ミーティング資料

安達智哉
to-adachi@ist.osaka-u.ac.jp

2019 年 4 月 10 日

1 Server Disaggregation

1.1 Server Disaggregation に関する先行研究

Intel Corporation の White Paper [1] では、Disaggregated Server に基づくインテル®ラックスケールアーキテクチャを示している。このアーキテクチャでは、マザーボード上で CPU / DRAM モジュールと NIC /ドライブモジュールを分離している。この手法より、CPU とメモリを他のコンポーネントを交換することなく、交換することが可能であり、コンピューティングおよびストレージのパフォーマンスや容量を手頃な価格で向上させることができる。交換作業は簡単であり、従来のように何時間もかかることはない。4 本のネジを外して CPU / DRAM モジュールをスライドさせ、新しい CPU / DRAM モジュールを取り付けることで交換作業は完了すると述べている。サーバの構成を図 1 に示す。また、交換作業の工程と所要時間を図 2 に示す。

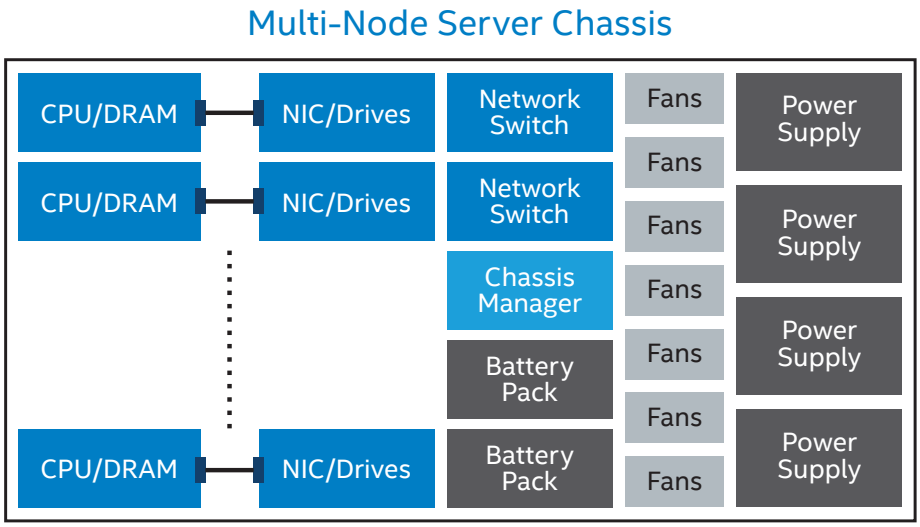


Figure 1. The disaggregated server architecture is characterized by a CPU/DRAM module and a NIC/Drives module that can be refreshed independently of each other and of the rest of the server components.

図 1: サーバ構成

Table 1. Faster Refresh Is Now Possible

Old Method	New Method
Six different technician skills, five handoffs: • Data center manager • Physical rack and stack technician • Network cabling technician • Network configuration engineer • Server/OS configuration engineer • Batch clustering administrator (for new system name configuration)	Two technician skills, one handoff: • Board replacement technician • Server/OS configuration engineer
35 hours of work time per rack[1]	**8 hours of work time per rack[1]**

[1] Based on internal testing.

図 2: コンポーネントの交換工程

また、文献 [2]、[3] では、CPU とメモリの分離も前提とした Disaggregated Server アーキテクチャについて述べている。課題は、CPU とメモリ間のネットワークの構成である。一つは、新しいメモリアーキテクチャを必要とすることである。CPU とメモリを分離することにより、両者の物理的な距離が増大する。これに対応するためは、両者を結ぶネットワーク帯域幅の大きな飛躍を必要とする。現在 Intel では、光ファイバーケーブルの束を必要とし、CPU とメモリを接続することを想定している。しかし、それをサポートするためには全く新しいクラスのスイッチアーキテクチャを設計しなければならないと考えられる。Intel が想定する将来の分離型ラックは、以下の要素から構成されると述べている。

- 複数個のプロセッサトレイ
- 複数個のシステムメモリトレイ
- SSD もしくは HDD からなる多数のストレージトレイ
- 上記のトレイをすべて結び付けるファブリックネットワーク

