4　高速伝送・相互接続技術

4 High-speed Transmission and Interoperability Technology

4-1　JGNⅡを利用した高速トランスポートプロトコルの研究

4-1 Transport Protocols for Fast Long-Distance Networks : Comparison of Their Performances in JGNⅡ

熊副和美　神山勝司　堀　良彰　鶴　正人　尾家祐二
KUMAZOE Kazumi, KOUYAMA Katsushi, HORI Yoshiaki, TSURU Masato, and OIE Yuji

要旨

TCP（Transmission Control Protocol）は、確認応答制御による信頼性の提供 ウィンドウサイズの増減に基づく輻輳制御の実施、実装の容易さなどから、インターネット上の多くのアプリケーションのトランスポートプロトコルとして利用されてきた。しかし TCP は、その輻輳制御アルゴリズムに起因して、長距離・大容量ネットワーク上で効率的な通信を実現することは困難であることが知られている。このため、現在、広域化、広帯域化したバックボーンネットワークを背景として出現してきた Grid、ネットワークストレージ等の新しいアプリケーションに適したトランスポートプロトコル実現のための研究開発が活発化している。本稿では、既存の TCP に替わる新しいトランスポートプロトコルとして提案されている様々な高速トランスポートプロトコルを対象として、JGNⅡ上で実施している性能評価結果を報告する。

The majority of network applications adopt the Transmission Control Protocol (TCP) as the transport layer protocol on IP networks. This is because the TCP has important two functions; one is the error control function providing a reliable, error free data transmission and another is the congestion control mechanism realizing a modest sharing of network resources. However, the current TCP is not always suitable for applications which require highly reliable and high speed transfer of a huge amount of data over long haul networks, such as in the Grid Computing environment. Therefore, various new transport protocols have been proposed with the aim of the efficient use of abundant resources in fast long-distance networks. In this paper, we discuss the results of throughput characteristics of some practical high-speed transport protocols on JGN (Japan Gigabit Network) Ⅱ.

[キーワード]
高速トランスポートプロトコル，長距離・大容量ネットワーク，テストベッド
High-speed transport protocol, Fast long-distance network, Testbed

1　まえがき

メイル、FTP、WEB などインターネット上のアプリケーションの多くはトランスポートプロトコルとして TCP を採用している。しかし、現 TCP（Standard TCP）は、TCP がフロー開始時の

ウィンドウサイズを制限していること（スロースタート）と、フロー制御と輻輳制御が伝搬遅延時間 RTT（Round Trip Time）を単位として行うことから、高遅延、広帯域ネットワークにおいては効率的なデータ伝送の実現が不可能である。例えば Standard TCP を利用した場合、パケットサイズ MTU（Maximum Transmission Unit）が 1500［byte］、RTT が 100［ms］の環境で、コネクション 1 本で 10［Gbps］の帯域幅を使いきるためには、パケット廃棄が起きない状況が 1 時間 40 分継続することが条件となり現実的でない[1]。一方、コアネットワークの大容量化を背景として、Grid に代表されるような長距離高速通信を要求するアプリケーションが現実のものとして登場している。これらのアプリケーションでは　大容量データを高速かつ信頼性を保ちながら伝送することが要求されるため、Standard TCP に替わり効率的なデータ伝送を実現する様々な高速トランスポートプロトコルが提案されている。

本稿では、既存の高速トランスポートプロトコルを対象して JGNⅡ上で実施している伝送実験の結果を紹介する。**2** で実験対象としている高速トランスポートプロトコルを紹介し、**3** で実験環境を、そして **4** でこれまでに得られた実験結果を示す。

2　高速トランスポートプロトコル

実験では Standard TCP の輻輳制御に変更を加えたプロトコルとして、(1) HSTCP[1]、(2) Scalable TCP[2]、(3) FAST TCP[3]、(4) BIC[4]、そして (5) HTCP[5] を取り上げた。また、UDP プロトコルを基礎としたプロトコルとして、(6) UDT[6] プロトコルを対象としている。

