L-014

長距離・広帯域ＩＰネットワークにおける コンテンツのリアルタイム伝送プロトコルの提案

A Study on Real-Time Transport Protocol for Contents on Long Fat IP Networks

金子 豊† 竹内 真也† 山本 真† 藤田 欣裕†
Yutaka KANEKO, Shinya TAKEUCHI, Makoto YAMAMOTO and Yoshihiro FUJITA

1．まえがき

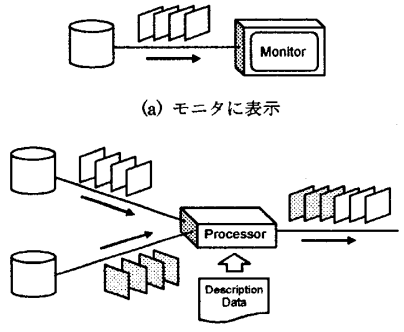
IP ネットワーク上での分散処理を応用した放送局内の制作・送出システムを検討している[1,2]．分散配置されたサーバ内の素材を使って，リアルタイムに再生や加工をするには，ネットワークを介して素材データを安定して伝送できる方式が必要となる．

IP ネットワークで広く使われている TCP は，長距離・広帯域ネットワーク環境では，その帯域を十分に生かすことができない．また，UDP は，送信側の同期でデータ伝送が行われるため，異なるサーバに蓄積した複数の素材をリアルタイムに加工する場合には，素材間の同期を取るための機構が必要となる．

本論文では，長距離・広帯域な IP ネットワーク上で，映像や音声などのストリームデータをリアルタイムに伝送するためのプロトコルとして FMTP(Flow Media Transport Protocol)を提案する．提案する方式は，UDP と TCP を組み合わせたアプリケーションレベルの伝送プロトコルであり，受信側の同期に合わせたストリームデータを伝送するフロー制御機構を持つ．また，１回に限定された再送処理機構を必要に応じて使うことができる．

2．想定するシステム環境

分散処理環境における放送局内システムの検討を進めている．ここでは，ネットワーク上に分散配置された素材や機器を自由に組み合わせて番組制作を行い，番組制作に利用した機器，素材，処理内容を記述データとして保存することで，その記述データがあればどこからでも番組の再生や，修正作業ができる環境を目指している[3,4]．このシステムでは，図１に示すように，ネットワークを介してサーバ内の素材をモニタに表示をさせたり，記述データにしたがって複数の素材をリアルタイムに加工させたりする．



(a) モニタに表示

(b) 記述データに従った加工

図１ コンテンツのリアルタイム伝送の例

†NHK 放送技術研究所（ネットワークシステム），
NHK Science & Technical Research Laboratories

3．TCP および UDP による問題点

リアルタイム伝送に TCP と UDP を使った場合の主な得失を表１に示す．

図１に示したシステムの実装において，フロー制御機構を持つ TCP が使えると，受信側の同期に合わせてデータを受信することができるため，異なるサーバに蓄積された素材であっても，フレーム精度で素材間の同期をアプリケーションレベルで簡単に行うことが出来る．しかし，TCP による伝送は，RTT(Round Trip Time)が小さく，またパケットロスが十分に小さなネットワーク環境下であれば，非圧縮のハイビジョン映像など大量のデータをリアルタイムに伝送することが可能であるが，RTT が大きな長距離・広帯域なネットワーク環境では，その帯域を十分に生かした伝送が困難なことが指摘されている[5,6]．

これは，TCP の輻輳回避フェーズの機構が主な原因である．広く実装に使われている TCP Reno[7-9]では，AIMD(Additive Increase Multiplicative Decrease)アルゴリズムとよばれる方式がとられる．この方式では，送信側の輻輳ウインドウサイズは，1RTT で 1MTU 分のウインドウサイズしか増加しないため，RTT が大きな場合にはウインドウサイズの増加が非常に遅く，また，パケットロスがあると，ウインドウサイズを半分にまで落とすという変化幅が大きな構造となっている．そのため，一般に使われている TCP の実装では長距離・広帯域なネットワーク環境ではスループットが十分に出ない結果となる．

