

# 高速大容量通信技術に関する評価報告

Evaluation report of high-speed, wide-bandwidth data communication technology

濱村 翠\* 藤野 遥子\* 原 周\*

Midori Hamamura, Yoko Fujino, Hiroshi Hara

次世代無線通信規格：5Gサービスの国内運用が本格開始され、5G規格相当の製品開発の需要が急激に高まってきている。5G規格では、高速大容量（eMBB）、低遅延（URLLC）、多数同時接続（mMTC）の3つの特長が定義されており、今後の通信システムの開発においても、ソフトウェアでの高速大容量通信の実現を求められることが予想される。そのような環境変化に備えた、通信速度向上の1つの策として、DPDK（Data Plane Development Kit：オープンソースの高速パケット処理用ライブラリとネットワークドライバのセット）を用いた、汎用サーバ上での高速大容量通信技術の確立を検討する。本稿では、この検討過程で検証した、DPDK適用効果について報告する。

In Japan, some of the cellular phone companies have launched the full services using 5G (the 5th generation technology standard for broadband cellular networks). By this, the demand for 5G products and systems increases rapidly. 5G has three main application areas, enhanced Mobile Broadband (eMBB), Ultra Reliable Low Latency Communications (URLLC), and Massive Machine Type Communications (mMTC). We must implement these merits with the software in the future telecommunication system. We are studying to construct the high-speed and wide-bandwidth data transmission technology on a general-purpose server with DPDK (Data Plane Development Kit). In this paper, we report the result of this consideration.

## 1. まえがき

2020年、従来の4Gに比べ飛躍的な進化を遂げ、5Gサービスの運用が本格的に開始された（表1）。通信システム分野においても、5Gの特長（高速大容量、低遅延、多数同時接続）に対応した性能を持つ製品開発の需要が高まりつつある。2030年代に商用化が期待されているBeyond 5G<sup>(\*1)</sup>では、持続可能な新たな価値の創造に資する特長を付加する議論も始まっている。このような背景から、通信容量の拡大と高速化の流れはますます加速することが予想され、進化し続ける通信システム市場を開拓し続けていくためには、高度な通信技術を先行的に獲得しておくことが重要と考える。

通信システム市場は、IoTの普及により通信ニーズが多様化し、自動車、産業機器、ホームセキュリティ、ス

マートメータ分野のような多種多様なサービスに広がってきている。また、携帯電話事業者による公衆網を用いたサービスだけでなく、地域や事業者のニーズに応じて5Gを活用するシステム（ローカル5G）の導入も広がりつつある。今後、5Gへの移行と共にIoT関連サービスも更に充実し、高速大容量通信のニーズが高まっていくものと予測する。

高速大容量通信を実現する場合、従来は、専用のハードウェアやチップを用い、FPGA<sup>(\*2)</sup>やDSP<sup>(\*3)</sup>で開発することが一般的であった。しかし、通信ニーズが多様化している現在、実現機能に特化した高価な専用ハードウェアではなく、多種多様なサービスを同時に実現できる安価な汎用サーバを用いた通信ソフトウェア開発の需要が見込まれる。つまり、汎用サーバ上のソフトウェアで高速大容量通信を実現しておくことが価格/性能面で大きなアドバンテージとなる。さらに、5Gサービスの運用が始まり、5G規格相当の製品開発の需要が急激に伸びつつある今、ソフトウェアによる高速大容量通信技術の早期確立は急務である。

本稿は、通信速度の向上が期待できる技術の1つである、

表1 4G / 5G 特長比較

項目	4G	5G
伝送速度	0.1 ~ 1 Gbps	10 ~ 20 Gbps
遅延	10 ミリ秒	1 ミリ秒以下
接続数	10 万台 / km <sup>2</sup>	100 万台 / km <sup>2</sup>

\* 関西事業部 第三技術部

DPDKに関する評価結果について報告するものである。

## 2. DPDK を用いた高速大容量通信

DPDK は、高速パケット処理を実現するネットワーク処理高速化技術の1つである。2010年にIntel DPDKとしてリリースされた。リリース当初は、Intel製の一部NIC<sup>(\*)4)</sup>のみへの対応であったが、2013年にDPDK.orgが設立され、DPDKに名称を変えて以降、AMDのaxgbeや、Atomic Rulesのark、CiscoのeNic、Intelのe1000など<sup>(\*)5)</sup>、徐々に対応するNICを増やし、汎用性を高めている。