(1) - (5) はその動作のために、送信側マシンのみにインストールが必要なプロトコルであるのに対して、(6) は送受マシンの両方にインストールが必要なプロトコルである。これらのプロトコルは、提案者又は研究者によって実装が行われ、その内容がインターネット上で公開されており、本実験でも提供されている実装を利用して実験を実施している。

Standard TCP の輻輳ウィンドウ（cwnd）は、式（1）、（2）に示す AIMD（Additive Increase Multiplicative Decrease）アルゴリズムに従ってその増減が行われる。ここで MSS（Maximum Segment Size）を最大 TCP データ長（MTU から IP ヘッダ長と TCP ヘッダ長を引いた値）とすると Standard TCP は AIMD（a, b）= AIMD（MSS/cwnd, 0.5）と表される。

ACK 受信時：$cwnd = cwnd + a$ ・・・(1)

パケット廃棄検出時：

$cwnd = cwnd(1-b)$ ・・・(2)

HSTCP も cwnd を AIMD アルゴリズムに従って更新するが、そのパラメータは AIMD（a , b）= AIMD（a（cwnd）, b（cwnd））と、その時点での cwnd を基にして決定される。Scalable TCP は cwnd を AIMD アルゴリズムに従って更新し、そのパラメータは AIMD（0.01 , 0.125）である。

BIC-TCP は cwnd の更新を利用可能帯域幅の探索問題としてとらえ、cwnd の更新を行うプロトコルである。すなわち、パケット廃棄が検出されるまでは Additive increase に従って cwnd を増加させる。パケット廃棄が検出されると、その近傍に利用可能な帯域幅があると考え、その後は binary search（二分探索）を実施して利用可能な帯域幅への収束を目指す。

上記三つのプロトコルが、パケット廃棄の検出を契機として cwnd の更新を行うのに対して、HTCP は AIMD 増加パラメータを最後にパケット廃棄を検出してからの経過時間の関数として決定するプロトコルである。すなわち、Δを最後に輻輳が検出されてからの経過時間［s］とすると、AIMD アルゴリズムの a を、a（Δ）で変化させるプロトコルである。

FAST TCP はパケット廃棄だけでなく、パケットがエンド-エンドで経験する遅延（RTT）を利用して、式 (3) に従って cwnd の更新を行うプロトコルである。

RTT を周期として：

$cwnd = cwnd(baseRTT / avgRTT) + \alpha$ ・・・(3)

ここで baseRTT は最小計測 RTT であり、avgRTT は平均 RTT である。α は遅延、帯域幅を基に設定されるパラメータである。

また、UDT プロトコルは、既存の UDP プロトコルに信頼性を付加し、高速データ転送を実現するプロトコルである。UDT は、アプリケーションレベルで動作するプロトコルであり、限られ

たフロー数の環境において利用することを前提としたプロトコルである。パケットペアを利用した利用可能帯域幅の計測を行い、その情報を基に送信レートを適応的に変化させるレートベース制御を行う。

これら提案方式に関してシミュレーションやテストベッド上での評価が実施されている[7][8]。もともと高速トランスポートプロトコルは、豊富にある資源をいかにして効率的に使いきることができるかということを目的として提案されたものであるので、これまでの評価においても、そのような特性の検証を目的としたシナリオにおける評価が主であった。本稿では、そのような各プロトコルの基本的な特性に加えて、次のような観点から、策定したシナリオにおいて実験を実施している。すなわち、将来コアネットワークだけでなく、よりユーザに近いアクセスネットワークも大容量となった場合（＝高速インターネット）、各ユーザは高速トランスポートプロトコルを利用したデータ伝送に興味を持つものと考えられる。実際既にBIC プロトコルは Linux カーネルに組み込んであり（Linux 2.6.7）、設定を行いカーネルを再構築することで、また、アプリケーションレベルで動作する UDT は、開発者のホームページからダウンロードして簡単な手順を踏んでインストールするだけで、すぐに利用できる状況にある。しかし、もともとこれら高速トランスポートプロトコルは、ネットワーク資源が十分ある状況で、いかに効率良く、限られたユーザ数で資源を利用することができるかということを目的として提案されたものであり、そもそも高速インターネットで想定されるような不特定多数のユーザが共存するような状況での利用は想定されておらず、また、そのような状況を仮定した利用における報告もなされていない状況である。そこで本実験では、既存の高速トランスポートプロトコルが高速インターネットで利用された状況を想定して実験を実施している。