長距離・広帯域ネットワーク下でのスループットの性能改善方法として，AIMD を改良した方式[10,11]，RTT の測定値から輻輳を予測する方式[12,13]，UDP に信頼性を追加した方式[14]などが提案されている．

表１ TCP と UDP の得失

	長 所	短 所
TCP	・フロー制御による受信側マスターの伝送 ・セルフクロッキングによる安定したストリーム伝送	・長距離・広帯域ネットワークでスループットが出ない ・再送処理機構が伝送のリアルタイム性を乱す
UDP	・伝送処理遅延が少ない ・マルチキャストによる１対Ｎの配信が可能	・送信側がマスターになるため、受信側の同期にあわせる場合は、別途その機構が必要 ・再送が必要な場合は別途その機構が必要

4．提案方式

提案する FMTP では，受信側が送信する同期信号を兼ねた ACK パケットに同期して，送信側がデータパケットを送信する．ACK パケットの送信には TCP を，データパケ

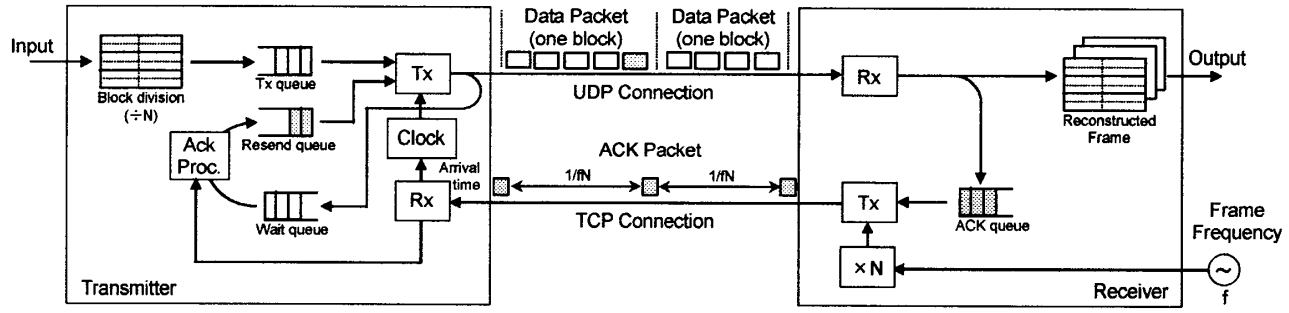


図２ FMTPによる送受信装置の構成

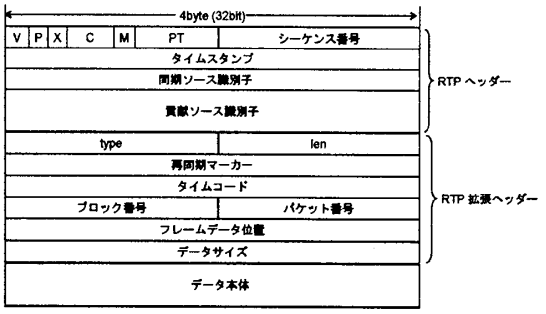


図３ データパケットの構造

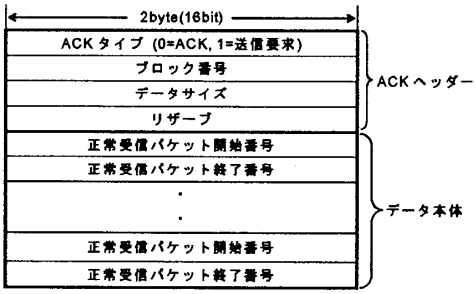


図４ ACKパケットの構造

ットの送信にはUDPを用いる．FMTPの送受信装置の構成を図２に示す．

送信するフレームデータはあらかじめ決められたブロック分割数(N)に分割し，１つのACKパケットに対して１ブロック分のデータをパケット化して送信する．ブロックの送信間隔はACKパケットの到着間隔から決定する．５章で述べる実装では単純移動平均により送信間隔を決定した．