文献 [4] では、サーバとストレージをラックスケールで分離し、制御の時間スケールが与える影響について調査している。この文献では、最も時間スケールの細かい制御として、IO 処理毎にストレージ構成を変化させるモデルを想定している。この場合、ストレージの再割り当てに伴うオーバーヘッドが大きくなると述べている。一方、複数の IO 処理を実行する間、ストレージ構成を変化させないモデルを想定した場合、IO ごとにストレージ構成を変化させるモデルと比較して、大幅にオーバヘッドが軽減されると述べている。また、ストレージ構成を変化させる頻度は、数日から数時間になると述べている。図 3 に制御のタイムスケールの細かさとその特徴をまとめた表を示す。また、図 4 に IO ごとにストレージ構成を変化させるモデルにおいて発生するオーバヘッドを示す。

Summary of Disaggregation Scenarios

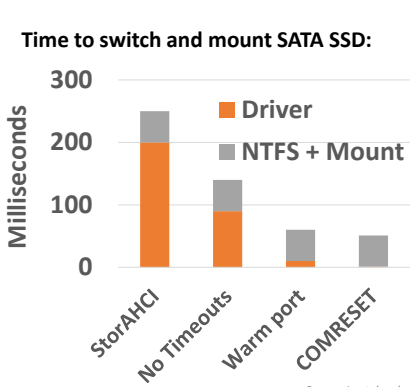
	Configuration Disaggregation	Failure disaggregation	Dynamic Elastic Disaggregation	Complete Disaggregation
Storage stack redesign	No	Small	Moderate	High
Online controller	No	Not necessarily	Yes	Yes (on IO path)
Reconfiguration frequency	O(rack lifetime)	O(server failures)	O(hours-days)	O(IO rate)
Reconfiguration overhead	None	Not under normal operation	Low	High

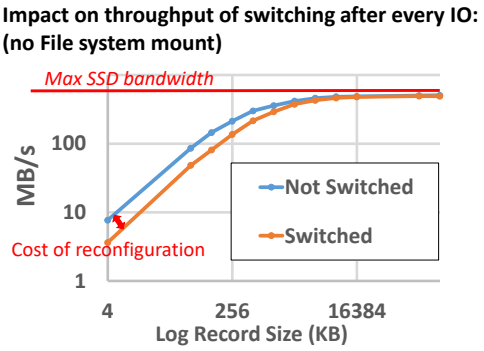
Sergey Legtchenko – Understanding Rack-Scale Disaggregated Storage

図 3: ストレージ構成の制御のタイムスケールの細かさとその特徴

Complete Disaggregation, seriously?

- Can we reconfigure per IO?





Sergey Legtchenko – Understanding Rack-Scale Disaggregated Storage

図 4: 細かいタイムスケールでのストレージ構成制御を行った場合のオーバーヘッド

文献 [5] では、リソースを分離した際の主な課題として、メモリ、SSD/HDD、GPU/FPGA などのリモートリソースプールにアクセスする際にインターコネクトとスイッチで発生する待ち時間を挙げている。その理由は、リモートリソースプールのアクセスレイテンシが本来のアクセスレイテンシと比較して大幅に増大すると、プロセッサ、ハイパーバイザ、OS、またはアプリケーションレベルでスレッドの並列処理が利用されない限り、パフォーマンスが大幅に低下する可能性があると述べている。また、リソース分離の規模として、ラック単位、PoD 単位、データセンタ単位の 3 種類を想定している。それぞれのモデルを図 5 及び図 6 に示す。

今日ローカルに接続されたメモリにアクセスするための帯域幅と待ち時間は、それぞれ 920 Gb / 秒と 75 ns である。PCIe スイッチ（第 3 世代）は 150 ns 程度の遅延を発生させ、トップオブラック IP スイッチおよび Infiniband スイッチの遅延は最大 800 ns である。そのためこの文献では、ラックを超えてメモリリソースを分離するためには、シリコンフォトニクスと光回路スイッチ（OCS）が唯一の選択肢であると述べている [6, 7, 8]。そして、そのようなネットワークの構成は、従来のものと比べてはるかに高いコストであると述べている [9]。

またこの文献では、以下に示すようなさまざまなインターコネクトおよびスイッチテクノロジのポート間レイテンシを仮定した上で、GPU/FPGA、SSD/HDD のリソースに関しては、低遅延の TOR スイッチ、PCIe スイッチ、または InfiniBand スイッチを用いることで簡単にリソース分離が可能であるとも結論付けている。