従来のアプリケーションは、カーネル上で動作し、ハードウェアを意識することなく、カーネルの提供するシステムコールを呼び出すことで、ハードウェアを制御する。カーネルは、ハードウェアの制御を共通的に処理できるように実装していることから、個々の制御の性能面では、ハードウェアの持つ性能を、最大限に発揮できていない。カーネルの通信処理は、この、本来の実力を発揮できていない部分に含まれる。これに対しDPDKは、従来カーネルで行っていたNICの制御をユーザ空間で行うことを目的とし、カーネル機能をバイパスすることで、NICの性能を最大限まで引き出す高速パケット処理を実現している(図1)。

このように理論的には、DPDKを用いることで、通信に関するカーネルの冗長な処理部分を通さずに、高速大容量通信を実現できる。しかし、製品開発に導入するためには、導入の容易さはもちろんのこと、メリット、デメリットを押さえたうえで判断する必要がある。そこで現在、高速大容量通信を模擬する環境を用意し、

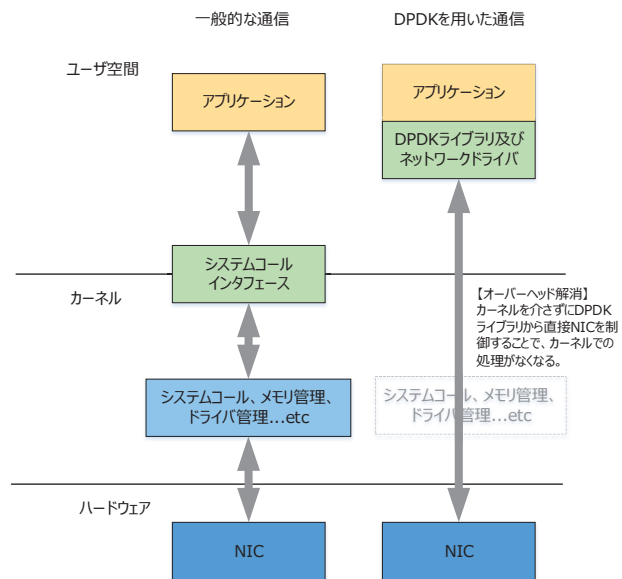


図1 DPDKの仕組み

DPDKを適用した試作ソフトウェアの開発、評価に取り組んでおり、その過程で得たDPDK導入のための制約、留意点を交え、次章以降で評価環境、方法、結果、課題について順に述べていく。

## 3. 評価環境

本章では、DPDK適用評価の環境について記述する。

### 3.1 ハードウェア構成

本評価で使用するハードウェア環境を表2に示す。CPUは後述する通信制御ソフトウェア・モデルを適用するため、6コア以上のものを選定し、DPDKに対応したNICを搭載する。

### 3.2 評価環境

DPDKの評価環境を図2に示す。

試験機は1GB NICを搭載し、送信/受信の各機材を用意する。測定系は、2系統用意し、1Gbps環境と、2Gbps環境を構築する。2Gbps環境では、各機材をそれぞれ2セット用意する。

伝送スループットを測定するために、ネットワーク・パフォーマンス測定ツールiperf 2.0.10を使用する。受信機をサーバモードで、送信機をクライアントモードで起動し、評価用ハードウェア経由でパケットの送受信を行う設定とする。

表2 DPDK動作ハードウェア

項目	説明
CPU	Intel Xeon CPU D-1577
	クロック周波数 1.30GHz
	CPU数 1
	コア数 16
NIC	メモリ容量 16Gbyte
	Intel Ethernet Connection X552 / X557-AT 10GBASE-T

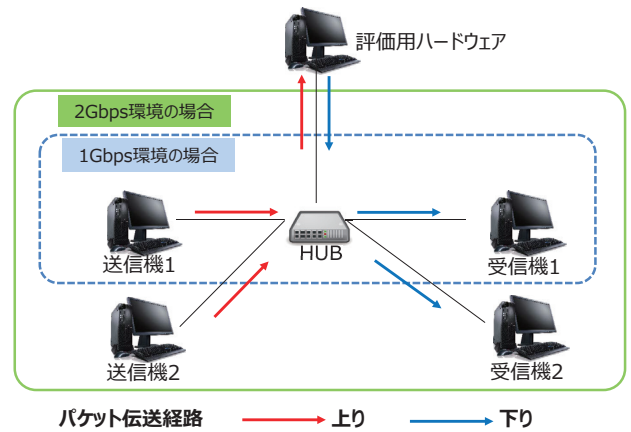


図2 DPDK評価環境 (1Gbps / 2Gbps)

