する。4 章では、3 章で述べた環境下で行った実験の結果について述べ、考察を行う。最後に、5 章で、まとめと今後の課題について述べる。

2 モバイルコアネットワーク

本章では、本報告で扱うモバイルコアネットワークのネットワーク構成、及び1台の UE がネットワークに接続されるまでに処理されるシグナリングについて記述する。

2.1 ネットワーク構成

図 1 に、本報告で評価対象とするモバイルコアネットワークを示す。図 1 のネットワーク内には、携帯端末や M2M/IoT 端末などを表す User Equipment (UE), UE と無線で通信を行い、後述する MME, S-GW と制御信号を交換する基地局である evolved NodeB (eNodeB), 及び, UE の認証や無線網内での移動処理, 外部ネットワークとのパケット通信を制御する Evolved Packet Core (EPC) ノードが存在する。図 1 における各 EPC ノードの機能を以下に示す。

Mobility Management Entity (MME): UE の認証, 無線網内での移動処理やパケットの経路設定の制御などを行い, EPC 内でのシグナリング処理の中核となるノード。

Home Subscriber Server (HSS): ユーザごとの契約情報や認証用のキーデータ, MME のアドレスなどを管理するノード。

Serving Gateway (S-GW): MME からの制御に基づき, UE と P-GW の間で IP パケットの伝達を行うノード。端末が eNodeB 間を移動した際のアンカーポイントとしても機能する。

Packet Data Network Gateway (P-GW): LTE ネットワーク網の外部 (External IP Network) と IP パケットをやり取りするノード。

それぞれのノード間は、以下に挙げる IP ネットワーク上に構築される論理的なインターフェースによって接続されている。

S1-C (S1-MME): eNodeB と MME 間を接続し, eNodeB を経由して UE と制御信号を交換するための制御用インターフェース。

S1-U: eNodeB と S-GW 間を接続し, eNodeB を経由して UE と IP パケットを交換するためのデータ用インターフェース。

S6-a: MME と HSS を接続し, MME の制御に基づき, ユーザ認証情報や位置情報を伝送するための制御用インターフェース。

S11: MME と S-GW を接続し，MME の制御に基づき，UE と S-GW 間のデータ転送経路の設定情報などを S-GW と交換するための制御用インターフェース。

S5/S8: S-GW と P-GW を接続し，ユーザデータを伝送するためのデータ用インターフェース。

SGi: P-GW と外部ネットワークを接続し，外部ネットワークと UE の間で IP パケットを送受信するためのデータ用インターフェース。

2.2 シグナリング処理手順

図 1 に示すモバイルコアネットワークにおいて，UE が外部ネットワークとパケットをやり取りするためには，UE と eNodeB，eNodeB と S-GW，及び S-GW と P-GW の間で，UE ごとに固有のベアラ（トンネル，回線とも呼ばれる）を確立する必要がある。図 2 は，それらのベアラの確立に必要な処理と，ノード間のシグナリングパケットのやり取りを時系列で示したものである。図 2 の主な処理の流れを以下に示す [15]。

1. UE と eNodeB は Radio Resource Control (RRC) プロトコルを用い，UE と eNodeB 間のベアラである Radio Bearer を確立する (RRC Connection Request ~ RRC Connection Complete)。
2. MME は UE およびユーザ情報を HSS に問い合わせ，結果を基に S-GW に対して S5/S8 Bearer を確立する制御メッセージを送信する。S-GW はこのメッセージに基づき，P-GW に対してベアラ確立要求を送り，P-GW はこのメッセージにより，S5/S8 ベアラを確立し，UE 用の IP アドレスを割り当てる (Initial UE Message ~ Create Session Response)。
3. MME は eNodeB と S-GW 間の S1 Bearer を確立するために，eNodeB を経由して UE に対してメッセージを送信する。UE はこのメッセージに対する返信を eNodeB に送信し，この情報を基に S1 Bearer が確立される (Initial Context Setup Request ~ Initial Context Setup Response)。

このように，UE がデータ送受信を開始するまでには，多数のシグナリングメッセージの交換及び処理が必要となる。特に M2M/IoT 端末を収容する場合において，送受信するデータサイズが小さい場合には，相対的にシグナリングメッセージ処理負荷がモバイルコアネットワークに与える影響が大きくなる。一方で，そのような端末は Average Revenue Per Unit (ARPU) が通常の端末に比べて低いため，大量の M2M/IoT 端末を収容することによって

発生するコストを回収することが困難である。そのため、C/U 分離やノードの仮想化、ネットワーク仮想化などの技術を適用することによって、コスト削減を行うことが求められている。