- Arista 製などの低レイテンシ TOR スイッチ（380〜1000ns）[10]
- Arista 製などの低レイテンシ spine-leaf スイッチ（2〜10μs）[10]
- Mellanox 製の InfiniBand スイッチ（700ns）
- Calient 製の光回線スイッチ（ $< 30ns$）[11]
- PCIe スイッチなど H3 プラットフォームで作成されたもの（〜150ns）
- ラック、PoD、およびデータセンターの往復伝搬遅延 $5ns/m$
- ラック内：平均伝搬距離 $3m$（$15ns$）
- Intra PoD：平均伝搬距離 $50m$($250ns$)
- データセンター内：平均伝搬距離 $200m$($1\mu s$)

上記の結果をまとめたものを図 7 に示す。

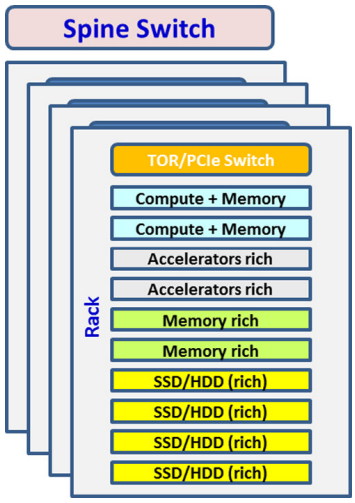


Fig. 3. In rack scale architecture, each of the nodes within the rack is specialized into being rich in one type of resources (computing rich, accelerator rich, memory rich, or storage rich).

図 5: ラックスケールでのリソース分離モデル

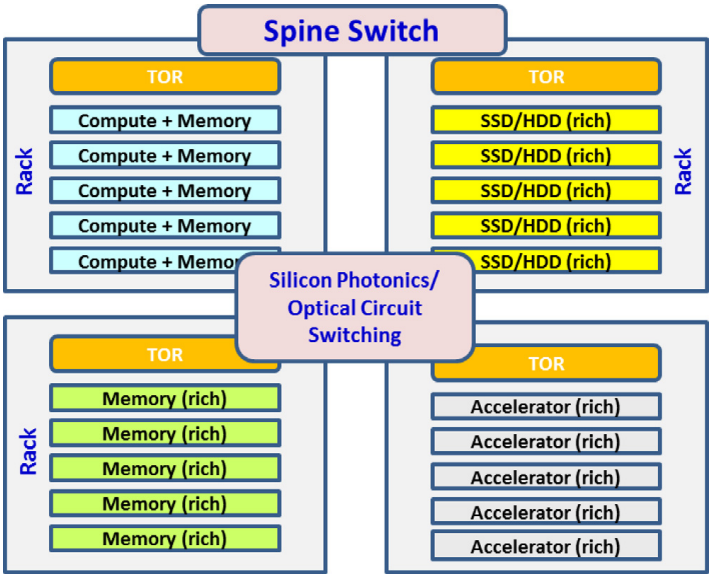


Fig. 4. Disaggregation architecture applied at the PoD or datacenter level.

図 6: PoD 及びデータセンタ規模でのリソース分離モデル

Table 1
Types of network for supporting composable systems at the rack, PoD and datacenter levels.

	Memory (DMI/DDR3)	GPU/FPGA (PCIe gen3)	SSD (SATA/SAS)	HDD (SATA/SAS)
Latency	∼75 ns	5–10 μs	∼100 μs	∼25 μs
Throughput	∼ 920 Gb/s	12 GB/s	∼100 K OPS	75–200 OPS
Rack (∼3 m, ∼15 ns)	Silicon photonics	TOR switch, PCIe switch, Infiniband switch	TOR switch, PCIe switch Infiniband switch	TOR switch, PCIe switch Infiniband switch
PoD (∼50 m, ∼250 ns)	Optical circuit switch (OCS)	Flat network (Spine–Leaf), OCS	Flat network (Spine–Leaf), OCS	Flat network (Spine–Leaf), OCS
Datacenter (∼200 m, ∼1 μs)	Optical circuit switch (OCS)	Flat network (Spine–Leaf), OCS	Flat network (Spine–Leaf), OCS	Flat network (Spine–Leaf), OCS