高速インターネットのようなネットワークでの利用を想定した場合、大きく次の二つの状況における性能を観測することが重要であると考える。一つめが、各プロトコルフローのネットワーク状況の変化への追従性に関して、すなわちパスの切替え等で生じる遅延や帯域幅の変動が高速トランスポートプロトコルフローに与える影響を調べるためのシナリオを策定した。また、将来のインターネットにおいても現在利用している各種プロトコル（Standard TCP、UDP）は残り、高速トランスポートプロトコルと共存すること、また、アプリケーションによって高速トランスポートプロトコルの使い分けが行われた場合、高速トランスポートプロトコル同士が共存することが考えられる。そこで二つめの項目としてネットワーク上に様々な異なるプロトコルフローが混在する状況を想定してシナリオを策定し、その上での評価を実施している。

3　実験環境

図 1 に実験環境を示す。実験では JGN Ⅱ 国内パスと国際パスの 2 種類を利用しており、各パスの特性を表 1 に示す[1]。実験では図 1 (b) (c) に示すようにネットワークエミュレータを利用して、遅延変動の影響やパスの切替え試験を実施している。

送信側、受信側端末の OS には Linux を採用している。これは、対象としたプロトコルのうち、UDT 以外の実装は Linux Kernel に対するパッチコードとして提供されているからである。また、性能指標としてはスループットを利用している。

1　JGNⅡの利用可能帯域幅は 10［Gbps］であるが本稿では 1［Gbps］環境での実験結果を報告する。また、今後 10［Gbps］環境での実験を予定している。

4　実験結果

4.1　基本特性

図 2 に、JGNⅡ国際回線上に各プロトコルを 1 本設定した時に観測されるスループット特性を示す。

(a) に Standard TCP、HSTCP、Scalable TCP を利用した時のスループット特性を示す。Standard TCP フローは、HSTCP、Scalable TCP を利用した場合と同様のスループットの立ち上がりが観測される。これは txqueuelen 等のパラメータチューニングを行っていることによる。しかし、パケット廃棄が検出され、輻輳回避モードに