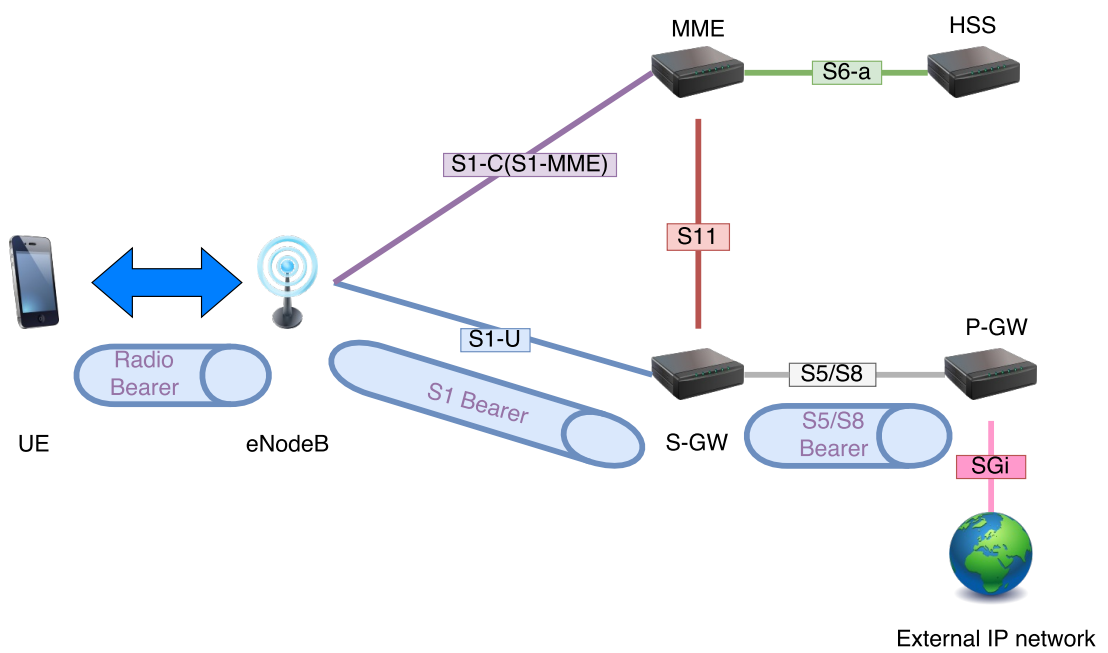


図 1: 評価対象とするモバイルコアネットワークアーキテクチャ

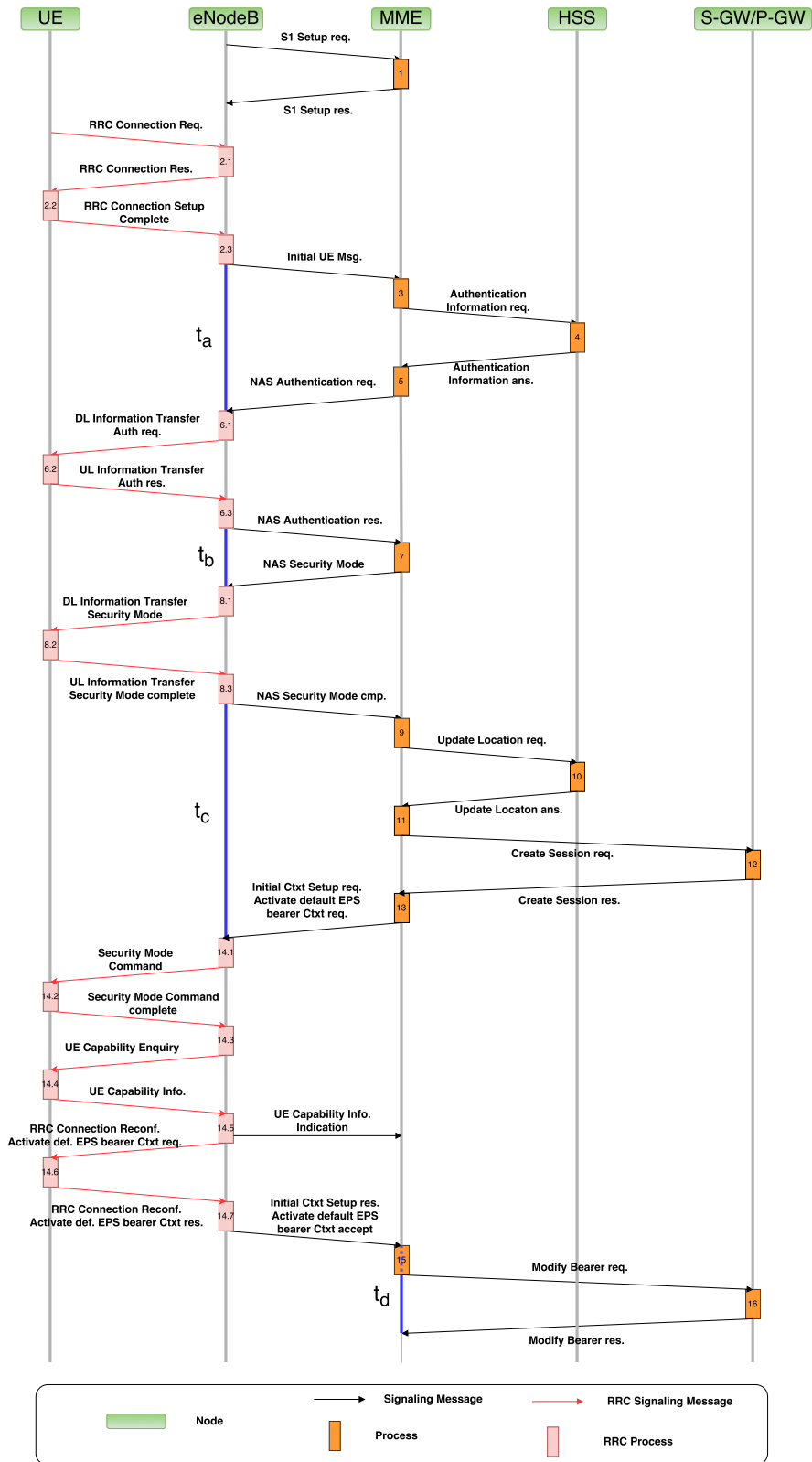


図 2: シグナリング処理手順

3 実験方法

3.1 実験環境

3.1.1 モバイルコアネットワークソフトウェア

本報告における実験に使用するモバイルコアネットワーク, UE, 及び eNodeB は, OAI [14] を用いて実現した. OAI はモバイルコアネットワークの機能を C 言語を用いて実装したオープンソースソフトウェアであり, UE 及び eNodeB を実際の機器, もしくはシミュレータ上で稼働するための機能と, EPC ノードを実現する機能を持つ. OAI においては, EPC ノードである S-GW, P-GW は同一ノードとして実装されており, S5/S8 インターフェースはプロセス間通信によって実現されている. そのため, 以降では S-GW と P-GW を同一ノードとして, SP-GW と表記する.

3.1.2 実験ネットワーク構成

図 3 に, 本報告で使用する実験環境の全体図を示す. 本実験環境においては, MME, HSS, SP-GW を, 1 台の仮想化ホスト上の 1 台ずつの仮想マシンにインストールした. 仮想化ホストは VMWare ESXi 6.0 update 2 を使用した. また, eNodeB および UE は, OAI が提供する OASIM と呼ばれるシミュレータを, 1 台の実マシン上にインストールすることで実現した. 以降では, この実マシンを eNodeB+UE マシンと呼称する. 表 1 に, 各ノードの資源量や OS に関する仕様をまとめた. 実験環境は報告者が所属する研究室のローカルエリアネットワーク内で構築したが, 他のトラフィックの影響を最小限に抑えるため, S11 および S6-a インターフェースは, 研究室内ネットワークから独立したセグメントに設定した.

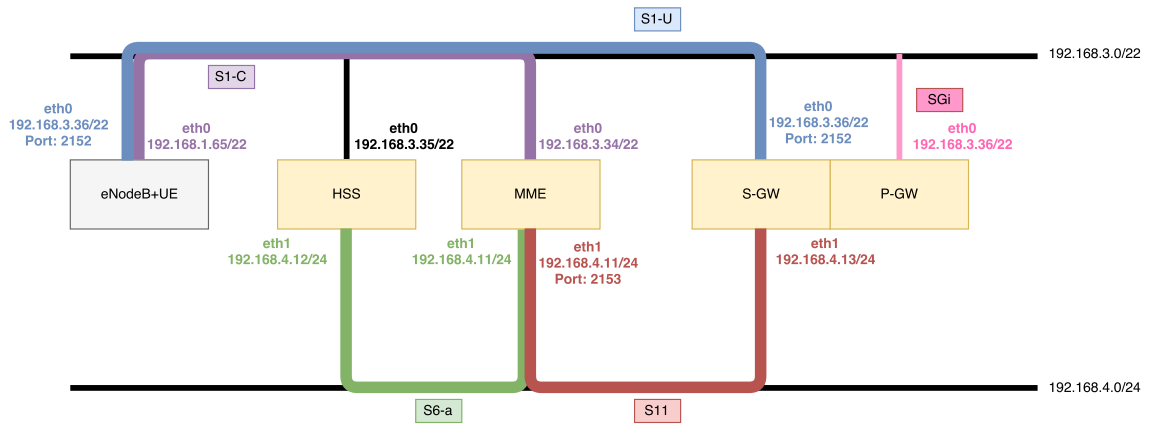


図 3: 実験ネットワーク環境

表 1: 各ノードを実装したサーバの仕様

Node Name	Virtual/Physical	Operating System	Kernel Version	CPU Clock [GHz]	CPU Core	Memory Size [GB]
eNodeB+UE	Physical	Ubuntu 14.04 LTS	4.4.0-62-generic	3.40	8	32
MME	Virtual		3.13.0-24-generic	1	1	4
HSS			4.7.5			
SP-GW						

3.2 実験手順

本報告では、MME および SP-GW に対して人工的に負荷を発生させ、UE がネットワークに接続されるまでの時間、及び各 EPC ノードにおけるシグナリング処理時間を評価する。実験に使用するツールを以下に示す。

stress: マシンに対して様々な人工的な負荷を与えるためのコマンドである。stress コマンドは、与えるオプションによって、以下に挙げる種類の負荷をシステムに与えることができる。

1. `sqrt()` を繰り返し実行することによる CPU 負荷
2. `sync()` を繰り返し実行することによる I/O 負荷
3. `malloc()/free()` を繰り返し実行することによるメモリ負荷

本報告では、EPC ノードに与える負荷として、1. の CPU 負荷のみを用いる。

cpulimit: 指定したプロセスの CPU 使用率の最大値を制御するコマンドである。指定したプロセスに対して SIGSTOP と SIGCONT の 2 つのシグナルを繰り返し送ることによって、プロセスの CPU 使用率の最大値が設定値になるように制御を行う。

top: システム全体のパフォーマンスを監視するためのコマンドである。本報告では、stress によって生成された負荷プロセスが、実際に総 CPU 時間に対してどの程度の CPU 時間を占めているかを監視するために使用する。top コマンドが出力する CPU 使用率は、直前に情報を表示した時点から、次に情報を表示するまでの、総 CPU 時間に対してプロセスの占める CPU 時間から算出される。

tcpdump: pcap ライブラリを利用したパケットキャプチャツールである。実験環境を構成する各ノードにおいて、各シグナリング処理メッセージの到着時刻、及び送出時刻を計測し、シグナリング処理時間を算出するために使用する。tcpdump で計測される時間の精度は 10^{-6} 秒である。これは、本報告における実験によって得られるシグナリング処理時間やベアラ確立時間に対して十分な精度である。

MME および SP-GW に与える CPU 負荷は、stress コマンドと cpulimit コマンドを用いて、表 2 に示す通りの組み合わせで与える。また、MME に対しては、100% を超える負荷として、cpulimit を用いず、stress によって生成される負荷プロセスの数を増やした場合の実験も行う。負荷プロセスの数は 1~4 個とする。