#### 4. 評価方法

本章では、DPDK の有用性を判断するために、表3に示すとおり、段階的に評価を行う。

2つの評価手順では、それぞれ目的に応じた評価用のソフトウェアを準備する。本評価で使用するソフトウェアを表4に示す。

以降、各手順で使用する評価対象ソフトウェアの詳細と、評価する測定項目について述べる。

##### 4.1 対向通信モデルでの DPDK 評価

SWA、SWB の2モデルを用いて評価を行う (図3)。

SWA、SWB は、共に送信機から受信したパケットを受信機に転送する同一の機構とし、パケット送受信部分のみを置き換える。SWA は Linux Kernel の標準通信ライブラリを、SWB は DPDK ライブラリを使用して両者の性能を比較する。DPDK は受信契機を常時監視する仕組みにより性能が発揮されるため、SWB は受信/送信

表3 DPDK 評価手順

手順	評価内容	説明
1	対向通信モデルでの性能評価	DPDK 適用/非適用のプログラムを各々試験機と対向させ、DPDK による速度性能の優位性を確認する。
2	通信制御ソフトウェア・モデルへの適用評価	通信制御ソフトウェアをモデル化し、評価手順1の結果と比較して DPDK 適用効果のあるシステム要件を抽出する。

表4 評価ソフトウェア

項目	名称	バージョン/説明
OS	Ubuntu	18.04 LTS
Kernel	Linux	4.15.0-72-lowlatency
ライブラリ	DPDK	19.11
評価対象	SWA	DPDK 非適用プログラム
	SWB	DPDK 適用プログラム
	SWC	DPDK 適用プログラム (通信制御ソフトウェア・モデル)

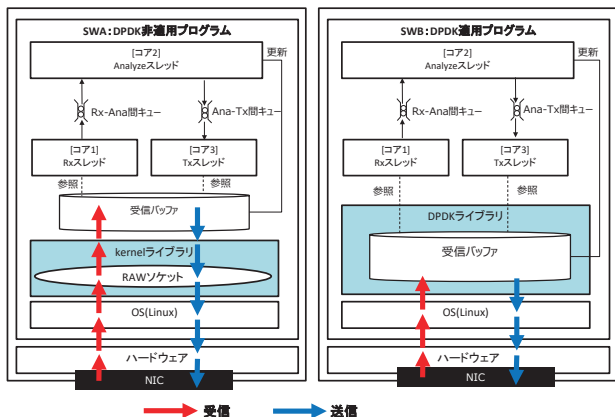


図3 ソフトウェア構成図 (評価手順1)

各々専用のコアを割り当てる。SWA はSWB との比較評価を行うため、SWB と同じコア割り当てとする。

##### 4.2 通信制御ソフトウェア・モデルへの適用評価

SWB に一部受信パケットを外部に転送する機構を持たせた SWC モデルを用いて評価を行う (図4)。

SWC は、主に自社で開発している通信制御ソフトウェアに DPDK を適用する場合を想定し、モデル化している。通信制御ソフトウェアが取り扱うパケットは、表5のように制御データパケット (C-Plane、M-Plane) とユーザデータパケット (U-Plane) に大別できる。

通常、制御データパケットの伝送量はユーザデータパケットの数パーセント程度で、制御データパケットが複雑な処理を伴うのに対し、ユーザデータパケットは簡易な処理と転送処理が中心となることが多い。このようなデータ特性から、通信制御ソフトウェアではデータ種別に応じて処理を分離して設計している。高速大容量通信によりパケット伝送量が増えるケースは主にユーザデータパケットと想定されるため、SWC では、その処理部分を分離して DPDK を適用したモデルとする。DPDK は NIC を占有するため、ユーザデータパケットと制御データパケットの双方を DPDK 適用プロセスで受信するが、制御データをプロセス間通信により Linux Kernel 上で動作するプロセスへ転送することで処理を分離する。

これらを踏まえ、図4で示す SWC は受信パケットの

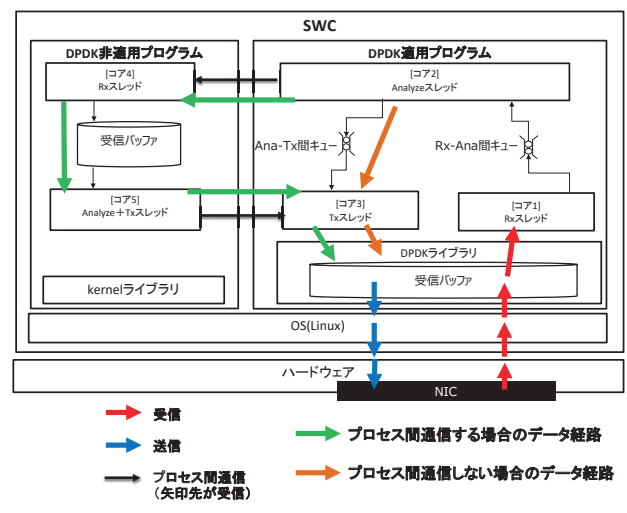


図4 ソフトウェア構成図 (評価手順2)