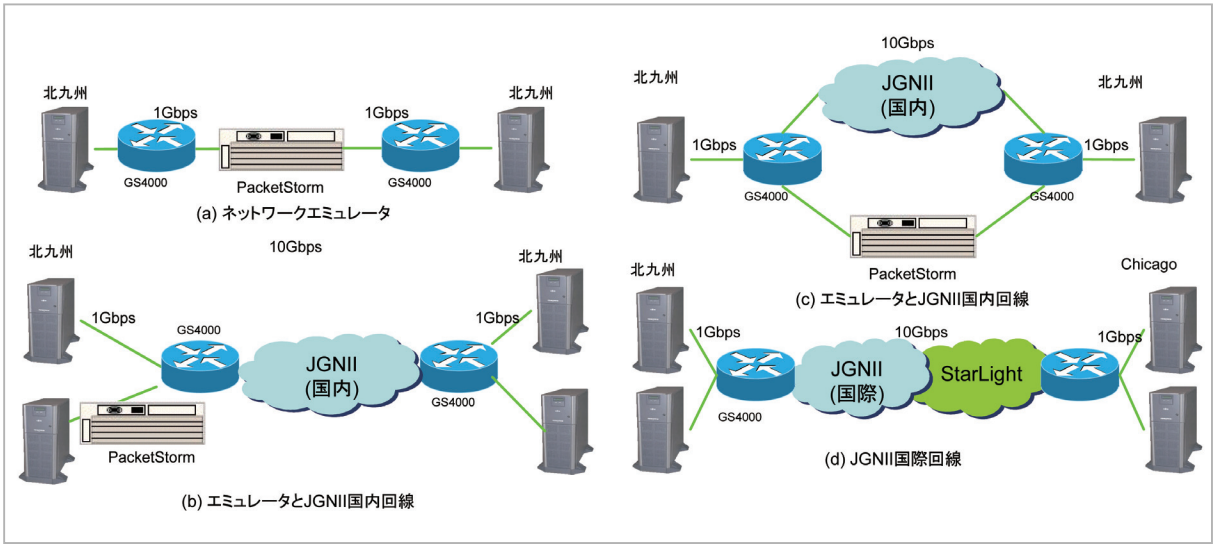


図1 実験ネットワーク構成

表1 実験パスの種別

	RTT[ms]	Bandwidth[Gbps]
Network Emulator	0-10000	1
JGNII国内折り返しパス	38	1
JGNII国際パス	180	1

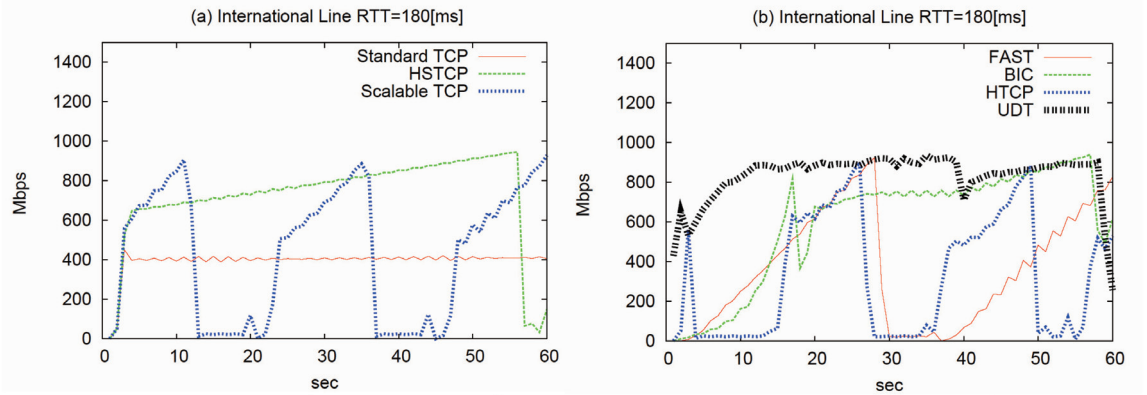


図2 シングルコネクション利用時のスループット特性（JGNⅡ国際回線）

入ると、Standard TCP フローのスループットの増加の割合は頭打ちになっている。これに対して、HSTCP、Scalable TCP フローはパケット廃棄が検出された後もスループットの増加の割合が大きく、最大スループットに到達するまでの時間も短い。(b) に示す FAST、BIC、HTCP、UDT の各プロトコルを利用した場合に見られるスループット特性も Standard TCP フローの特性と比較して立ち上がりが早く、最大値も大きくなっていることが分かる。パケット廃棄が検出されてからの経過時間を基にウィンドウ増加パラメータを決定する HTCP フローと、遅延情報を基にウィンドウ制御を行う FAST フローでは、パケット廃棄が生じた後、スループットが急激に減少しているがその後急激に増加することを繰り返している様子が分かる。BIC を利用した場合は、binary search

