Vol.2018-IOT-43 No.9
Vol.2018-SPT-30 No.9
2018/9/28

Network Processor を用いた高精度遅延計測装置の開発

谷所 基行[†1]　大久保 克彦[†1]　江原 鉄男[†1]
中山 典保[†1]

概要：SDN/NFV 環境の品質確保のためには，低コストかつ高精度な遅延測定環境が必要であるが，専用測定器は高価であり，ソフトウェア測定器は精度が得られないといった問題がある．これらの問題を解決するため，Network Processing Unit (NPU)を搭載した SmartNIC を用いて，サブマイクロ秒の高精度な転送遅延計測装置を開発した．本稿では，その実現方法・実装について示すとともに，本装置を用いて測定した遅延の測定結果について報告する．

キーワード：NPU, SmartNIC, 遅延測定

High precision latency measurement by using Network Processor

TANISHO MOTOYUKI[†1]　OOKUBO KATSUHIKO[†1]　EHARA TETSUO[†1]
NAKAYAMA NORIYASU[†1]

Abstract: It is necessary to offer the latency measurement system that satisfies both conditions that are the reasonable cost and the high precision, to the engineers who operate SDN/NFV services in order to keep the qualities. However, there are 2 major problems. One is generally said that the solution implemented in hardware is too expensive. And the other is that the solution implemented in software is too low in precision.
Because of that, we have developed a latency measurement system by using general purpose IA server equipped with Network Processing Unit (NPU)-based SmartNIC. This concept can meet the both conditions such as cost and precision. This paper shows the design and implementation of our solution, and also shows some of the latency measurement results.

Keywords: NPU, SmartNIC, latency measurement

1. はじめに

近年，5G を主とした，リアルタイム性が必要な分野への適用による新たなサービスを展開しようという動きが活発になってきている．代表的な例で言えば，遠隔地ユーザーとの協力または対戦オンラインゲームやスポーツ観戦におけるリアルタイムな音声･動画のストリーム配信サービス，医療や金融，車載への活用など多岐にわたる．これらのサービス提供においては，End-to-End で 1~10m 秒といった低遅延通信の実現が要求されると考えられている [1] [2]．また，これらサービスを運用するために必要とされるネットワークシステムには，これまでの専用ハードウェアを調達して構築するのではなく，近年 SDN/NFV で盛んに提案されている汎用サーバーを調達して仮想的なネットワークを主として構築する方法が増えてきた．

しかし，汎用サーバーを主として構築されたネットワークシステムの場合，専用ハードウェアでは起こり得ないソフトウェアに起因した遅延が発生することがある．サービスの提供者は，発生した遅延の原因を早期に突き止め，適切な改善が求められるが，遅延はシステムの輻輳状態などにより運用中に大きく変動することがある．例えば，一部のユーザーパケットで一時的に大きな遅延が発生し，通信品質が低下したとしても，サービス提供者はそのような遅延が発生していた事実を把握できない．この問題を解決するためには，常設可能，かつ全パケットの遅延量を測定できる環境が必要となる．ただし，前述の要件を満たす測定器の候補に挙げられる，専用測定器やソフトウェア測定器はそれぞれ，この測定環境に適さない問題がある．

それらの問題を解消するために，我々は NPU を搭載した 40GE 対応 SmartNIC を用いたサブマイクロ秒の高精度な遅延計測装置を開発した [3].

以下，本稿では開発した測定器の実装概要と検証の実施例について示す．まず 2 章で遅延の種類と対応する測定器の条件について整理し，3 章で要求実現に向けた施策と 4 章で実装の詳細について述べる．5 章では開発した測定器を用いて，実際に運用中のシステムに対して検証した結果とその考察について述べ，6 章で論文をまとめる．

†1 富士通アドバンストテクノロジ株式会社
FUJITSU ADVANCED TECHNOLOGIES Ltd.