図 7: 各リモートリソースプールにアクセスする際の遅延制約とそれを満たすためのネットワーク構成

1.2 Server Disaggregation と私の研究との関係

現在のモバイルネットワークでは、EPC ノードにおける CPU とメモリのリソース消費の予測が難しような状況や変動が大きいような状況においても、どちらかがボトルネックにならずに、効率的にリソースを活用するアーキテクチャを考えることは重要である。実際、CPU とメモリのリソースを効率よく活用する研究は、データセンタなどの分野では行われている。文献 [12] では、server disaggregation の考えをデータセンタに適用し、CPU やメモリなどのリソースをモジュール化し、需要に合わせて自由に組み替えることを可能にすることにより、リソースの効率的な利用が可能であることを示している。しかし、server disaggregation にはいくつかの課題がある。まず、CPU とメモリのリソースを分離するためには、大きなコストがかかる点が挙げられる。文献 [2]、[3] では、CPU とメモリを分離するためには、両者を結ぶための新しい高帯域ネットワークが必要になると述べている。また、両者の物理的な距離が増加することによって発生する遅延も考慮する必要があるため、新たなメモリアーキテクチャの構築が必要であると述べている。文献 [9] では、メモリを分離するためには、低遅延かつ高帯域のネットワーク接続が必要となるが、それを実装するためにコストは従来と比較して大幅に増加すると述べている。実際、文献 [1] では、Disaggregated Server に基づくインテルのラックスケールアーキテクチャを示しているが、このモデルでも CPU とメモリの分離はできていない。課題の 2 つ目として、短いタイムスケールでの制御が難しいという問題が挙げられる。例えば、ストレージをモジュール化してサーバと分離する手法について述べられている文献 [4] では、時間スケールの細かいストレージ制御を行った場合、ストレージの再割り当て処理に伴うオーバーヘッドが大きくなると述べている。また、頻繁にリソース構成を変化させることは、コスト面や消費電力の面でも不利である (注:まだ根拠はない)。

このように、server disaggregation を用いたリソース制御では、リソース制御に伴うオーバーヘッドの発生が避けては通れない課題となると予想される。特に、モバイルネットワークのように、突発的なトラヒックの増加が発生し、数分以下のオーダでリソース量の制御を行う必要があるネットワークにおいては細かいタイムスケールでの効率的なリソース制御が求められる。

そこで、本研究ではモバイルネットワークに特化した、柔軟かつ効率的な、EPC ノードにおける CPU とメモリ間の負荷のオフロード方法を考案する。具体的には、ネットワークの負荷に合わせて、UE の状態を制御することにより、メモリおよび CPU に与える負荷のバランスを変化させる。UE の状態の制御は、UE が最後にデータを送信したあと、Connected Inactive 状態から Idle 状態に遷移するまでの時間を設定することで実現する。この方法により、CPU が過負荷である場合は、UE が最後にデータを送信したあと、Connected Inactive 状態から Idle 状態に遷移するまでの時間を長く設定することにより、メモリの負荷を増加させる代わりに CPU の負荷を削減することが可能である。またその逆に、メモリが過負荷である場合は、この時間を短く設定することにより、CPU の負荷を増加させる代わりにメモリの負荷を削減できる。この時間の再設定処理は、数分単位のオーダーで可能でありかつ、それに伴い発生するオーバーヘッドは、server disaggregation と比較して僅かである (注:提案手法のオーバーヘッドの大小に関する記述は、今後の研究結果によって修正します)。また、既存のシグナリングアーキテクチャに変更を加えることが可能であれば、秒単位の時間スケールでの制御も可能であると考えられる。

しかし、提案手法には限界もある。それは、対応可能なリソース需要に制限があることである。なぜなら、提案手法では、限られた CPU およびメモリのリソースを効率的に利用することは可能であるが、双方のリソースが過負荷になるようなリソース需要には対応できないからである。特に長期的かつ大規模なリソース需要の変化に対しては、server disaggregation を用いたリソース制御の方が適している。また、提案手法を用いなかった場合には常時 Connected Inactive 状態であっ

た UE が、提案手法を用いることにより Idle 状態へと遷移する可能性があるため、データを送受信にかかる遅延時間が増加するなど、QoS の低下が発生する可能性がある。しかし、電気メータや気温計のようなリアルタイム性を必要としない IoT 端末であれば、これらの QoS の低下は無視できると考えられる。