送信するデータパケットは図３に示すRTPパケット構造[15,16]とし，FMTPで利用するヘッダーデータはRTPの拡張ヘッダー内に格納した．各パケットは（タイムコード，ブロック番号，パケット番号）の組により識別される．タイムコードはフレーム単位にあらかじめ付けられた番号，ブロック番号はブロック分割時に付けるシーケンシャルな番号，パケット番号は同一ブロック内のパケットを識別するためのシーケンシャルな番号である．また，拡張ヘッダーには，受信側で容易にフレームを再構成できるように，フレーム先頭からのバイト位置をヘッダーに格納している．また，今回は利用していないが，再同期マーカー，データ本体のバイトサイズを拡張ヘッダーに格納した．

FMTPでは再送は１パケットあたり１回に限定し，カレントのデータパケット同様，１つのACKパケットに対し１ブロック内の再送パケットを送信する．受信側はデータパケットを受信すると，各パケットのヘッダーの内容から，パケットロスを判定し，１ブロック毎に図４に示すACKパケットを作成する．再送が必要ない場合は，すべてのパケットを受信したものとしてACKパケットを送信する．転送開始時やデータパケットの到着が遅れた場合など，ACKパケットの送出タイミング時に送出するACKパケットがない場合には，代わりに送信要求パケットを送出する．

なお，FMTPは輻輳制御機構を持たないため，あらかじめQoSなどで帯域が保証されたネットワークでの利用を想定している．

5. 実験結果と考察

提案方式の性能を評価するため，FMTPをLinuxに実装し，1GEthernet (MTU=1500byte)環境で，ネットワークエミュレータを介した伝送実験を行った．実験環境を図５に示す．比較対象としてLinuxに標準実装のTCP[7,17]と，Linuxに実装されたScalable TCP[10,18]，FAST TCP[13,19]，UDT[14,20]を用いた．

Scalable TCP(STCP)は，TCP RenoのAIMDに変更を加えた方式であり，TCP Renoに比較して，積極的にウインドウサイズが更新される．FAST TCP(FTCP)は測定したRTTから輻輳を予測し，RTT毎にウインドウサイズを更新する方式である．UDTはUDPによるデータ転送に，受信側からの制御情報の通信プロトコルを追加することで，信頼性のあるデータ転送を実現した方式である．

伝送するデータとして表２に示すビットレートの異なる３種類の非圧縮映像データを用いた．受信側では，データを受信するとともに，映像出力ボードを介して映像モニタに表示し，目視による映像データを確認した．パケットロスは，ロスなし，1/10000pkt (BER:8.3×10^{-9})，1/1000pkt (BER:8.3×10^{-8})の３種類，RTTを0, 10msec, 100msecの３種類で実験を行った．

データパケットサイズは，SD，HD-mono画像ではフラグメンテーションの起こらない1500byte，HD画像では，RTPパケット化のオーバーヘッドを減らすために24000byteとした．また，予備実験によりTCPではRTT=200msecで2msec程度まで定期的なACKパケットの送受信が可能なことを確認したため，ブロック分割数をN=15とした．この場合，フレームレートが29.97Hzでは，ACKパケットの送出間隔は約2.2msecとなる．

表３に各プロトコルの平均スループットと，映像モニタでの目視による出力結果を示す．図６，７に(RTT=10msec,

表 2 実験に用いた動画像の種別

Type	Size	Rate
SD	720x508pel, 8bit, 4:2:2, 29.97Hz	175Mbps
HD-mono	1920x1080pel, 8bit, 4:0:0, 29.97Hz	497Mbps
HD	1920x1035pel, 8bit, 4:2:2, 29.97Hz	953Mbps