実験は以下の手順で実行する。

表 2: MME/SP-GW に与える CPU 負荷の組み合わせ

MME [%]	SP-GW [%]
-	-
20	-
40	-
60	-
80	-
100	-
-	20
-	40
-	60
-	80
-	100
20	20
40	40
60	60
80	80
100	100

1. ノードにおいて、 $n\%$ の負荷を与える場合には、以下のコマンドを実行することにより、CPU 負荷を発生させる。

```
stress -c 1 &  
cpulimit -l n -p <PID of stress>
```

ノードに与える CPU 負荷を、負荷プロセス数を用いて変化させる場合には、以下のコマンドによって、 m 個の CPU 負荷プロセスを発生させる。

```
stress -c m
```

2. `top` コマンドを実行し、`stress` コマンドによって生成された負荷プロセスの CPU 使用率 (CPU usage) を計測し、外部ファイルに記録する。計測間隔は 5 秒とする。

```
top -b -d1 | grep <PID(s) of stress> > output
```

すなわち、`top` コマンドによって記録される負荷プロセスの CPU 使用率は、負荷プロセスが 5 秒間の総 CPU 時間に対して占めた CPU 時間の割合となる。

3. 各 EPC ノード、及び eNodeB+UE マシン上で `tcpdump` を起動し、パケットキャプチャを開始する。
4. eNodeB+UE マシン上で OAISIM を起動し、ベアラ確立までのシグナリング処理を発生させる。

以上の実験によって得られたデータを用いて、以下のように定義される評価指標を算出する。なお、本報告においては、UE 及び eNodeB はシミュレータによって実現されているため、UE 及び eNodeB で実行されるシグナリング処理に関する評価は行わない。

シグナリング処理時間: 各 EPC ノードにおける、個々のシグナリング処理にかかる時間。

各ノードにおけるパケットキャプチャ結果を用いて、それぞれのシグナリングメッセージが到着してから、次のシグナリングメッセージが送出されるまでの時間を算出することによって得る。

総シグナリング処理時間: 各 EPC ノードにおける、シグナリング処理にかかる時間の総和。前項のシグナリング処理時間を総和することによって得る。

ベアラ確立時間: UE がベアラを確立するためにかかった時間から, UE 及び eNodeB における時間を減じたもの. 図 2 における $t_a + t_b + t_c + t_d$ と定義する. eNodeB+UE マシン, 及び MME におけるパケットキャプチャ結果を用いて算出する. t_a, t_b, t_c 及び t_d は, それぞれ以下のように説明される.

t_a : eNodeB が Initial UE Message を送出してから, NAS Authentication Request を受け取るまでの時間.

t_b : eNodeB が NAS Authentication Response を送出してから, NAS Security Mode を受け取るまでの時間.

t_c : eNodeB が NAS Security Mode Complete を送出してから, Initial Context Setup Request を受け取るまでの時間.

t_d : MME が Initial Context Setup Response を受け取ってから, Modify Bearer Response を受け取るまでの時間. 本来は, eNodeB が Initial Context Setup Response を送出してから, MME が Modify Bearer Response を受け取るまでの時間とするべきであるが, ノード間の時刻同期精度が計測結果に与える影響を考慮し, MME におけるパケットキャプチャ結果のみを利用して算出する. 具体的には, eNodeB と MME 間の伝搬遅延を, Initial Context Setup Response に限って無視している.

実験は, 与える負荷の設定値に対してそれぞれ 600 回行う. 得られた実験結果から上述の評価指標を算出し, その平均値と分布を示すことにより, 実験結果に対する考察を行う.

4 実験結果と考察

本章では、3章で述べた実験方法に従って行った実験から得られたデータを示し、考察を行う。

4.1 ベアラ確立時間

図4に、ノードに人工的に与えたCPU負荷量に対する、ベアラ確立時間の推移を示す。この図より、CPU負荷が大きくなるにつれ、ベアラ確立時間が大きくなっていることが分かる。また、MMEのみに負荷を与えた場合よりも、SP-GWのみに負荷を与えた場合のほうが、ベアラ確立時間の増加量が大きいことがわかる。この結果は、図2に示す、各ノードにおけるシグナリングメッセージの処理回数から判断すると直感的ではない。すなわち、シグナリングメッセージの処理回数はMMEの方が多いにも関わらず、ベアラ確立時間はMMEにCPU負荷を与える場合の方が小さい。また、後述するシグナリング処理時間と比較すると、ベアラ確立時間が非常に大きくなっている。

実験結果を精査した結果、図4に見られるベアラ確立時間の増加は、ノードにおけるシグナリング処理時間の増加が原因ではなく、ノード間のシグナリングメッセージの伝搬遅延時間の増大が原因であることが分かった。これは、EPCノード及びそれらを接続するネットワーク環境が、全て仮想化環境内に構築されていることに起因すると考えられる。すなわち、ノードにCPU負荷を与えることにより、仮想化環境全体の負荷が増大することで、ノード間のパケット送受信処理時間が増大したものである。この結果は、本実験において用いた仮想化環境内における、ネットワーク処理のための資源と、仮想マシン処理のための資源の分離性の欠如に起因する問題であると考えられるため、本報告における実験の目的である、シグナリング処理時間に関する結果ではない。しかし、近年、EPCノード群をまとめて1つの仮想化環境を用いて提供するvirtualized EPC (vEPC) [16]が注目されていることを鑑みると、無視できない結果であると考えられる。

4.2 MMEにCPU負荷を与える場合

図5に、MMEにCPU負荷を与えた場合において、CPU負荷に対する各ノードにおける各シグナリング処理時間の変化を示す。この図における実線はMMEにおけるシグナリング処理、破線はSP-GWにおけるシグナリング処理、一点鎖線はHSSにおける処理を表している。また、図6に、総シグナリング処理時間に関する結果を示す。図6(a)には、総シグナリング処理時間の平均値をEPCノード別に分類して示している。また図6(b)には、総シグナリング処理時間の累積補分布を示している。図5より、CPU負荷が高くなるにつ