そのため、提案手法と server disaggregation やスケールアウト/スケールインを組み合わせることにより、より効果のあるリソース管理が可能であると考えられる。例えば、長期的かつ大規模なリソース制御は server disaggregation を用いて行う一方で、server disaggregation で対応できないような短いタイムスケールでのリソース制御は提案手法を用いて行う。もしくは、リソース需要の大きな変動に対してはスケールアウト/スケールインを用いて対応した上で、細かなリソース需要の変化に対しては提案手法を用いて対応する。上述ように、server disaggregation やスケールアウト/スケールインと提案手法組み合わせ、双方のデメリットを補うことで、モバイルネットワークのリソース制御をより良く実現できると考えられる。

2 UE の状態遷移に伴う、シグナリングの発生数の調査

2.1 Idle 状態から Connected 状態への遷移

文献 [13] に示されている、コネクション確立に伴うシグナリング図を図 8 に示す。この図を見ることによって、UE が Idle 状態から Connected 状態に遷移する際に各ノードで処理する (送受信する) シグナリングの数は以下の様に求めることができる。

- UE :9
- eNodeB :12
- MME :5
- SGW :2

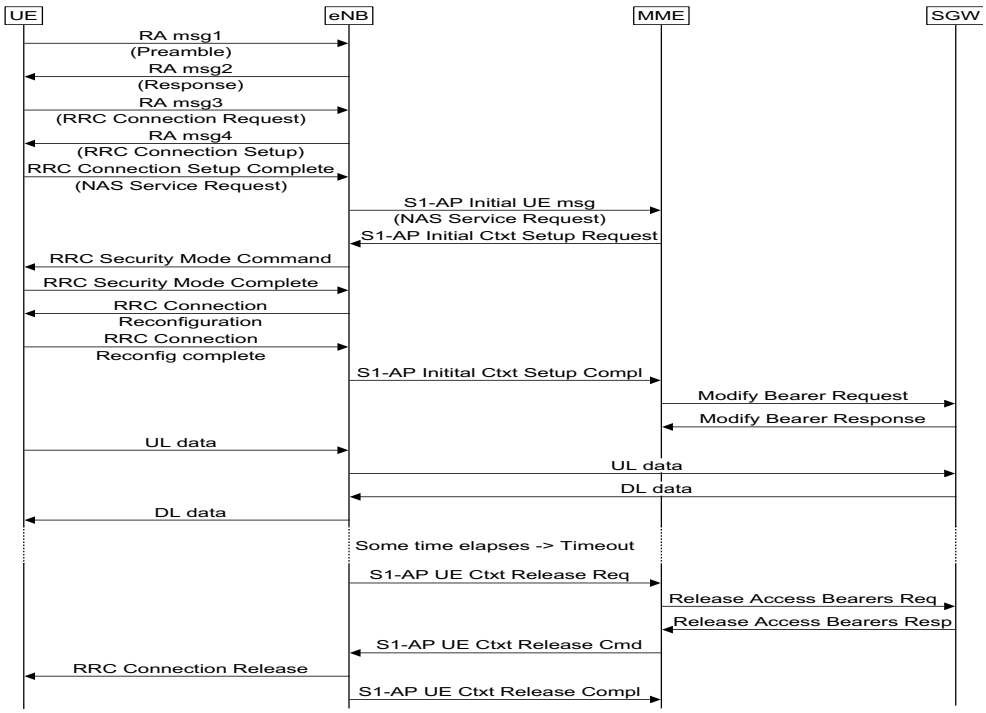


図 8: Legacy connection setup

2.2 Connected Inactive 状態から Connected 状態への遷移

文献 [14] では、RRC Connected Inactive 状態から Connected 状態へ遷移する際のシグナリング図を示していた。そのシグナリング図を図 9 に示す。図 9 を見ると、UE-RAN 間のシグナリングが 5 回発生していることがわかる。また、コアネットワーク側にはシグナリングは発生していないこともわかる。UE が Connected Inactive 状態から Connected 状態に遷移する際に各ノードで処理するシグナリングの数は以下の様になる。