表5 通信制御ソフトウェア処理パケット

種別	説明	伝送量	プロトコル
C-Plane M-Plane	制御、監視等のイベントにより発生するパケット	小	TCP
U-Plane	音声、映像等の定常的に伝送されるパケット	大	UDP

転送経路を2つ設定する。プロセス間通信を介さずにDPDKでパケットの転送を行う経路と、プロセス間通信を介してパケットの転送を行う経路である。

### 4.3 測定項目

本評価で確認した測定項目を表6に示す。各項目はデータ長ごとに180秒間測定する。

表6 測定項目一覧

項目	測定ツール	周期
ジッタ	iperf	1秒
レイテンシ	ping RTT (往復時間)	1秒
スループット	iperf	1秒

※ジッタとは、伝送時のデータの揺らぎのこと。レイテンシとは、データの応答時間のこと。スループットは、データの伝送速度を表す。

## 5. 評価結果・考察

各測定項目の結果を以下に示す。また以降、SWA、SWB、SWCのデータと併せ、比較用に評価用ハードウェアを経由せずに試験機をエンド・ツー・エンドで直結して測定した値（以降、直結）も示す。

### 5.1 ジッタ

2Gbpsでの各データ長でのSWA、SWB、直結の平均ジッタ測定結果を図5に示す。

SWBでは、32～1024byteまで直結に近い測定結果となった。SWAと比べ、データ長あたりの変動も少ない。SWAは32～1024byteまで送信機1と2で差異もみられるが、総じてSWBはSWAよりジッタが低い。

これらの理由として、以下が挙げられる。

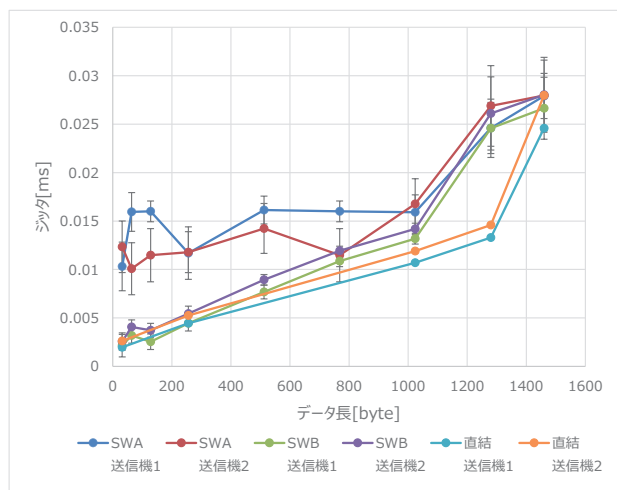


図5 SWA、SWB、直結平均ジッタ

Linux KernelとDPDKのパケット処理を比較した場合、DPDKのほうが1パケット当たりの処理効率が良い。これは、Linux Kernelのパケット処理では、netfilterによるパケットフィルタリングやルーティングが行われるが、DPDKではそれらを実施しないため、パケット処理の効率が良いことが要因と推測する。

さらに、データ長が1024byte以降では、DPDK適用有無に関わらず、ジッタが増加している。これは、測定に使用した送信機の性能限界により、送信時点で既にジッタが増大したことによるものである。5.3節で後述するとおり、データ長が1024byte以上の場合は、1つの送信機当たりのスループットが1Gbps近くに到達していた。送信機に搭載されているNICは1GbEであるため、性能限界に近いスループットが入力されていたことになる。このような状況ではNICの性能がボトルネックとなり、測定結果が悪化したものと考えられる。本問題を解消するには、1Gbps以上の性能を有するNICを送信機に搭載する必要がある。

さらに、SWBと直結は長時間稼働した場合に、周期的な変動や連続的なバーストが発生しないかを確認するために、1000秒間の測定を行った。ジッタ発生分布は、正規分布に従うと仮定し、代表値としてデータ長32byteでの測定結果から得られた確率密度分布を、図6に示す。

図6から、SWBと直結の分布傾向はおおむね同じであり、周期性や連続的なバーストの発生等の特異な傾向は見られない。

### 5.2 レイテンシ

2Gbpsでの各データ長でのSWA、SWB、直結の平均レイテンシ測定結果を図7に示す。

SWAは全体的にレイテンシの変動が大きい。一方で、SWBは変動が小さく、直結に近い測定結果となった。また、データ長が1024byte以降では、DPDK適用有無に関わらず、レイテンシが増加している。