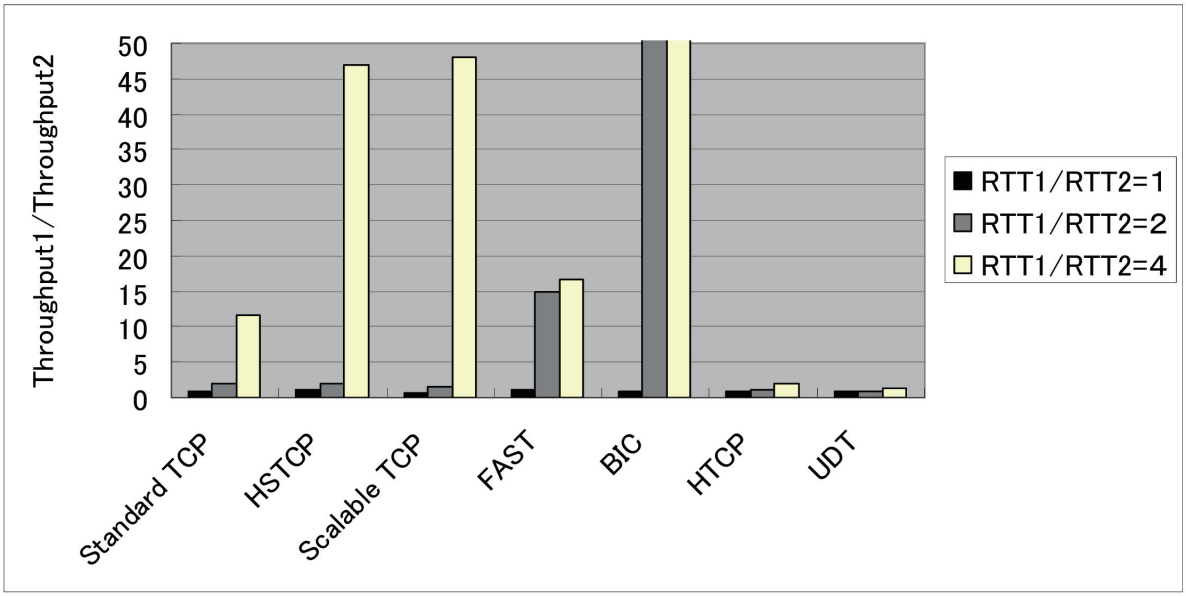


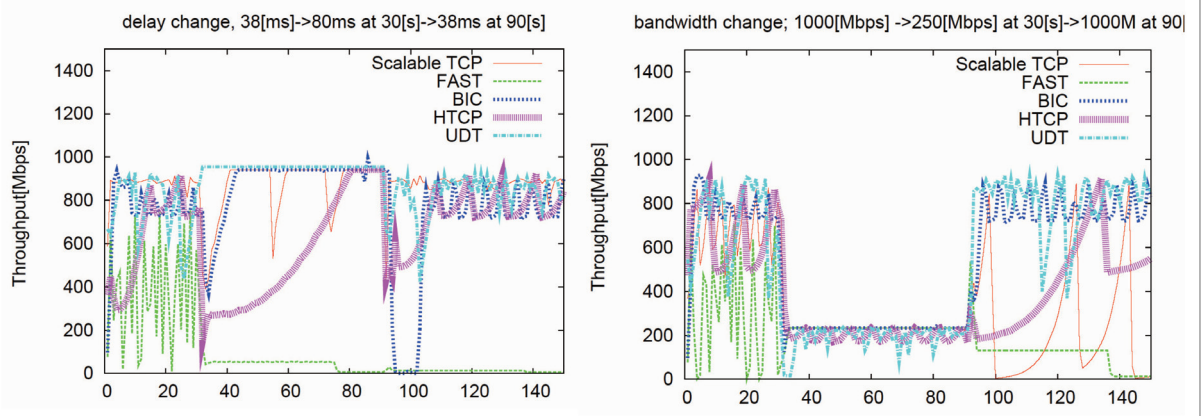
図3　RTTの異なるフローが混在した場合のスループット特性

を利用したウィンドウ制御を行うため、スループット特性にも急激な増減が見られず安定した特性が得られることが分かる。定期的な可用帯域幅の測定を行うUDTプロトコルは、フロー確立直後から高いスループットが観測できているのが分かる。途中何度かスループットの低下が見られるが、これは直前に複数のNACKパケットが受信され、その結果、ウィンドウサイズが縮んでスループットが低下しているためである。