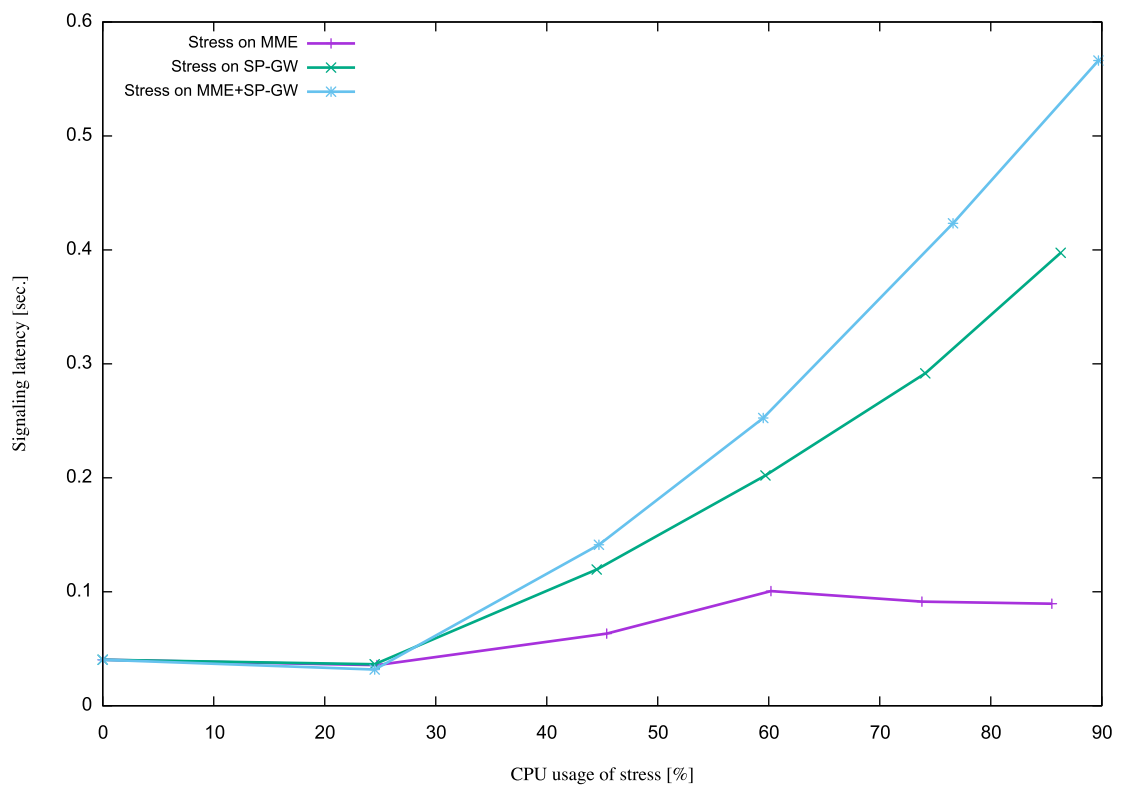


図 4: ベアラ確立時間の変化

れて、600回の実験結果の平均値であるにも関わらず、シグナリング処理時間が不規則に変動していることがわかる。このことは、図 6(a) に示すように、MME における総シグナリング処理時間の平均値が CPU 負荷に対して不規則に変動していることから読み取れる。これは、CPU 負荷が高くなることによって、小さい確率ではあるが、大きな処理時間が発生する場合があることを示唆している。図 6(b) から、CPU 負荷が 40% 以上の場合に、およそ 1% の確率で、総シグナリング時間が 10 倍から 100 倍に増大し、およそ 1 秒に達する可能性があることがわかる。また、CPU 負荷が高くなるにつれ、総シグナリング処理時間の増大幅が大きいことも分かる。このような総シグナリング処理時間の増大は、実験回数に関わらず、図 6(b) に示したものと同様の確率で発生することも分かった。このような総シグナリング処理時間の増大は、特に送受信データが小さい M2M/IoT の場合や、データ転送時間に厳しい制約があるようなアプリケーションの場合には無視できないと考えられる。

一方、図 6(b) により、上述のような大きな総シグナリング処理時間が発生しない場合においては、CPU 負荷が大きくなるにつれて、総シグナリング処理時間が小さい場合が多くなることが分かる。これは、仮想化ホスト上の物理 CPU、あるいは仮想マシン上の仮想 CPU 能力の動的なスケールリング機構によるものであると推測される。すなわち、CPU に人工的な負荷を与えることによって、CPU 能力そのものが高くなることで、シグナリング処理時間が短くなったと考えられる。

図 7 および図 8 に、`stress` コマンドによって負荷プロセス数を変化させた場合における実験結果を示している。これは、図 5 及び図 6 の場合よりも、より大きな CPU 負荷を与えていることを意味する。図 7 に、負荷プロセス数とシグナリング処理時間の関係を、また図 8(a) に、総シグナリング処理時間の変化を示す。これらの図を図 5 および図 6(a) と比較すると、より大きな負荷を与えることによって、シグナリング処理時間の不規則な変動が、より大きくなっていることが分かる。また、図 8(b) に示す、総シグナリング処理時間の累積補分布を、図 6(b) と比較することにより、CPU により大きな負荷を与えることによって、総シグナリング処理時間が非常に大きくなる確率がより高くなり、またその時の総シグナリング処理時間がより大きくなることがわかる。

4.3 SP-GW に CPU 負荷を与える場合

図 9 及び図 10 に、SP-GW に CPU 負荷を与えた場合の実験結果を示す。これらの図から、MME に CPU 負荷を与えた場合と同様に、CPU 負荷が高くなるにつれ、SP-GW におけるシグナリング処理時間が非常に大きくなる場合が存在することが分かる。

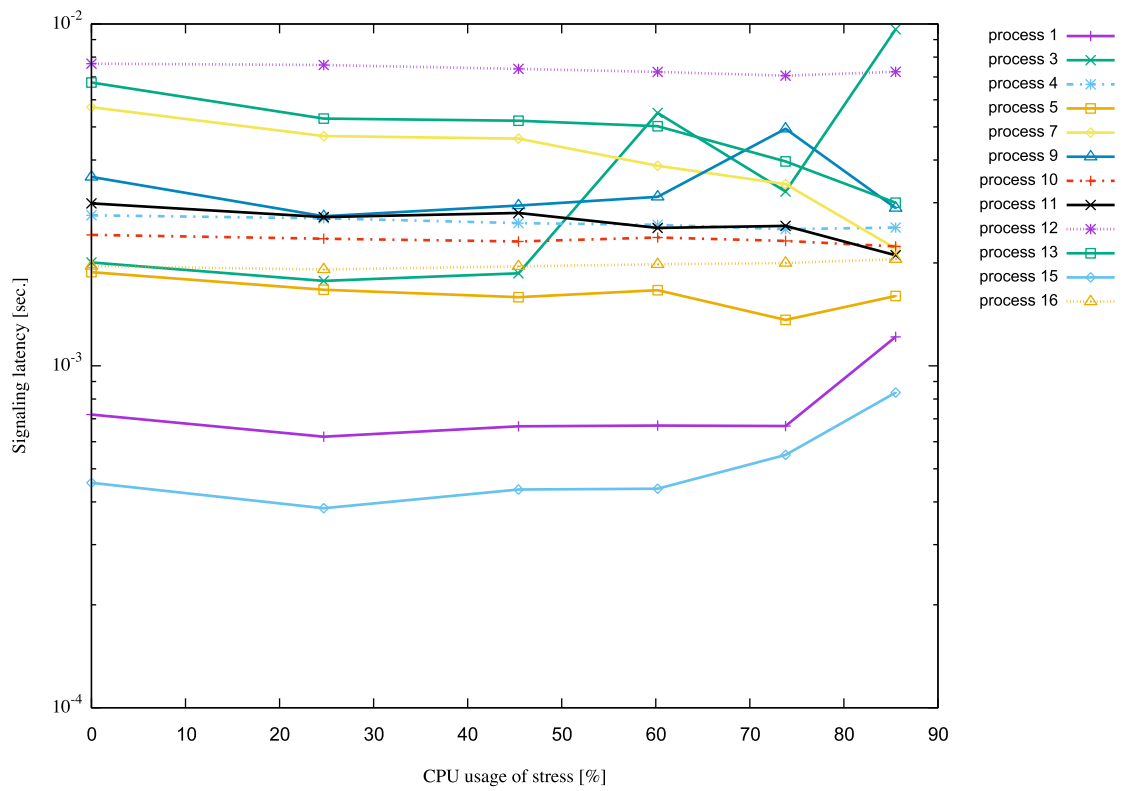
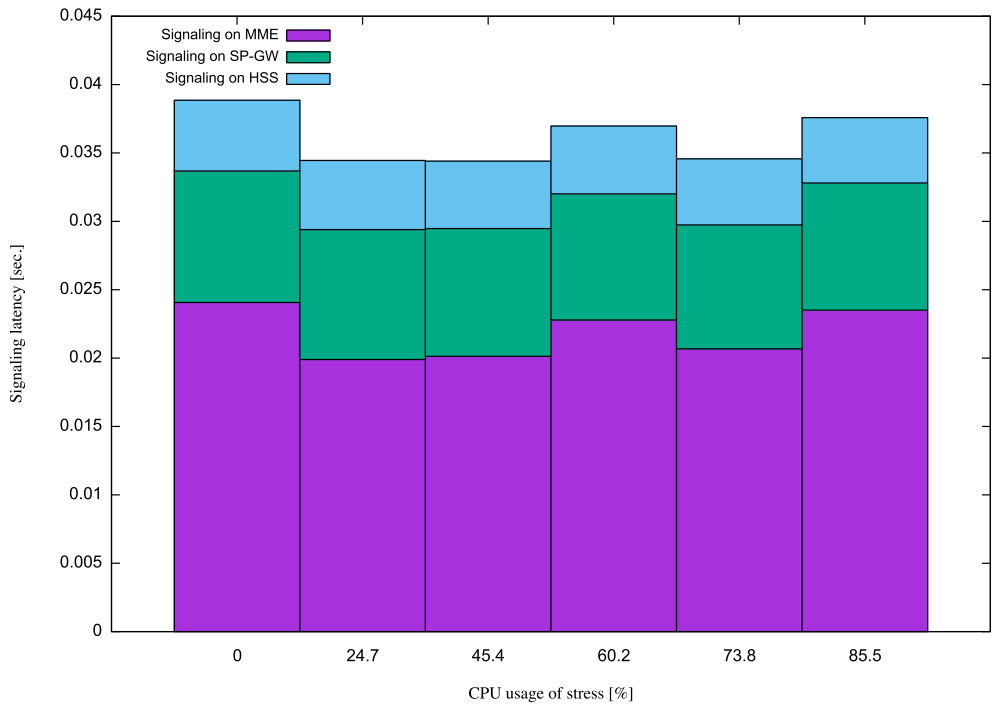
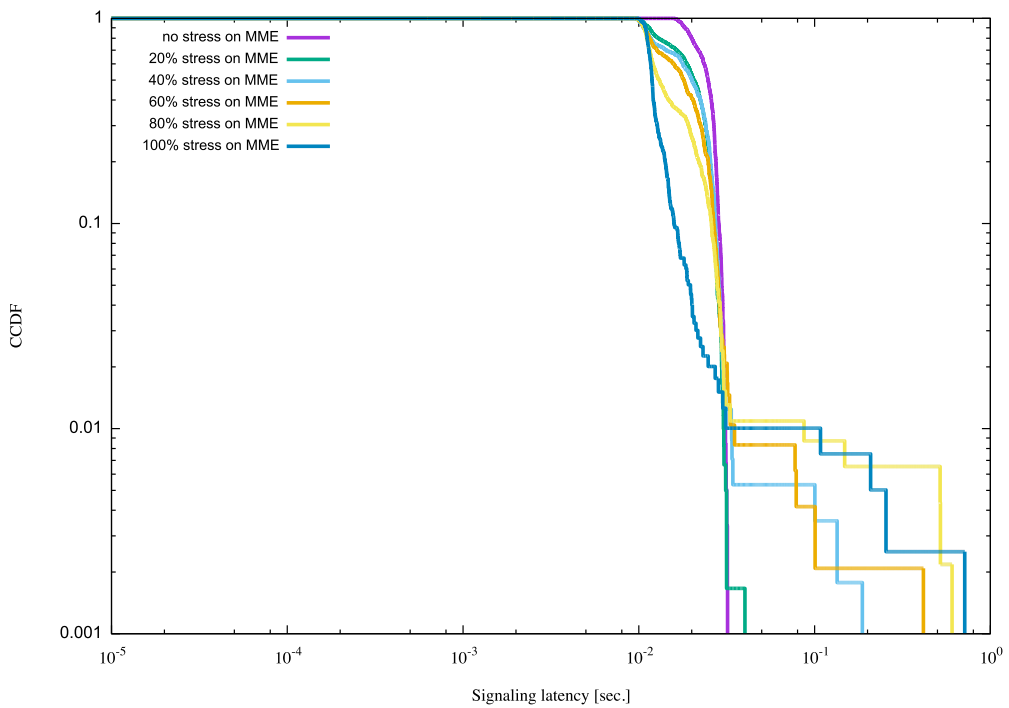