Vol.2018-IOT-43 No.9
Vol.2018-SPT-30 No.9
2018/9/28

2. 遅延の種類と対応する測定器条件

前項で述べたソフトウェアに起因した遅延とは，主に(1)ソフトウェア更新によるパケット処理の変化，(2)割り込み処理による遅延増加，(3)フローテーブル更新による一時的な遅延増加，の 3 種類に大別される．これらはソフトウェアの実装方法次第で大きく変わるため，安定的な運用維持のためには遅延評価を運用者自身が行う必要がある [4]．その遅延評価をするための測定器として，一般に専用測定器やソフトウェア測定器が挙げられるが，上記 3 種類のケースにおいて，専用測定器やソフトウェア測定器が要求を満たすか整理する．

(1) ソフトウェア更新によるパケット処理の変化

ここでは，Open Flow Switch を採用する場合を考える．Open Flow Switch におけるコントローラーはソフトウェアであるため，更新処理が起きると，処理中のパケットに影響し，遅延が変動する可能性がある．ソフトウェアにもよるが，更新のたびに測定が必要となることを考慮すると，常設できる程度に安価に入手可能な測定器が求められ，高価な専用測定器は採用しづらい．

(2) 割り込み処理による遅延増加

CPU が高負荷状態のときに，何らかの割り込み処理が発生すると，パケット処理が待たされて，遅延が増加してしまうことがある．これらを評価するためには意図的に負荷をかけられるような性能を有する測定器が求められ，専用測定器は可能であるが，ソフトウェア測定器では性能を出すことは難しい．

(3) フローテーブルの更新による一時的な遅延増加

Open Flow Switch はパケット処理高速化のためのハードウェアを含むこともあり，大半のパケットは低遅延で処理される．しかし，一時的に多くのフローが流れた場合，ハードウェアが持つフローテーブルは有限であるため，エントリーが溢れてしまうと，ソフトウェアで別途管理されているフローテーブルに登録するため，パケットはサーバーに転送される．結果としてソフトウェアで処理されることとなり，低遅延を維持できなくなる．ソフトウェアで処理されたパケットは前述の更新処理や割り込み処理の影響を受け，さらに大きな遅延を発生することが容易に想像される．こうした大きな遅延量を持ったパケットを取りこぼしてはならないため，測定器には必然的に全パケットの遅延測定が求められる．専用測定器ならば容易だが，ソフトウェア測定器の場合性能的に容易ではなく，揺らぎによって得られた遅延量の精度も低い．

3. 要求の実現に向けた施策

2 章で述べたことから，専用測定器やソフトウェア測定器は，求められる全ての要求を満たさない．全てを満足するためには，従来とは異なる新しい測定器が必要である．

そこで我々は，NPU を搭載した SmartNIC と汎用サーバーを組み合わせることによる，コストを抑えたハードウェア構成に加え，従来汎用サーバー上で動作していたソフトウェア測定器の処理の一部を SmartNIC にオフロードすることで高性能化・高精度化を実現する手法を提案する．この方式に基づいて開発した測定器について，次節以降にその概要を示す．

3.1 SmartNIC+汎用サーバー

SmartNIC とは昨今，複数のメーカーから提供されている，NPU や FPGA などのプログラマブルなデバイスを搭載した NIC (Network Interface Card)である．従来の汎用 NIC は，MAC や VLAN フィルタリング，ARP 応答，チェックサムといった単純な処理であれば設定変更で対応可能だが，複雑なプロトコル処理やトラフィック制御，あるいは特殊な機能追加などには対応できない．SmartNIC は，汎用 NIC では対応できなかったパケット処理を可能にするために登場した．これにより，従来汎用サーバー上で処理されていた複雑なパケット処理をオフロードすることが可能となり，性能向上や品質改善が期待される．

3.2 ソフトウェア測定器の機能オフロード実現性

我々は 3.1 で述べた，SmartNIC によってパケット処理がオフロードできる点に着目し，オープンソースのソフトウェア測定器の処理オフロードの実現性を検討した．