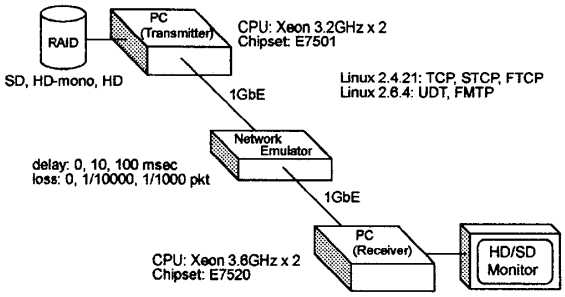


図 5 実験システムの構成

loss=0)のときと，(RTT=100msec，loss=10^{-4})のときのスループットを示す．

TCP ではロスがなければ RTT=10msec までは HD-mono を送受信することができるが，ロスがある場合には困難であることが分かる．これは文献[6]の結果とほぼ一致する．

STCP, FTCP はいずれもオリジナルの TCP に比較して改善効果が見られたが，帯域をフルに使う HD の伝送では，十分な伝送レートを得ることが出来なかった．また，平均値のスループットは間に合っているものの，ビデオ映像としてはフレームがスキップする場合があった．

UDT は STCP，FTCP に比較して改善効果が大きく，ロスがなければ HD 映像の伝送でほぼ帯域を使い切る性能が得られた．但し，ロスがある場合には，スループットが低下する結果となった．また，HD-mono の伝送では，バースト的な送信が観測された．

FMTP では RTT=100msec 以内，ロス率 10^{-3} 以内ではほぼ問題なく送受信できることが確認できた．約 23 秒毎にスループットが落ち込んでいるが，これは ACK パケットの送信間隔が一時的に長くなるためである．その原因は調査中であるが，実装上の問題と考えられる．

FMTP の送信側に到着する ACK パケットの到着間隔のヒストグラムを図 8 に示す．図中の平均値は，ACK パケットの到着間隔の単純移動平均(標本数 10000)の平均値と標準偏差を表している．RTT が大きく，パケットロスが大きくなるほど，0 秒で到着するパケットが増加，すなわち，ACK パケットの到着間隔がバースト的になるが，移動平均をとることで，ほぼその影響を取り除くことができることが分かる．

6. まとめ

IP ネットワーク上で映像や音声などのストリームデータを安定してリアルタイム伝送するためのプロトコルを提案した．実験の結果，高遅延の環境であっても安定してストリーム伝送を行うことが確認できた．今後は，記述データに従った複数パスのストリームデータの同期再生処理，10GEthernet 環境での評価実験などを行う予定である．