図 5: MME のみに負荷を与えた場合のシグナリング処理時間



(a) CPU 負荷に対する平均値の変化



(b) CCDF

図 6: MME のみに負荷を与えた場合の総シグナリング処理時間

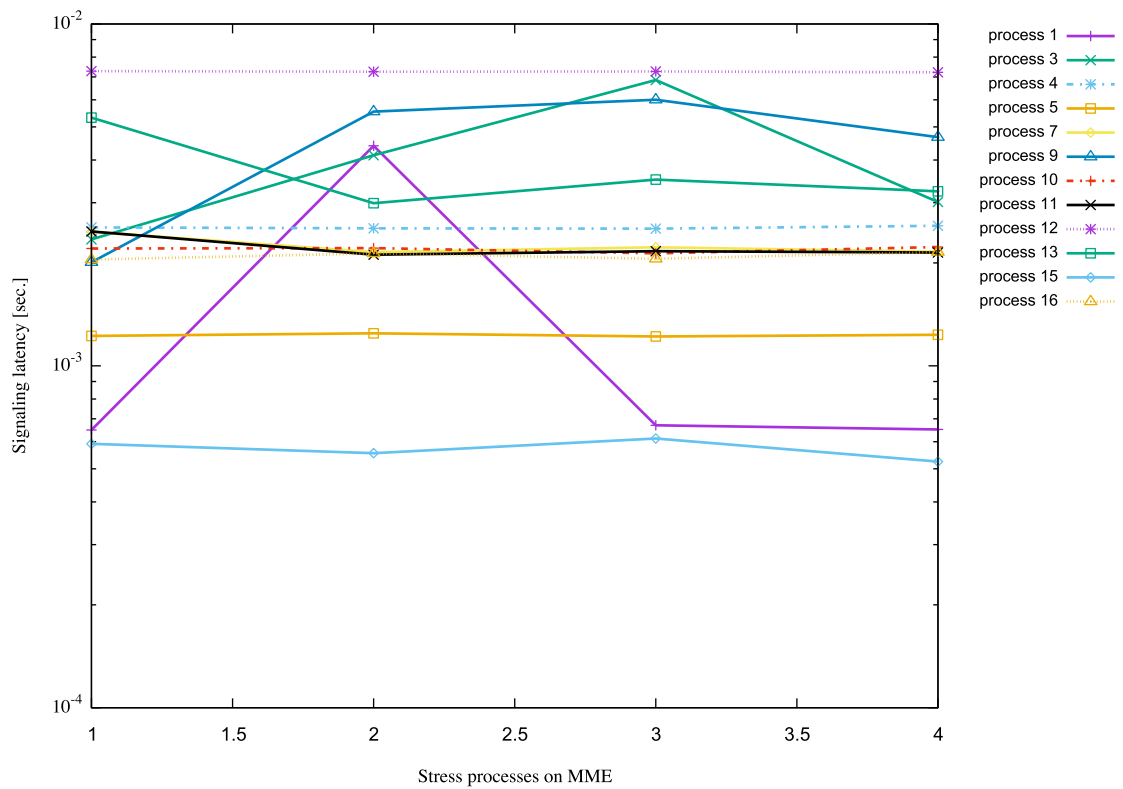
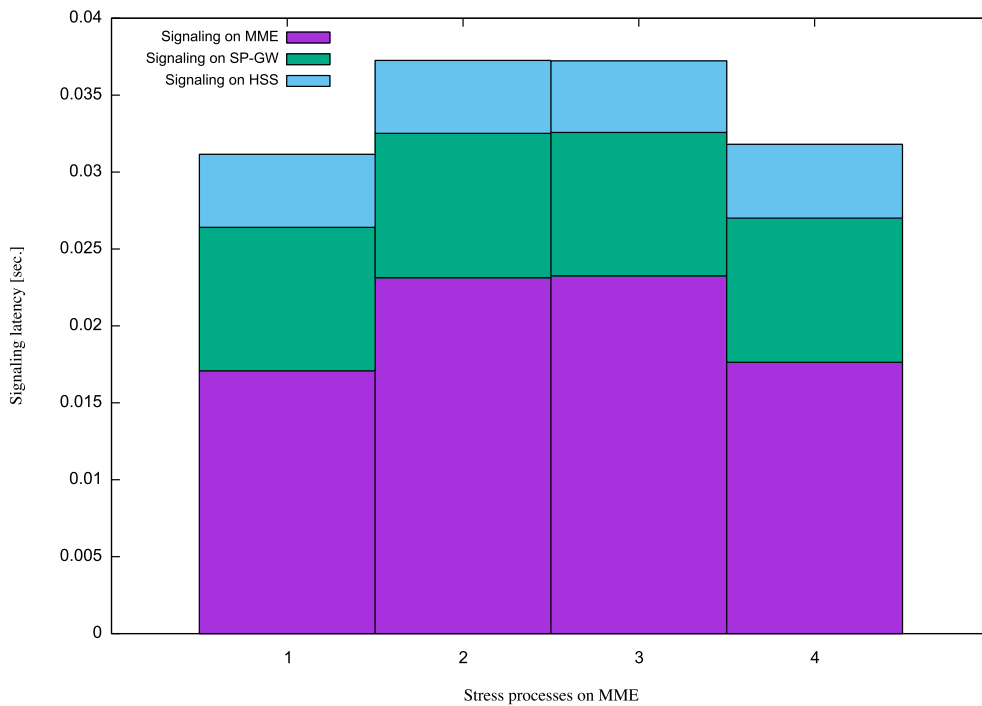
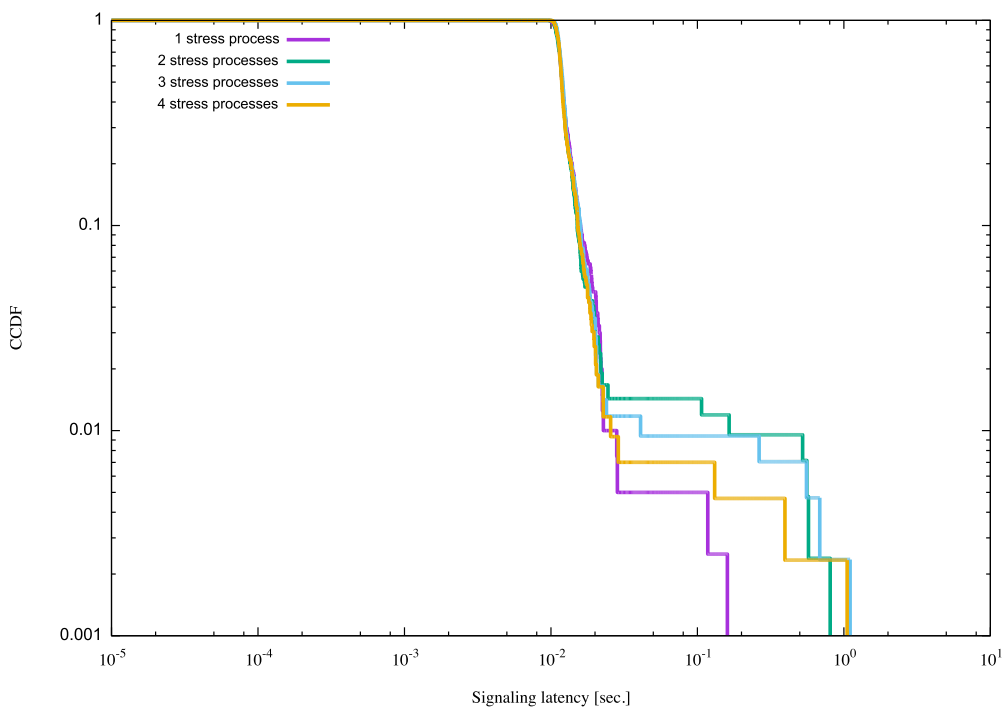