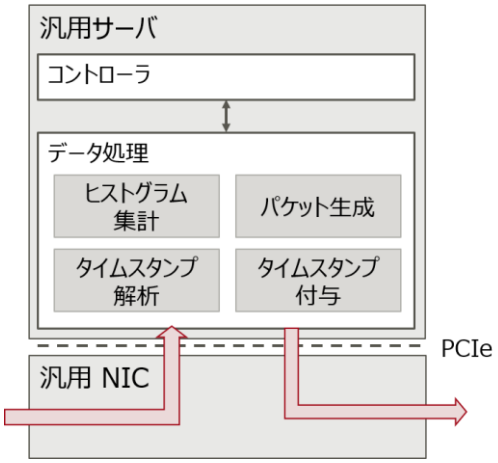

図 1 ソフトウェア測定器の概念図
Figure 1 Block diagram of Software measurement

まずは，ソフトウェア測定器が持っている問題について整理するため，図 1 に一般的なソフトウェア測定器の概念図を示す．ソフトウェア測定器は，主にコントローラーとデータ処理の 2 つのブロックで構成されており，データ処理ブロックで生成されたパケットが，PCIe 接続された汎用 NIC を経由して送受信される仕組みとなっている．図 1 か

らわかるように，送受信処理は PCIe を経由するので，この帯域がボトルネックとなり．高性能化を困難にしている．

次に，遅延測定に必要となるタイムスタンプの付与や解析処理について考える．汎用 NIC を経由したパケット送受信が，たとえマイクロ秒精度でできていたとしても，2 章で述べたように，ソフトウェアには割り込み処理による遅延が発生することがある．もし受信したパケットを解析する直前で，割り込み処理が発生した場合，正しい遅延量を算出できず，最悪ミリ秒オーダーの揺らぎによって，誤った遅延量を集計することになる．

送受信した全パケットの遅延を集計するヒストグラム処理についても同様の問題がある．CPU の負荷が大きくなると取りこぼしが発生してしまい，大きな遅延が発生したパケットも正しく集計できなくなる．

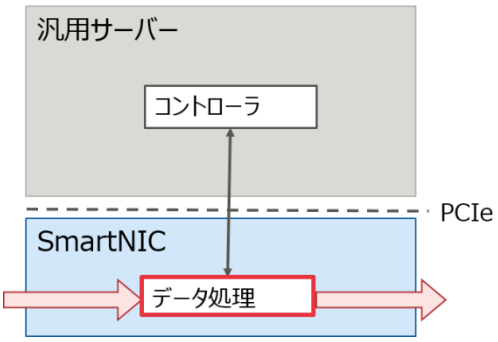

図 2　ソフトウェア測定器の SmartNIC オフロード
Figure 2　Offloading Software measurement to SmartNIC

前述した制約を解消するためには，データ処理部を汎用サーバー上で実行しないことである．我々はこのデータ処理部を SmartNIC にオフロードし，PCIe 帯域による性能ボトルネックを解消すると共に，遅延測定に影響する揺らぎを回避することを目指した(図 2).

4. 実装

4.1 アーキテクチャ

図 3　Agilio-CX 1x40GE (ISA-4000-40-1-2)
Figure 3　Agilio-CX 1x40GE (ISA-4000-40-1-2)

今回採用した SmartNIC は，Netronome 社製の Agilio-CX 1x40GE (ISA-4000-40-1-2)である(図 3) [5].

この SmartNIC に搭載される NPU(NFP-4000)は，約 70 個のパケット処理用コアの並列実行が可能であり，パケット生成・終端，およびプロトコル処理までを実装し，フルワイヤー転送(14.88096[Mpps] x 4[port])が実現可能である．また，ハードアシストとして MAC Rx/Tx における timestamp 付与機能を活用することで，パケットの送信直前および受信直後の timestamp を取得することが可能となり，揺らぎの無い高い信頼度の遅延量が算出可能となる(図 4).