高速インターネットを含むネットワーク上には異なるRTTを持ったフローが混在する。そこで次に、異なるRTTを持つフローが混在した状況で各プロトコルを利用した場合に観測されるスループット特性を図3に示す。図1(b)に示す構成で2本のフローflow1(RTT1＝38[ms])、flow2を設定し、flow2のRTTをRTT1、2RTT1、4RTT1と設定した場合の各フローの平均スループット値の割合を示す。図3より、RTT比が1：1の時はいずれのプロトコルを利用した場合もフロー間のスループット値に違いは見られないが、RTT比1：2の場合はFAST、BICプロトコルフローにおいて、RTT比が1：4の場合はHSTCP、FAST、BIC プロトコルフローにおいて、フロー間のスループット比にRTT比以上の大きな違いが見られることが分かる。これに対してHTCP、UDTプロトコル利用時にはRTT比の違いによってスループット特性に大きな違いが見られないことが分かる。

4.2　状況の変化に対する追従性

次に、通信途中でパスの切替えが生じ、パスの状況が変動した場合を例として、各プロトコルが示す、状況の変化に対する追従性を調べる。図4(a)にパスのend-to-end遅延値が変動した場合の特性を示す(最初の30[s]間は38[ms]、次の1分間は80[ms]、次の1分間は38[ms]と変化)。結果から、遅延の変動は各プロトコルフローのスループット特性に影響を与えているがその程度はプロトコルの種別によって異なることが分かる。UDT、Scalable TCP、BICを利用した場合、他のプロトコルフローと比較して、遅延の変動を受ける影響が小さく、遅延の変動にスループット特性が追従していることが分かる。これに対してHTCPフローは、短い遅延から長い遅延へ変化した場合はスループットレベルが回復するのに時間がかかっているが、長い遅延から短い遅延に変化した場合は迅速にスループット特性が回復していることが分かる。またHSTCPは、パスが切り替わり遅延が変化したことによって、スループットが低下しているが、Standard TCPと比較して迅



(a) 遅延の変動の影響　　(b) 帯域幅の変動の影響

図4 パスの切替えがスループット特性に及ぼす影響

表2 異なる高速トランスポートプロトコルフロー共存時の合計スループット特性

	HSTCP	Scalable	FAST	BIC	HTCP	UDT
HSTCP	720	381	319	535	600	814
Scalable	381	492	383	495	625	825
FAST	319	383	402	301	745	790
BIC	535	495	301	711	287	744
HTCP	600	625	745	287	489	580
UDT	814	825	790	744	580	749

速にスループット特性が回復している。これらに対して、FAST プロトコルを利用した場合、遅延が変動するとスループット特性が大きく劣化している。このようにデータ伝送の途中でエンドエンド遅延が大きく変動することは FAST プロトコルフローの性能に大きな影響を及ぼすことが分かる。また、図 4 (b) に、パスの帯域幅が通信途中で変動した場合の特性を示す (最初の 30 [s] 間は 1 [Gbps]、次の 1 分間は 250 [Mbps]、次の 1 分間は 1 [Gbps])。図より、UDT、BIC は帯域幅が増減した場合もその変化にほぼ遅れることなく追従している。その他の高速プロトコルフローも帯域幅の増減に追従しているが、FAST は、狭い帯域幅から広い帯域幅へ切り替わった後にスループットが回復しない様子が観測される。

4.3 異種プロトコルフロー混在時の特性

次に異なる高速トランスポートプロトコルが共存した場合の特性を示す。表 2 は異なるプロトコルフローを 1 本ずつ、合計 2 本設定した時の合計のスループット特性を示す。表中、色を塗ったプロトコルの組合せは、その合計のスループット値が同種のプロトコルを 2 本設定した場合の合計スループットよりも低くなっている場合を示す。