図 7: 負荷プロセス数を用いた場合のシグナリング処理時間



(a) 総シグナリング処理時間



(b) 累積補分布

図 8: 負荷プロセス数を用いた場合の総シグナリング処理時間

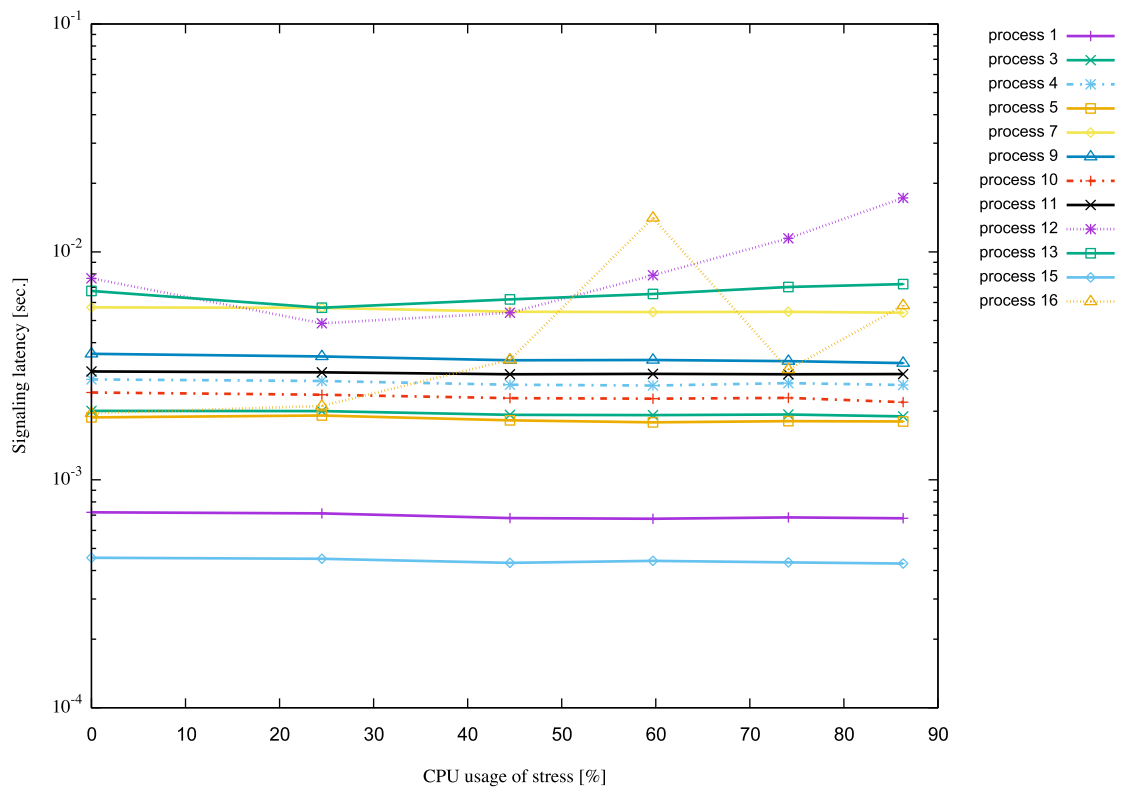
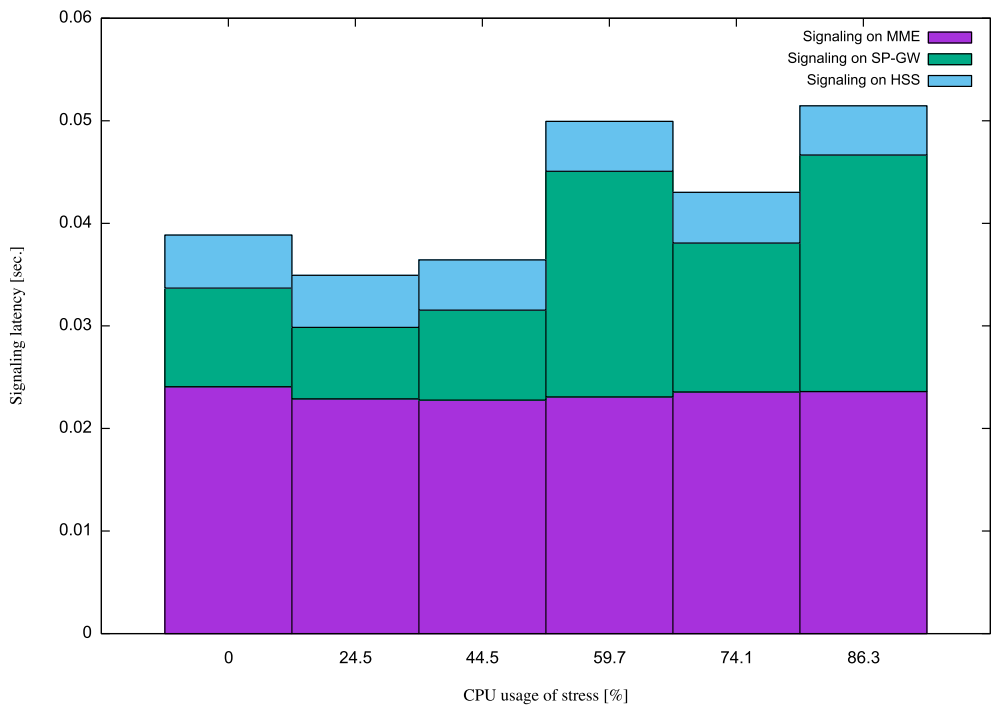
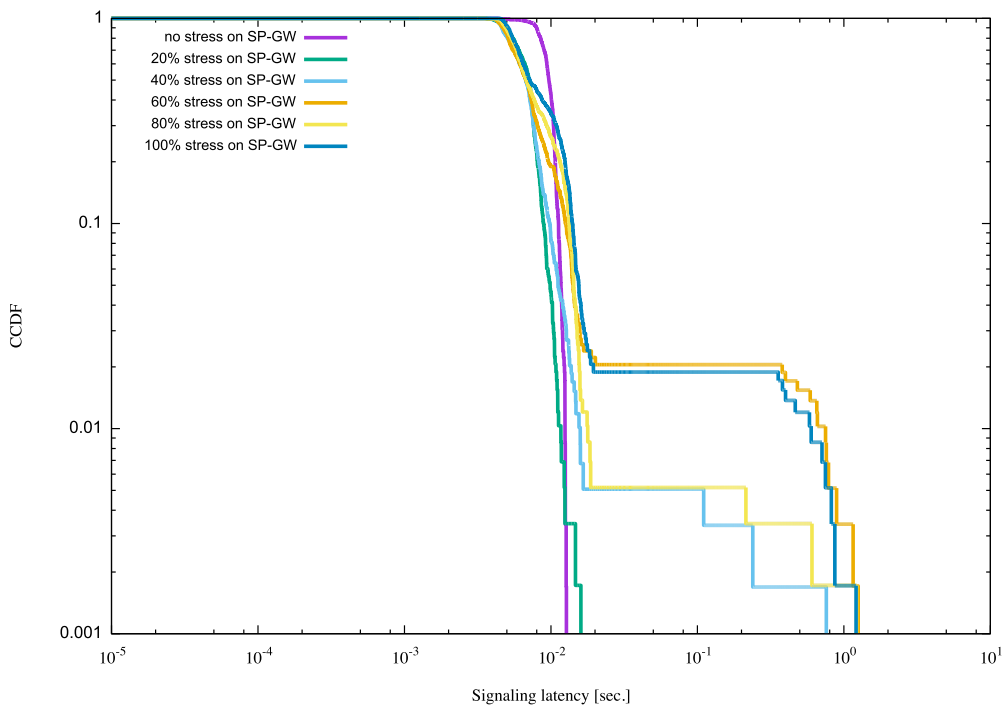