- UE :5
- eNodeB :5
- MME :0
- SGW :0

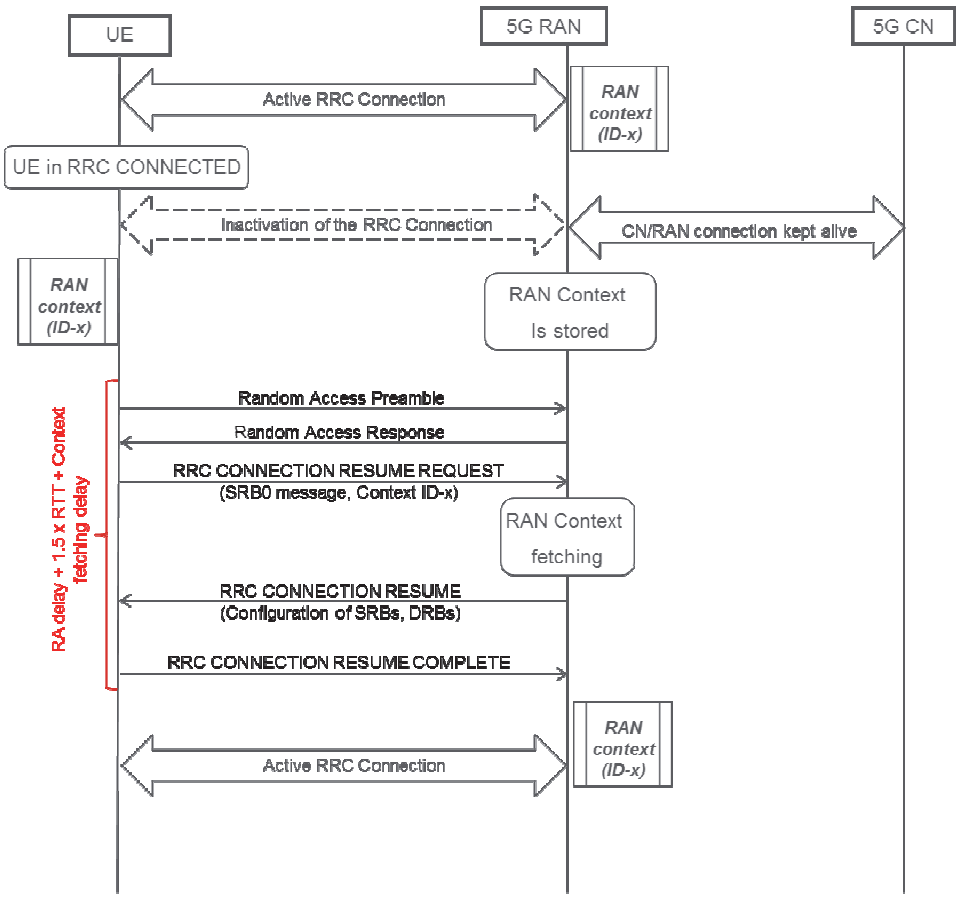


図 9: Signaling for the RRC CONNECTED INACTIVE to RRC CONNECTED transition for the novel state model

2.3 Connected 状態から Idle 状態への遷移

Connected 状態から Idle 状態へ遷移する際に発生するシグナリングに関しては、図 8 を見ることで確認できる。具体的には、図 8 の S1-AP UE Ctxt Relese Req 以降のシグナリングが、Connected 状態から Idle 状態へ遷移する際に発生するシグナリングに該当する。このシグナリングでは、eNodeB および MME の持っている UE のコンテキストおよび、MME と SGW のベアラ、RRC コネクションの解放を行っている。UE が Connected 状態から Idle 状態に遷移する際に各ノードで処理するシグナリングの数は以下の様になる。

- UE :1
- eNodeB :4
- MME :5
- SGW :2

2.4 Connected 状態から Connected Inactive 状態への遷移

Connected 状態から Connected Inactive 状態へ遷移する際に発生するシグナリングに関しては、図 8、図 9 および第 2.2 節、第 2.3 節を参考にすることにより推定可能である。

まず、図 9 および第 2.2 節より、Connected Inactive 状態から Connected 状態へ遷移する際には、UE-eNodeB 間において RRC コネクションを確立するためのシグナリングのみが発生し、コアノード側にはシグナリングが発生しないことがわかる。このことから、Connected 状態から Connected Inactive 状態への遷移する場合は、RRC コネクションを解放するためのシグナリングが発生する一方でコアノード側にはシグナリングは発生しない可能性が高いと予想できる。

RRC コネクションを解放する際に発生するシグナリングの数は、図 8 および第 2.3 節を見ることにより確認できる。

このことから、Connected 状態から Connected Inactive 状態への遷移するに各ノードで処理するシグナリングの数は以下の様になる。

- UE :1
- eNodeB :1
- MME :0
- SGW :0

2.5 Connected Inactive 状態から Idle 状態への遷移

Connected Inactive 状態から Idle 状態へ遷移する際に発生するシグナリングに関しては、第 2.3 節および第 2.4 節で求めたシグナリング数を比較することにより推定可能である。第 2.3 節では、Connected 状態から Idle 状態に遷移する際には UE のコンテキストおよび MME と SGW のベアラ、RRC コネクションの解放を行うと述べた。また、第 2.4 節では、Connected 状態から Connected Inactive 状態への遷移する際には、RRC コネクションを解放すると述べた。以上の知見から、Connected 状態から Idle 状態に遷移する際のシグナリングから、RRC コネクションの解