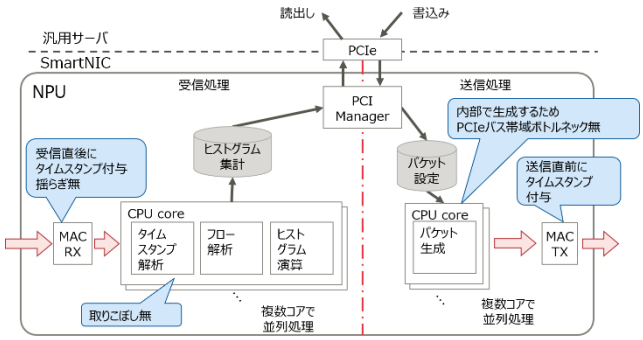

図 4　NPU による測定器の実装概念図
Figure 4　Diagram of Implementation in NPU

同様の実現方法に，FPGA による報告例もあるが [4]，NPU は FPGA に比べ，パケット処理に特化しており，機能制約はあるが，ファームウェアが容易に開発可能で柔軟な機能変更が可能であり，かつ低コストである．

このように，NPU が持つ特徴を最大限活用し，高性能と高精度な遅延測定の要件を満足する測定器を実現した．

4.2 測定器単体の性能評価

開発した測定器が実検証で使用可能なレベルのものであるか，2 つの観点で測定器単体を評価した．

4.2.1 フルワイヤー転送性能評価

本評価では，測定器のポートに対しループバック接続することによって，測定器単体の転送性能を測定した．テストパケットとして，MAC アドレスや IP アドレスの値に 16,000 フローを割り振ったプロトコルデータを作成した．また，パケット長を 64 バイトから 1500 バイトまでの 7 種類のパターンでそれぞれ作成し，最大転送レートで送信したときの測定結果が図 5 である．図 5 より，ソフトウェア測定器は長いパケット長でのみ 10Gbps 転送が出ているのに対して，開発した測定器は全てのパケット長において，10Gbps 転送が可能であることを確認した．

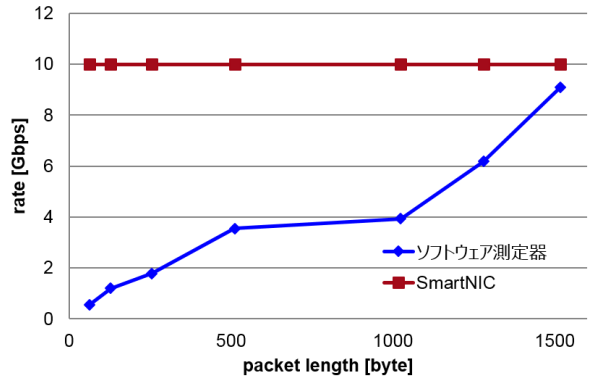

図 5　フルワイヤー性能評価（1 ポート当り）
Figure 5　Evaluation of full-wire performance (per port)

4.2.2 マイクロ秒精度の遅延機能評価

評価環境構成の一例を示す．この例では，測定器の対向側として Open Flow Switch を接続する構成とした．図 6 にその構成図を示す．

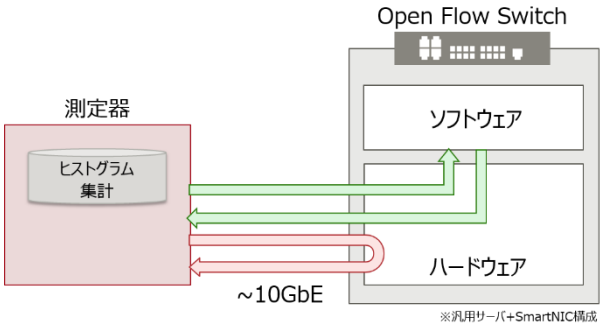

図 6　遅延性能評価の構成図
Figure 6　Evaluation configuration for delay measurement

対向側の Open Flow Switch は，汎用サーバーと SmartNIC で構成されている．4.2.1 と同様に 16,000 フローのテストパケットを 10Gbps で送信する．遅延分布が明示的になるよう，Open Flow Switch 側は，そのテーブル設定において，特定のフローにマッチしたとき，サーバー側にパケットが転送され，ソフトウェアスイッチで折り返す設定とし，その他のフローではハードウェアスイッチで折り返す設定とした．