例えば HSTCP フローを合計 2 本流した場合の合計の平均スループットは 720 [Mbps]、Scalable TCP フローを合計 2 本流した場合の平均スループットは 492 [Mbps] である。これに対して HSTCP と Scalable TCP フローを 1 本ずつ、合計 2 本流した場合の合計の平均スループットは 381 [Mbps] であり、この値は同種のプロトコルフローが 2 本流れている場合の合計スループットのいずれよりも低くなっており、異なるプロトコルフローが共存した場合にお互いにスループット特性に影響を及ぼしていることが分かる。本表から、高速トランスポートプロトコルの組合せによっては合計スループットが低下するものがあり、他のプロトコルフローの性能に対して悪影響を与える場合があることが分かる。

5　むすび

本稿では、JGNⅡの国内回線、国際回線を利用して実施している高速トランスポートプロトコル伝送実験の内容を紹介した。本稿では、特に、既存の高速トランスポートプロトコルを高速インターネットのようなネットワーク上で利用することを想定して実施した実験結果を報告した。この他の結果に関しては[9][10]に紹介している。今後は、最大帯域幅が10[Gbps]の環境における実験を実施する予定である。また、各プロトコルの実装として、共通のネットワークスタック上に各プロトコルを実装したコードが提供されており[8]、このコードを利用した実験も進めることを予定している。

謝辞

JGNⅡを利用した実験に際し、ご協力いただいているJGNⅡ国内NOC、APAN東京XPメンバーの皆様に感謝いたします。また、StarLight上の端末を実験に提供してくださっているイリノイ大UIC NCDMチームの皆様に感謝いたします。

参考文献

1 S.Floyd, "HighSpeed TCP for Large Congestion Windows", RFC 3649. Experimental. December 2003.
2 Scalable TCP : http://www-lce.eng.cam.ac.uk/~ctk21/scalable
3 FAST : http://netlab.caltech.edu/FAST
4 BIC :http://www.csc.ncsu.edu/faculty/rhee/export/bitcp/index.htm
5 HTCP : http://www.hamilton.ie/net/htcp
6 UDT : http://ust.sourceforge.net
7 R.L.Cottrell et al., "Characterization and Evaluation of TCP and UDP-based Transport on Real Networks", PFLDnet2005, Feb. 2005.
8 Experimental Evaluation of TCP Protocols, http://www.hamilton.ie/net.eval
9 K.Kumazoe, K.Kouyama, Y.Hori, M.Tsuru, and Y.Oie, "Transport Protocols for Fast Long-Distance Networks : Evaluation of Their Penetration and Robustness on JGNII", PFLDnet 2005, Feb. 2005
10 熊副和美，神山勝司，堀良彰，鶴正人，尾家祐二，“JGNⅡを利用した高速トランスポートプロトコルの評価”，電子情報通信学会技術研究報告，IN2005-82，pp.37-42，仙台市，2005年9月.

熊副和美（くまぞえかずみ）
拠点研究推進部門北九州 JGNⅡリサーチセンター専攻研究員
ネットワーク性能評価

神山勝司（こうやまかつし）
拠点研究推進部門北九州 JGNⅡリサーチセンター特別研究員
ネットワーク性能評価

堀　良彰（ほりよしあき）
拠点研究推進部門北九州 JGNⅡリサーチセンター特別研究員　博士（情報工学）
情報ネットワーク工学

鶴　正人（つるまさと）
拠点研究推進部門北九州 JGNⅡリサーチセンター専攻研究員　博士（情報工学）
ネットワーク性能計測、ネットワークモデリング、ネットワーク解析

尾家祐二（おいえゆうじ）
拠点研究推進部門 JGNⅡ研究開発プロジェクト総括責任者　工学博士
情報ネットワーク工学