放に関するシグナリングを除いたものが、Connected Inactive 状態から Idle 状態への遷移する際のシグナリングと推定できる。その際に各ノードで処理するシグナリングの数は以下の様になる。

- UE :0
- eNodeB :3
- MME :5
- SGW :2

2.6 Connected Inactive 状態における、状態遷移を伴わないデータ送信

文献 [15] では、小さなデータ量であれば、Connected Inactive 状態から Connected 状態へ遷移することなく、データ送信が可能であると述べている。まず、UE-eNodeB 間において RA preamble および RA response シグナリングを実行する。その後、UE から eNodeB に対して RRC Active Message を送信するが、それに対してデータをピギーバックする。最後に、eNodeB から UE に対して、RRC Inactivate Message を送信してシグナリングは処理は終了である。その際に各ノードで処理するシグナリングの数は以下の様になる。

- UE :4
- eNodeB :4
- MME :0
- SGW :0

2.7　シグナリング数のまとめ

先行研究を調査することにより、状態遷移に伴うシグナリングの発生数が一部であるが、明らかになった。図 10 に示す状態遷移図と共に、状態遷移に伴って発生するシグナリングに関する情報を、表 1 に示す。

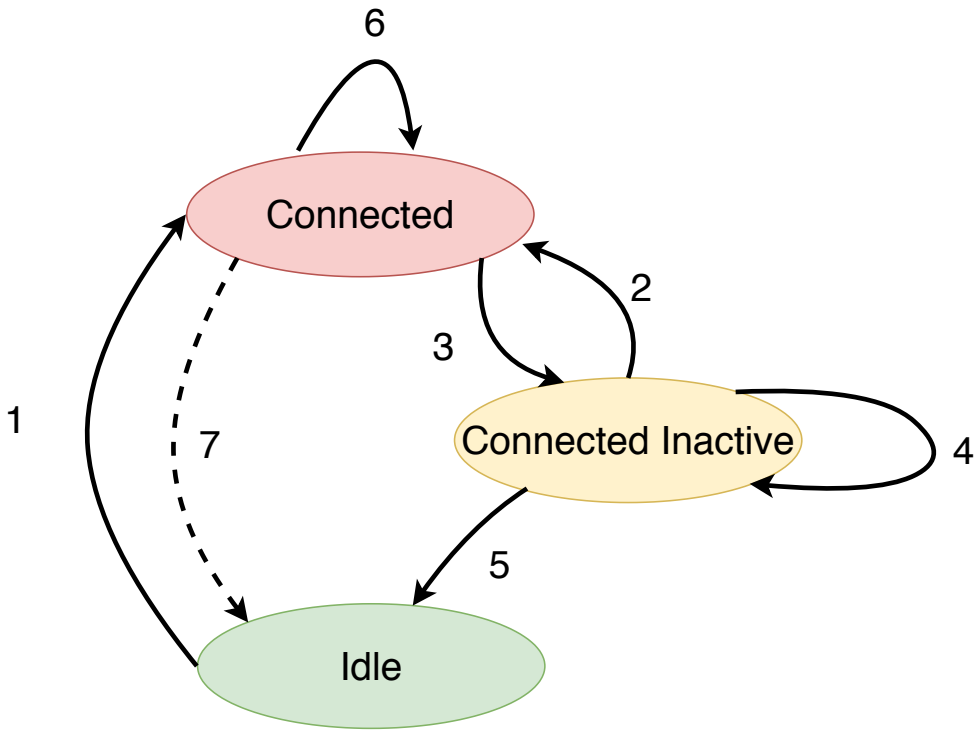


図 10: state transition

表 1: Signaling Load

遷移 ID	シグナリング処理数				遷移条件
	UE	RAN	MME	SGW	
1	9	12	5	2	Packets transmission
2	5	5	0	0	2 or more packets transmission
3	1	1	0	0	Connected timer expiration
4	4	4	0	0	One packet transmission
5	0	3	5	2	Connected Inactive timer expiration
6	0	0	0	0	Packets transmission
7	1	4	5	2	Connected timer expiration