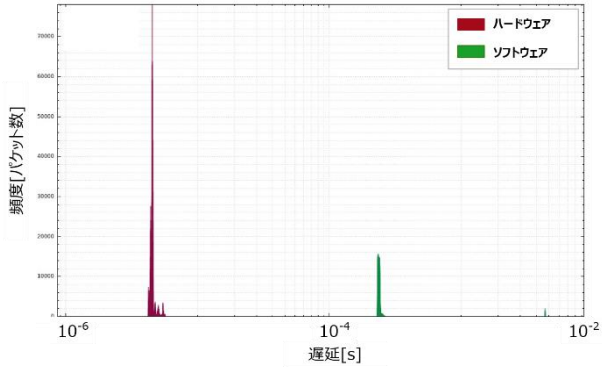

図 7　遅延性能評価（ヒストグラム）
Figure 7　Evaluation of delay measurement

図 7 のヒストグラムは，瞬間的な遅延量のスナップショットであり，横軸に遅延量，縦軸はパケット数となっている．ハードウェアで折り返されたパケットは，1.1~1.2us 程度の遅延分布となったが，ソフトウェアで折り返されたパケットは，0.2~0.3ms 程度の遅延分布が見られた他，0.9~1ms 付近に大きな遅延を持ったわずかな分布も検出された．以上のことから，ハードウェアで処理される遅延の少ないパケットから，ソフトウェアで処理される遅延の大きいパケットも取りこぼすことなく，広い範囲の遅延を検出することが可能であることを確認した．

尚，4.2.1 のループバック接続による環境下では，320ns 程度の遅延量が発生していることを確認しており，本評価結果には，測定器自身が持つ遅延量も含まれていることを考慮されたい．

5. 運用中ネットワークの遅延検証実験

開発した測定器を用いて，実際に運用されているネットワークに接続し，遅延の常時測定を試みたので詳細について述べる．

5.1 検証内容

検証内容は，2018/6/13~2018/6/15 に行われた，ネットワークの相互接続検証において，今回開発した測定器を用いて，遅延測定を実施した．

検証の構成としては，測定器からハードウェアスイッチやハードウェアルーターを経由して，ソフトウェアルーターに接続した構成となっている．測定期間は，稼動を開始した 2018/6/12 13:00 から稼動が終了する 2018/6/15 17:00 までで，測定条件としては，128 バイト，100Mbps，UDP プロトコルのテストパケットを終夜送信し，最小・最大・平均遅延時間および遅延分布を常時測定するといったものである．測定結果の収集にあたっては，オープンソースのログデータ解析/可視化ツールである Kibana やその収集するデータベースである Elasticsearch を活用した．

次節以降，実際の測定結果を述べていく．

5.2 検証結果

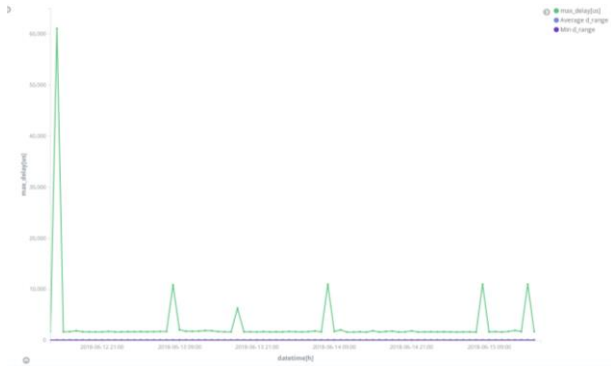

図 8　最大遅延時間の全測定結果
Figure 8　Total result of maximum delay time

図 8 は，期間中に測定した最大遅延時間を全てプロットしたもので，横軸が時刻，縦軸が最大遅延時間となっている．全体の測定結果の特徴としては，通常は 100us を中心とする遅延分布が得られていたが，一時的に大きな遅延量を持ったものが，わずかながらに検出される結果となった．