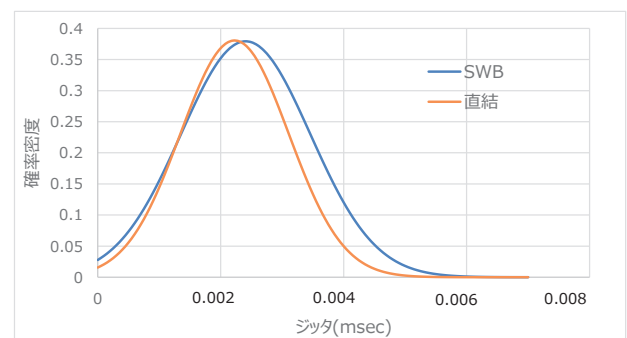


図6 ジッタ発生の確率密度分布（データ長32byte）

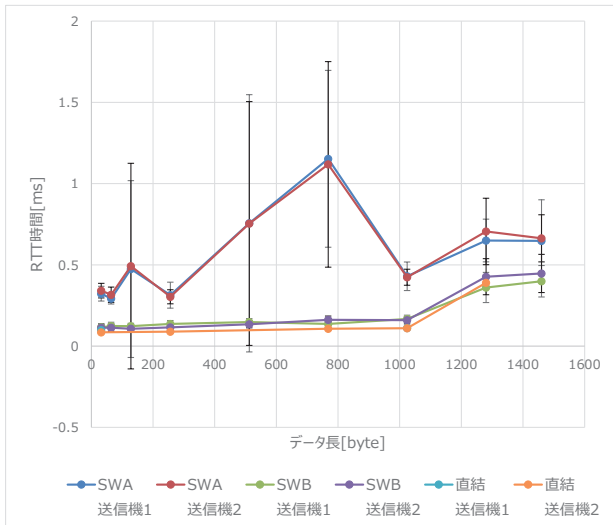


図7 SWA、SWB、直結平均レイテンシ

この傾向は、ジッタと同様であることから、ジッタと同じ要因が発生しているものと考えられる。

レイテンシも、長時間稼働した場合に周期的な変動や連続的なバーストが発生しないかを確認するために、1000秒間の測定を行った。レイテンシの分布も、正規分布に従うと仮定し、代表値としてデータ長32byteでの測定結果から得られた確率密度分布を図8に示す。

図8から、直結とSWBでは中央値に差はあるものの、分布傾向はおおむね同じであり、周期性や連続的なバーストの発生等の特異な傾向は見られない。

### 5.3 スループット

1 Gbps、2 Gbps での各データ長でのスループットの結果を以下に示す。

SWCの測定に関しては、SWAとSWBで差異が見られない場合は、SWCも同様の結果になると考え、SWBとSWAで差異がみられた2GbpsのUDPのみ測定を行った。測定内容も、通信制御ソフトウェア・モデルを適用した適用効果を確認するために、プロセス間通信のトラフィックを変化させ測定を行った。一般的に、通信制御ソフトウェアでは、制御データは、データ長が100byte未満のものが多く、必要とされる速度性能も1Mbps未満となるものが大半を占める。このことを踏まえ、表7に示す比率で、制御データとユーザデータを

表7 データ分割内容

データ	内容
制御データ	データ長：64byte 流量：SWBの全流量の1%
ユーザデータ	流量：SWBの全流量から制御データ分を除いたもの

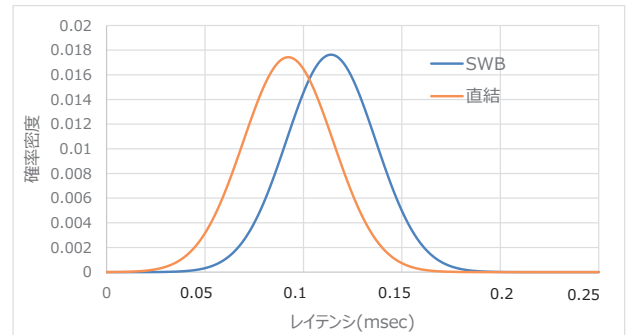


図8 レイテンシ確率密度分布 (データ長32byte)

分割し測定した。また、プロセス間通信による処理遅延及び通信遅延が、システム全体の速度性能に与える影響を確認するために、全データをプロセス間通信した場合も併せて測定した。

以下で、各測定結果の考察を述べる。

#### (1) 適用効果について

図9から、TCPとUDP共に、1GbpsではLinux KernelとDPDKを用いた場合で速度性能差はない。図10から、2Gbpsでは、DPDKの効果は顕著に現れた。

#### (2) 有効範囲について

図10のUDP通信において、データ長が1024byte付近で、送信機と受信機の一組当たりのスループットが、1Gbpsに到達し、NICの性能限界に至っていた。このため、今回の測定では1024byte未満を有効範囲として考える。