3　今後の課題

- 状態遷移に伴って発生するシグナリングの調査

- Connected Inactive 状態において “状態遷移を伴わないデータ送信”が可能なデータ量の調査。

参考文献

[1] “Disaggregated Servers Drive Data Center Efficiency and Innovation,” Intel Corporation, Technical Report (TR), Jun. 2017. [Online]. Available: https://www.intel.com/content/www/us/en/it-management/intel-it-best-practices/disaggregated-server-architecture-drives-data-center-efficiency-paper.html

[2] “Intel’s Disaggregated Server Rack,” Moor Insights Strategy, Technical Report (TR), Aug. 2013. [Online]. Available: http://www.moorinsightsstrategy.com/wp-content/uploads/2013/08/Intels-Disagggregated-Server-Rack-by-Moor-Insights-Strategy.pdf

[3] C. Devaki and L. Rainer, “Enhanced Back-off Timer Solution for GTP-C Overload Control,” Feb. 2016. [Online]. Available: http://www.freepatentsonline.com/y2016/0057652.html

[4] S. Legtchenko, H. Williams, K. Razavi, A. Donnelly, R. Black, A. Douglas, N. Cheriere, D. Fryer, K. Mast, A. D. Brown, A. Klimovic, A. Slowey, and A. Rowstron, “Understanding Rack-Scale Disaggregated Storage,” in *Proceedings of 9th USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage 17)*. Santa Clara, CA: USENIX Association, 2017. [Online]. Available: https://www.usenix.org/conference/hotstorage17/program/presentation/legtchenko

[5] C.-S. Li, H. Franke, C. Parris, B. Abali, M. Kesavan, and V. Chang, “Composable Architecture for Rack Scale Big Data Computing,” *Future Generation Computer Systems*, vol. 67, pp. 180 – 193, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X16302631

[6] A. Vahdat, H. Liu, X. Zhao, and C. Johnson, “The Emerging Optical Data Center,” in *Proceedings of Optical Fiber Communication Conference/National Fiber Optic Engineers Conference 2011*. Optical Society of America, 2011, p. OTuH2. [Online]. Available: http://www.osapublishing.org/abstract.cfm?URI=OFC-2011-OTuH2

[7] N. Farrington, G. Porter, P.-C. Sun, A. Forencich, J. Ford, Y. Fainman, G. Papen, and A. Vahdat, “A Demonstration of Ultra-low-latency Data Center Optical Circuit Switching,” in *Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, ser. SIGCOMM ’12. New York, NY, USA: ACM, 2012, pp. 95–96. [Online]. Available: http://doi.acm.org/10.1145/2342356.2342377

[8] A. Vahdat, “Delivering Scale Out Data Center Networking with Optics — Why and How,” in *Proceedings of OFC/NFOEC*, Mar. 2012, pp. 1–36.

[9] B. Abali, R. J. Eickemeyer, H. Franke, C. Li, and M. Taubenblatt, "Disaggregated and Optically Interconnected Memory: When will it be cost effective?" *CoRR*, vol. abs/1503.01416, 2015. [Online]. Available: http://arxiv.org/abs/1503.01416

[10] "Arista Networks Cloud Networking Portfolio." [Online]. Available: https://www.arista.com/jp/%20en/products/switches

[11] "Calient S320 Datasheet." [Online]. Available: http://www.calient.net/members-area/?redirect-to=/download/s320-optical-circuit-switch-datasheet/

[12] M. Mahloo, J. M. Soares, and A. Roozbeh, "Techno-Economic Framework for Cloud Infrastructure: A Cost Study of Resource Disaggregation," in *Proceedings of 2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*, Sep. 2017, pp. 733–742.

[13] 3GPP, "Study on architecture enhancements for Cellular Internet of Things (CIoT)," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 23.720, Mar. 2016, version 13.0.0. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2894

[14] I. L. Da Silva, G. Mildh, M. S ä ily, and S. Hailu, "A Novel State Model for 5G Radio Access Networks," in *Proceedings of 2016 IEEE International Conference on Communications Workshops (ICC)*, May 2016, pp. 632–637.

[15] S. Hailu, M. Saily, and O. Tirkkonen, "RRC State Handling for 5G," *IEEE Communications Magazine*, vol. 57, no. 1, pp. 106–113, Jan. 2019.