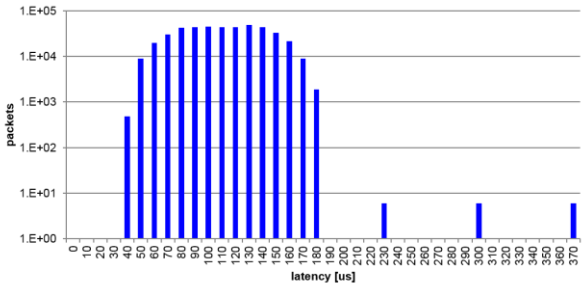

図 9　遅延分布（2018/6/15/15:56:32）
Figure 9　Delay histogram (2018/6/15/15:56:32)

例えば，図 9 で示した 2018/6/15/15:56:32 時点の遅延分布は，最小 42us，平均 110us，最大 370us となった．42us から 180us までは遅延が台形形状で分布していただけでなく，230us，300us，370us にも遅延がわずかに検出された．この台形形状の分布とは異なる位置で見られる遅延は，別の時間帯ではさらに大きく分布して検出されることがあった．それが図 10 である．

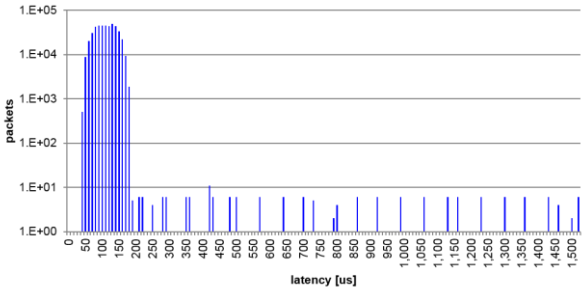

図 10　遅延分布（2018/6/15/15:56:27）
Figure 10　Delay histogram (2018/6/15/15:56:27)

図 10 は 2018/6/15/15:56:27 時点の遅延分布で，先ほどの図 9 の 5 秒前の測定結果である．先ほどと同様に台形形状の遅延分布のほか，220us，250us，290us，350us，360us，420us，430us，と大きな遅延が一定ではない間隔で検出され，最も大きなところでは 1.52ms という遅延が検出された．

さらに，全ての測定結果から，5ms を超える遅延が検出された時刻と最大遅延時間を抽出すると，表 1 となった．

表 1 5ms 以上の遅延を検出した測定時刻
Table 1 measurement date of 5ms~ delay detection

測定時刻	最大遅延時間 [ms]
2018/6/12/14:52	61
2018/6/13/08:32	10.8
2018/6/13/18:05	6.2
2018/6/14/08:32	11
2018/6/15/08:32	11
2018/6/15/15:56	11

表 1 から，毎朝 8 時 32 分頃に 11ms 程度の遅延と，それを除いた，1 日に一回 6ms から 11ms 程度の遅延が発生していたことが分かった．このうち，11ms を検出した時間の遅延分布が図 11 である．

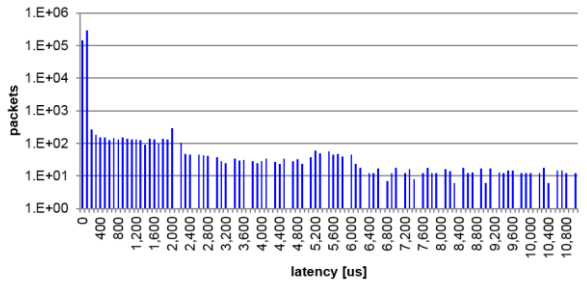

図 11 遅延分布（2018/6/15/15:56:52）
Figure 11　Delay histogram (2018/6/15/15:56:52

図 11 から，ほとんどのパケットは 100us 前後であったが，11ms 程度の遅延まで，ほぼ一様な頻度で発生している

ことがわかった．

5.3 考察

今回検証した遅延測定について，3 つの観点で考察する．

まず，平均遅延時間が 100us 程度であったことについて，これはソフトウェアルーターが汎用サーバーと汎用 NIC で構成されており，汎用 NIC のドライバーが性能を出すことになるため，ある程度パケットがバッファに溜まった状態に達するまで滞留させていた影響と考えられる．この点については，ウェブアプリケーションなど一般のアプリケーションでは影響は小さいが，ストレージや 5G，IoT などの低遅延が求められる一部のサービスでは性能などに影響があると考えられる．