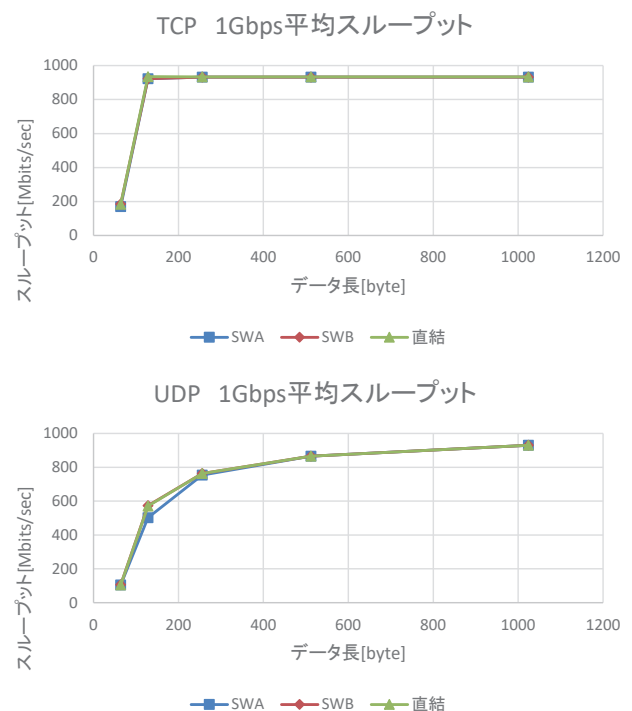


図9 1Gbps平均スループット

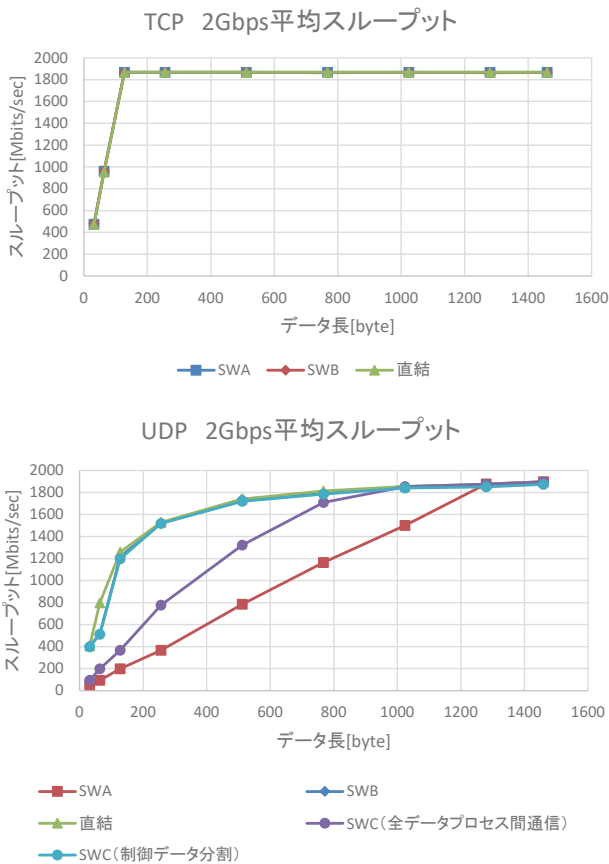


図 10 2 Gbps 平均スループット

(3) 通信制御ソフトウェア・モデルについて

表 8 から、今回の測定の有効範囲において、DPDK 適用による速度性能の改善効果は約 1.5～7.7 倍得られることが分かった。また、全データプロセス間通信を行った場合においても、SWA よりも約 1.4～2.2 倍速度性能が改善している。また、制御データ分割を行った場合は、約 1.5～7.5 倍の速度性能となり、SWB と同程度のスループットになった。

このことから、制御データを別プロセスで処理したとしても、DPDK を適用したユーザデータの通信には影響がないと考えられる。つまり、通信制御ソフトウェアの構成でユーザデータの高速度通信が実現可能であることがわかった。

2 Gbps では、速度性能の改善効果が得られたが、TCP 通信と UDP 通信では得られる結果に差が生じた。この点について考察する。

(1) TCP 通信について

今回の結果から、TCP での DPDK 適用有無による性能差はない。また、UDP と比べ比較的短いデータ長から通信帯域の限界まで速度性能が出ている。これは、TCP の具備するウィンドウサイズ機構により、データ長の短い

表 8 2 Gbps スループット比

データ長 [byte]	スループット比 (対 SWA) [倍]				測定 有効 範囲
	SWB		SWC		
	TCP	UDP	UDP		
			全データ プロセス間 通信	制御 データ 分割	
32	1.01	7.79	2.25	7.58	○
64	1.00	5.44	2.10	5.44	○
128	1.00	6.10	1.84	6.01	○
256	1.00	4.14	2.11	4.14	○
512	1.00	2.21	1.68	2.19	○
768	1.00	1.54	1.47	1.53	○
1,024	1.00	1.24	1.24	1.23	-
1,280	1.00	0.99	1.00	0.99	-
1,460	1.00	1.00	1.00	0.99	-