図 9: SP-GW のみに負荷を与えた場合のシグナリング処理時間



(a) CPU 負荷に対する平均値の変化



(b) 累積補分布

図 10: SP-GW のみに負荷を与えた場合の総シグナリング処理時間

4.4 MME と SP-GW の両方に CPU 負荷を与える場合

図 11 及び図 12 に、MME と SP-GW の両方に CPU 負荷を与えた場合の実験結果を示す。これらの結果からも、CPU 負荷が増大するにつれて、MME と SP-GW におけるシグナリング処理時間の平均値の変動が大きくなること、また、その原因は、およそ 1% の確率で発生する、シグナリング処理時間の増大が原因であることが分かる。

4.5 考察

上述した通り、各 EPC ノードにおける総シグナリング処理時間は、高い CPU 負荷を与えた場合に、小さいが無視できない確率で、大きな増加を見せることがあり、平均値を評価した場合においても、不安定になる傾向があることが明らかになった。このことから、多数の UE がモバイルネットワークに接続するような環境において、シグナリング処理を行うノードの CPU 負荷が高まることによって、一部の UE が通信を開始するまでに、極めて大きな時間を要することが懸念される。また、4.1 節における議論から、特に仮想化環境を用いて EPC を実現する場合には、ノードの負荷がネットワーク処理にも影響を与える可能性が示唆された。さらに、仮想化ホストあるいは仮想マシンの CPU 能力の自動制御が、シグナリング処理時間に影響を与える可能性も明らかになった。

これらのことから、セルラネットワーク内に多数の M2M/IoT 端末を収容するためには、モバイルコアネットワークを構成するノードの資源、及びネットワーク処理のために必要な資源を適切に配分するとともに、それらの間の資源制御を慎重に行う必要があると考えられる。また、ノードの C/U 分離を行い、コントロールプレーンをクラウド環境に集約する場合などを想定すると、仮想化環境に起因する種々の問題に適切に対応する必要があると考えられる。

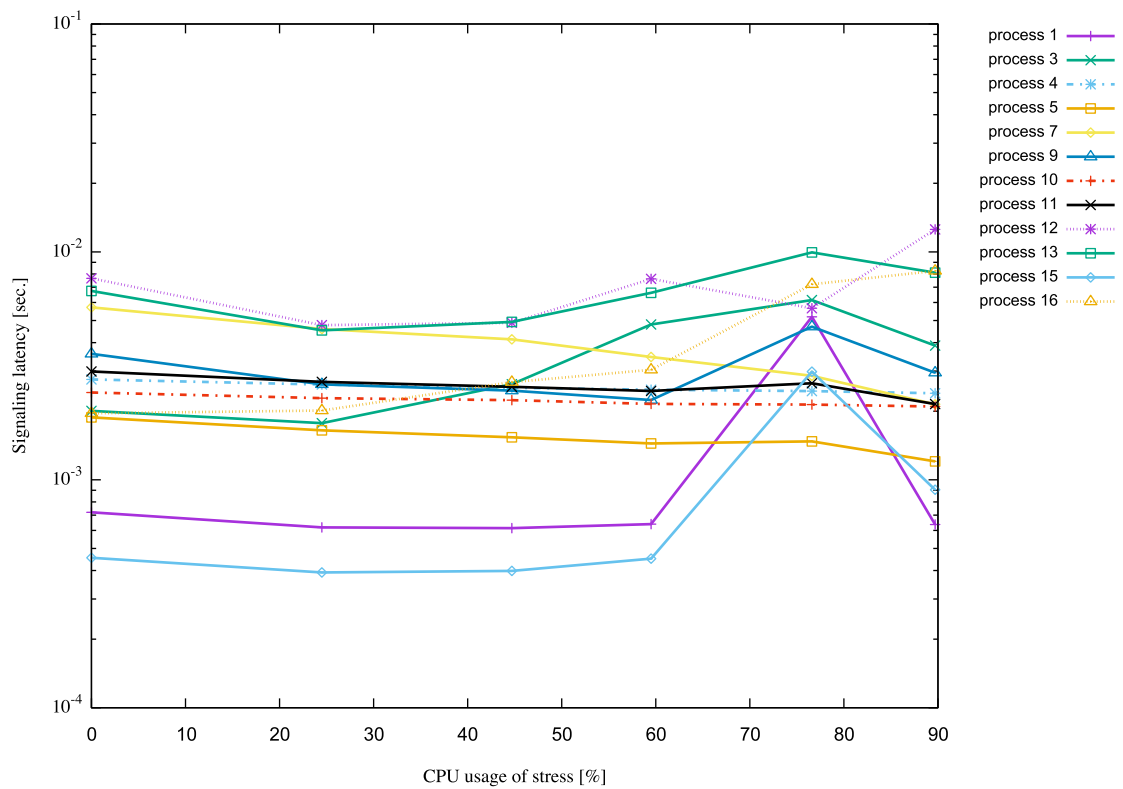
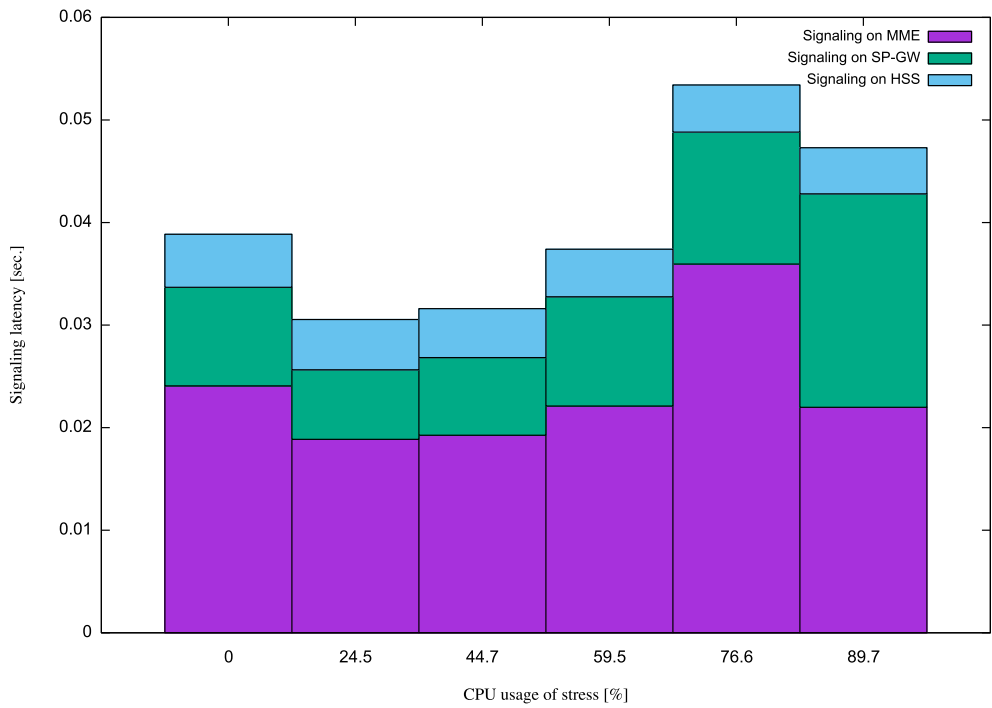
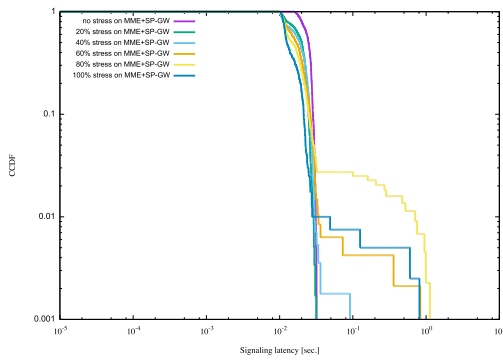