次に，数秒から数十秒間隔で検出された平均 1.5ms 程度の遅延について，これはソフトウェアルーターが動作するサーバー上で発生するソフトウェアによる割り込み処理などが原因であると考えられる．もし，このようなルーターが多段に接続され，それに伴い遅延も蓄積されると，音声や動画など体感的な品質にも影響を及ぼす可能性がある．また，遅延変動が大きいと音声が途切れたり，画面が乱れたりすることも発生しうる．

最後に局所的に検出された 11ms を超える遅延が発生したことについて，発生した日時が毎朝 8 時 30 分頃だったことを考慮すると，ソフトウェアルーターに対し何らかの定期的または手動による処理で負荷が高くなり遅延が増えた可能性が考えられる．等間隔に発生したものではなかったことを考慮すると，ARP キャッシュなどの影響は考えにくい．検証構成に含まれていたスイッチや他のルーターとの間で，経路の切り替えや輻輳などの発生による影響も考えられる．10ms クラスの遅延を検出したことから，リアルタイムアプリケーションに適用した場合，影響が顕著に表れる可能性がある．

6. おわりに

本研究では，汎用サーバーと SmartNIC というハードウェア構成を持ち，オープンソースのソフトウェア測定器におけるパケット処理を SmartNIC にオフロードするというコンセプトにより，サービス運用者が常設かつ常時測定が可能な遅延計測装置を開発した．この測定器は短パケットでフルワイヤー転送が可能で，かつ送信パケット全ての遅延量をサブマイクロ秒の精度で測定可能である．さらに，開発した測定器を実稼働中のソフトウェアルーターを主としたネットワークを検証したことにより，実運用中ネットワークの遅延測定が可能であることを実証しただけでなく，その測定結果から検証したシステムのあらゆる遅延量を検出し，その原因を探るアプローチまで可能とすることを実証した．

本施策によって，多くのネットワーク運用者が遅延によるサービス品質の低下を招くことを回避できるような貢献ができることを信じている．

また，今回実現した遅延測定機能は，我々が提供する測定器(SDT : Software-Defined Tester™)に追加予定である[a].

今回の検証では，実際に運用しているネットワークに対して測定用のパケットを送信する構成であったが，それではユーザーパケットの帯域を消費してしまうため，過度な負荷を与えた測定は困難である．これに対応するため，本稿で考案した測定器の構成を踏襲しつつ，ユーザーパケットの遅延を直接測定可能な装置の実現を，今後の課題と考えている．

謝辞

本研究に関して，京都産業大学の安田先生には多くのアドバイスをいただいたことに感謝する．

参考文献

[1] Myungjin Lee, Nick Duffield, Ramana Rao Kompella. Not All Microseconds are Equal: Fine-Grained Per-Flow Measurements with Reference Latency Interpolation
[2] Ramana Rao Kompella, Kirill Levchenko, Alex C. Snoeren, and George Varghese. Every Microsecond Counts: Tracking Fine-Grain Latencies with a Lossy Difference Aggregator
[3] 大久保 克彦．Software-Defined Tester(SDT)を用いた高精度遅延測定による SDN/NFV 品質向上, https://www.okinawaopenlabs.com/ood/2017/wp-content/uploads/sites/4/2017/12/fujitsu_2.pdf
[4] YASUDA YUTAKA, MIYOSHI TADEFUMI, FUNADA SATOSHI. Development of a FPGA-based measurement tool for latency on OpenFlow switch
[5] Agili-CX 1x40GbE SmartNIC, Netronome. https://www.netronome.com/media/documents/PB_Agilio_CX_1x40GbE.pdf

a) http://www.fujitsu.com/jp/group/fatec/services/platform/software-defined-tester/