場合もまとめて送信するためであると考えられる。実際に、データ長 32 byte を指定して送信した場合でも、伝送時の実際のデータ長は 400～2000 byte であり、指定したデータ長よりも長いパケットとなっている。データ長の長いパケットは、短いパケットよりもスループットを得やすいため、UDP の場合と比較して早い段階で速度性能が出る結果となっている。

また今回は送信側の性能も 2 Gbps であり、TCP 通信では送信側によって通信速度が制御されるため、結果的に Linux Kernel でも処理できる程度の速度となり、TCP 通信では DPDK の速度優位性は確認できなかった。今後、1 Gbps 以上の通信性能をもつ NIC などを用いて性能を確認する必要がある。

(2) UDP 通信について

今回の結果から、UDP では DPDK の適用有無による性能差があった。特にデータ長の短い場合において、より顕著であった。これは、パケット伝送量に差があるためと考えられる。

UDP 通信においては、同一帯域の場合、ヘッダなどが付加されるため、帯域上限まで送信するパケット量はデータ長が短いパケットほど多くなる。今回実測した結果、データ長 32 byte の場合のパケット量は、1460 byte の場合の約 27 倍であった。このように、UDP 通信ではデータ長の短い場合は、パケットの伝送量が多くなる。

データ長が短い場合に性能差がより顕著であったことから、DPDK は 1 パケットあたりの処理効率が Linux Kernel よりも高いことがわかる。

これは、DPDK が Linux Kernel を用いた場合とは異なり、ハードウェアからの処理契機となる割り込みやカーネル空間のソケットバッファへのコピー、プロトコルスタックの解析処理を行わずにユーザ空間にパケットが伝達されることに起因する。このため、DPDK は 1 パケットあたりの処理効率が高くなる。

## 6. 結論

本章では、本評価で得られた結果を基に、DPDK 適用効果とシステム要件を整理する。また、本測定のプロセスで判明した DPDK 適用時の注意点についても述べる。

### 6.1 DPDK 適用効果

本測定の結果を踏まえ、Linux Kernel で処理する場合と比較した DPDK 適用による効果は次の2つである。

- (1) 2 Gbps 以上の帯域で、特にデータ長が 256 byte 以下のパケットを扱う場合は、2～7.5 倍の大幅なスループット改善を期待できる。
- (2) ジッタ変動 0.006 ms 以下、レイテンシ 0.2 ms 以下に抑えられ、通信品質の向上が見込める。

今回の測定では、1 Gbps 以下のスループットでは適用効果は得られなかった。1 Gbps 以上のスループット要否が DPDK 適用の判断基準になると考える。

DPDK のホームページ等で公開されている測定結果は複数 NIC を使用して測定しているケースが多いため、これまでは単一の NIC での効果有無を判断できなかった。しかし、本評価では、1つの NIC を使用して DPDK の評価を行い、改善効果があることを確認できた。

今回の結果は、1つの NIC のみを搭載するシステムに対し、DPDK 適用する際の参考情報として活用できると考える。

### 6.2 システム要件

本測定結果を踏まえて、システムに DPDK を適用するための必要要件は次のとおりである。

- (1) NIC が DPDK に対応していること
- (2) DPDK が NIC を占有できること
- (3) DPDK を割り付けるコアを占有できること（複数コアが存在すること）

### 6.3 DPDK 適用時の注意点

DPDK を適用する場合の注意点として、受信処理がポーリング処理であることと、ネットワークプロトコルスタックを使用できない点が挙げられる。

DPDK は、パケットの受信において NIC をポーリングすることを前提として実装されている。このため、DPDK でパケット受信を行うプログラムは、NIC を常に監視することで、パケットの取りこぼしを防止することになるため、コアの CPU 負荷は 100% となる。受信パケットを転送するだけの単純なシステムであれば問題は無いが、受信したパケットを解析し、解析結果に応じて制御を行うシステムでは、処理量の増大に応じ

て、受信パケットの取りこぼしを引き起こす可能性が高まる。このような場合は、受信したパケットをプロセス間通信で他のプロセス（他のコア）に転送し、解析や制御を移譲する仕組みが必要となる。

また、DPDK は Linux Kernel のネットワークプロトコルスタックを使用できない。したがって、DPDK を使用してプロトコル処理を行う場合は、IP、ICMP といった利用頻度の高いプロトコルについても、独自でネットワークプロトコルスタックを開発する必要があるため、DPDK 適用による性能効果とネットワークプロトコルスタックに要するコスト/期間とのトレードオフを、十分に検証する必要がある。