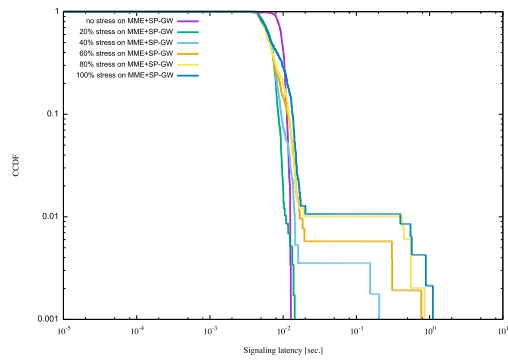
図 11: MME と SP-GW に負荷を与えた場合のシグナリング処理時間



(a) CPU 負荷に対する平均値の変化



(b) 累積補分布 (MME)



(c) 累積補分布 (SP-GW)

図 12: MME と SP-GW に負荷を与えた場合の総シグナリング処理時間

5 まとめと今後の課題

本報告では、大量の M2M/IoT 通信をセルラ無線に収容するためのモバイルコアネットワークアーキテクチャについての知見を得るために、モバイルコアネットワーク環境を実ソフトウェアを用いて実現し、その上で負荷実験を実施することにより、UE がネットワークに接続する際の遅延時間に関する評価を行った。その結果、EPC ノードに対して高い CPU 負荷を与えた場合には、小さいが無視できない確率で、シグナリング処理遅延時間が非常に大きくなる場合があることが明らかとなった。

今後の課題としては、複数の UE がモバイルコアネットワークに接続するようなシナリオを用いた実験を行うことで、本報告で用いたような人工的な負荷ではない、実環境において発生するような負荷を与える場合の評価を行いたい。また、構築した実験環境を用いて、EPC ノードの C/U 分離、クラウド環境を用いたノード間の資源配分、及びシグナリング軽量化手法などの実験的評価を行いたい。さらに、データ送信部分も含めたモバイルコアネットワークの性能評価なども重要である。

謝辞

本研究を進めるにあたり、多くの先生方、先輩方のご指導をいただきました。指導教員である松岡茂登教授には、研究者としての姿勢を教えていただき、時には研究内容に対する鋭いご指摘もいただきました。これらのご指摘により、私に足りなかった研究に対する視点や、関連技術・研究に関しての知見を得ることができました。ありがとうございます。村田正幸教授には、毎回のミーティングで私の研究内容に対しての様々なアドバイスやご指摘をいただきました。私の進める研究の位置づけや意義について何度も考える機会を頂き、誠に感謝しております。日々の研究、報告、そして本報告の作成は、長谷川剛准教授の丁寧なご指導無くしては成し遂げられませんでした。私の至らない点、知識不足で何度もつまづきそうになった時、時には厳しく、時には優しく、幾度も手助けを頂いたことを心から感謝しております。樽谷優弥助教には、日々の研究、および研究室での生活に至るまで、様々なアドバイスをいただきました。日頃の温かいご指導に、大変感謝しております。

松岡研究室の先輩方には、日頃の研究に対する御支援、御助言を頂き、更には研究室生活の様々な場面で手助けして頂きました。特に、同じ研究チームとして、日頃から研究に対する様々なアドバイスを頂き、私が悩んだことの解決を手助けして下さった阿部修也先輩には、感謝してもしきれません。本当にありがとうございます。最後に、同じ研究室のメンバーとして配属された同期の方々、日頃から良き仲間として楽しい時間を過ごすことができました。本当に、ありがとうございます。

参考文献

- [1] D. Astely, E. Dahlman, A. Furuskär, Y. Jading, M. Lindström, and S. Parkvall, “LTE: the evolution of mobile broadband,” *IEEE Communications Magazine*, vol. 47, no. 4, pp. 44–51, April 2009.
- [2] P. Marsch, I. D. Silva, O. Bulakci, M. Tesanovic, S. E. E. Ayoubi, T. Rosowski, A. Kaloxylos, and M. Boldi, “5G Radio Access Network Architecture: Design Guidelines and Key Considerations,” *IEEE Communications Magazine*, vol. 54, no. 11, pp. 24–32, November 2016.
- [3] F. Ghavimi and H. H. Chen, “M2M Communications in 3GPP LTE/LTE-A Networks: Architectures, Service Requirements, Challenges, and Applications,” *IEEE Communications Surveys Tutorials*, vol. 17, no. 2, pp. 525–549, Secondquarter 2015.
- [4] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications,” *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2347–2376, Fourthquarter 2015.
- [5] U. Raza, P. Kulkarni, and M. Sooriyabandara, “Low Power Wide Area Networks: An Overview,” *IEEE Communications Surveys Tutorials*, vol. PP, no. 99, pp. 1–1, January 2017.
- [6] L. Alliance, “LoRaWAN-what is it?,” A technical overview of LoRa and LoRaWAN, 2015.
- [7] “ソフトバンク株式会社プレスリリース: 低消費電力での IoT を実現する「LoRaWAN™」提供開始.” available at http://www.softbank.jp/corp/group/sbm/news/press/2016/20160912_01/.
- [8] “NTTdocomo 報道発表資料: 様々な IoT サービスに利用可能な LPWA 対応 IoT ゲートウェイ機器の実証実験を開始.” available at https://www.nttdocomo.co.jp/info/news_release/2016/11/15_00.html.
- [9] “Sigfox - The Global Communications Service Provider for the Internet of Things (IoT).” available at <https://www.sigfox.com>.

- [10] “京セラコミュニケーションシステム株式会社 ニュースリリース: IoT ネットワーク「SIGFOX」を日本で展開し、LPWA ネットワーク事業へ参入.” available at <http://www.kccs.co.jp/release/2016/1109/>.
- [11] *3GPP TS23.401 V13.6.1 “General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access*, March 2016.
- [12] 阿部 修也, 長谷川 剛, 村田 正幸, “シグナリング処理負荷を考慮したノード仮想化及びプレーン分離を適用したモバイルコアネットワークの性能評価,” 電子情報通信学会技術研究報告, vol. 115, no. 507, pp. 41–46, 2016 年 3 月.
- [13] 阿部 修也, 長谷川 剛, 村田 正幸, “C/U 分離を適用したモバイルコアネットワークにおける通信集約方式の性能評価 (ネットワークシステム),” 電子情報通信学会技術研究報告 = IEICE technical report : 信学技報, vol. 116, no. 382, pp. 25–30, December 2016.
- [14] “OpenAirInterface.” available at <http://www.openairinterface.org>.
- [15] V. S. Rao and R. Gajula, “Protocol signaling procedures in LTE,” White Paper, Radisys Corporation, 2011.
- [16] H. Hawilo, A. Shami, M. Mirahmadi, and R. Asal, “NFV: state of the art, challenges, and implementation in next generation mobile networks (vEPC),” *IEEE Network*, vol. 28, no. 6, pp. 18–26, November 2014.