## 7. 課題と今後の展開

本評価を踏まえ、課題及び今後の展開は次のとおり。

### (1) 10 Gbps 環境での検証

環境準備の都合で、1 GB NIC を 2つ使用して 2 Gbps の帯域を確保したネットワーク環境で測定を行った。しかしながら、1 GB NIC の性能限界により Linux Kernel と DPDK の TCP 通信での性能差異の確認で十分な測定が出来なかった。また、5 G の最大スループットは 20 Gbps まで到達するといわれていることから、測定環境の帯域の向上が必須である。直近では、市販品でも構築が容易な 10 GB NIC を用いたネットワーク環境を構築し、更なる評価を実施していく予定である。

### (2) パケットサイズ拡張による効果増大の検証

パケットサイズによる DPDK の優位性が確認された。本測定では適用に至らなかったが、ジャンボフレームを有効化した場合、パケットサイズが大きくなるため、より一層の改善効果が得られる可能性がある。このため、今後はジャンボフレームを有効化したネットワーク環境での評価も行う予定である。ジャンボフレームを有効にすることで、SWC ではプロセス間通信にかかる処理負荷も増えていくため、ジャンボフレームの有効化が、SWC の構成における性能に与える効果と影響も引き続き検証してゆく。

### (3) プロセス間通信の改善

本測定では、プロセス間通信の実現方式として、名前付きパイプを使用した。さらに、他のプロセス間通信方式、例えば、共有メモリ等の別のプロセス間通信を使用することで、更なる速度改善の余地を検討する。

### (4) ノウハウの獲得

DPDK を使用したソフトウェアの開発ノウハウの獲得が必要である。本測定のために、DPDK の開発環境の構築方法や、実機へのデプロイ方法といったノウハウを獲得できた。ただし、あくまで DPDK が有する多くの

機能のうち、一部機能の利用にとどめているため、評価できていない機能についても評価を進めてゆく。また、DPDK 適用時の制約として、DPDK の専用ネットワークドライバは Linux Kernel の制御外となることがある。これにより、Linux で使用できるネットワーク管理系コマンド (ifconfig 等) やデバッグ等が使えない。このように、保守性や解析性にデメリットがあるため、これらの制約についても解決してゆく。

## 8. むすび

本稿では、ソフトウェアの高速大容量技術実現に向けた 1 つの手段として、DPDK 導入による効果が、どの程度見込めるのか、Linux Kernel を用いた場合との比較評価について述べた。

評価の結果、DPDK 導入は有用であり、伝送速度の大幅な向上、通信品質の改善が得られることが分かった。加えて、評価作業を通じて、製品開発への導入に向けた課題設定が可能となった。

今後は、これらの課題を解決するとともに、我々の持つソフトウェア資産に DPDK を適用し、より具体的なソフトウェアの評価を行っていきたいと考えている。

- 
- \* 1 第 6 世代移動体通信システム：6G のこと
  - \* 2 Field-Programmable Gate Array
  - \* 3 Digital Signal Processor
  - \* 4 Network Interface Card
  - \* 5 最新のサポート状況は DPDK ホームページ参照

## 参考文献

- (1) DPDK : Data Plane Development Kit  
<https://www.dpdk.org/>
- (2) Harald Welte : The journey of a packet through the linux 2.4 network stack  
<http://ftp.gnumonks.org/pub/doc/packet-journey-2.4.html>

- (3) 荻原直彦：第 5 世代移動通信システム（5G）の今と将来展望，総務省総合通信基盤局（2019 年）  
[https://www.soumu.go.jp/main\\_content/000633132.pdf](https://www.soumu.go.jp/main_content/000633132.pdf)
- (4) 総務省：Beyond 5G 推進戦略（概要）（2020 年）  
[https://www.soumu.go.jp/main\\_content/000702111.pdf](https://www.soumu.go.jp/main_content/000702111.pdf)

Intel, Xeon は、Intel Corporation 又はその子会社の米国及びその他の国における商標又は登録商標です。  
AMD は、Advanced Micro Devices, Inc. の商標又は登録商標です。

Cisco は、Cisco Systems, Inc. 及びその関連会社の米国及びその他の国における商標又は登録商標です。

Ubuntu は、Canonical Ltd. の商標又は登録商標です。

Linux は、Linus Torvalds 氏の米国及びその他の国における商標又は登録商標です。

## 執筆者紹介

### 濱村 翠

2002 年入社。入社以来、携帯電話基地局や列車無線等の、情報通信システム、組込みシステムの開発に携わる。

### 藤野 遥子

2013 年入社。入社以来、情報通信システム、組込みシステムの開発に携わる。

### 原 周

2007 年入社。入社以来、衛星電話基地局やネットワーク機器等の、情報通信システム、組込みシステムの開発に携わる。