参考文献

[1] 山本真, 竹内真也, 金子豊, "ネットワーク利用制作・送出システムに向けて," NHK 技研 R&D, No.80, pp.30-37, Mar. 2001.
[2] 金子豊, 竹内真也, 山本真, "ネットワーク利用制作・送出システム," NHK 技研 R&D, No.90, pp.48-55, Mar. 2005.
[3] 金子豊, 竹内真也, 山本真, 藤田欣裕, "分散処理におけるオブジェクト合成のためのマクロ記述方法および GUI の検討," 情処技報告, 2004-AVM-44, pp.97-102, 2004.
[4] S. Takeuchi, Y. Kaneko, M. Yamamoto, M. Shibata, A. Narumi, S. Sunasaki, "A Program Production System using ID and File-data over IP Networks," SMPTE Motion Imaging Journal, Vol.114, No.3, 2005.
[5] 鶴正人, 熊副和美, 尾家祐二, "長距離高速通信のための TCP 性能改善技術の動向," 情報処理, Vol.44, No.9, pp.951-957, Sep, 2003.
[6] 山本真, 竹内真也, 金子豊, "Gigabit Ethernet を用いた非圧縮 HDTV の TCP 転送性能に関する検討," 信学総大, B-6-145, 2004.
[7] M. Allman, V. Paxson, W. Stevens, "TCP Congestion Control," RFC2581, Apr. 1999.
[8] 村山公保, 西田佳史, 尾家祐二, "トランスポートプロトコル," 岩波書店, 2001.
[9] M. Hassan, R. Jain, "High Performance TCP/IP Networking," Pearson Prentice Hall, 2004
[10] T. Kelly, "Scalable TCP: Improving Performance in Highspeed Wide Area Networks," ACM Computer Communication Review, 33(2), pp.83-91, Apr, 2003.
[11] S. Floyd, "HighSpeed TCP for Large Congestion Windows," RFC3649, Dec. 2003.
[12] L. Brakmo, S. O'Malley, L. Peterson, "TCP Vegas: New techniques for congestion detection and avoidance," Proceedings of the SIGCOMM '94, pp.24-35, Aug. 1994.
[13] C.Jin, D. X. Wei, S. H. Low, "FAST TCP: Motivation, Architecture, Algorithms, Performance," Proceedings of IEEE Infocom'04, Mar. 2004.
[14] Y. Gu, X. Hong, R. L. Grossman, "Experiences in Design and Implementation of a High Performance Transport Protocol," Proceedings of the ACM/IEEE SC2004 Conference, 2004.
[15] H. Schulzrinne, S. Casner, R. Frederick V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC1889, Jan. 1996.
[16] C. Perkins, "RTP," Addison Wesley, 2003.
[17] K. Wehrle, F. Pahlke, H. Ritter, D. Muller, M. Bechler, "The LINUX Networking Architecture," Pearson Prentice Hall, 2005.
[18] Scalable TCP. http://www-lce.eng.cam.ac.uk/~ctk21/scalable/
[19] FAST Homepage. http://netlab.caltech.edu/FAST/
[20] Sabul FAQ. http://www.ncdm.uic.edu/sabul.html/

表 3 実験結果（スループット[Mbps]）

		TCP						Scalable TCP						FAST TCP						UDT						FMTP(proposed)					
	LOSS / RTT	0		10^{-4}		10^{-3}		0		10^{-4}		10^{-3}		0		10^{-4}		10^{-3}		0		10^{-4}		10^{-3}		0		10^{-4}		10^{-3}	
SD	0	○	174	○	180	○	180	△	175	○	180	○	180	○	180	○	180	○	180	○	356	○	350	○	346	○	186	○	186	○	187
	10	○	180	×	97	×	29	○	180	○	180	×	54	○	180	○	180	○	180	○	365	○	355	○	353	○	187	○	184	○	188
	100	×	72	×	10	×	3	×	72	×	26	×	6	×	72	×	27	×	16	○	311	○	301	○	302	○	187	○	188	○	187
HD-mono	0	○	511	○	510	○	511	△	506	○	511	○	510	○	510	○	511	○	511	○	551	○	541	○	520	○	520	○	520	○	520
	10	○	511	×	93	×	30	△	504	×	240	×	58	○	511	○	511	×	257	○	553	○	547	○	515	○	519	○	520	○	520
	100	×	240	×	10	×	3	△	488	×	41	×	6	△	499	×	141	×	16	○	509	△	535	○	520	○	520	○	520	○	520
HD	0	×	882	×	704	×	558	×	897	×	663	×	580	×	691	×	678	×	662	△	957	×	380	×	247	○	970	△	968	△	970
	10	×	837	×	89	×	29	×	963	×	197	×	48	×	779	×	616	×	253	△	964	×	382	×	242	○	969	○	970	×	970
	100	×	212	×	10	×	3	×	827	×	43	×	7	×	740	×	142	×	16	△	951	×	783	×	301	○	971	○	974	△	973

記号はモニタ表示の結果（○：正常表示、△：たまにフレームスキップ、×：まにあわない）

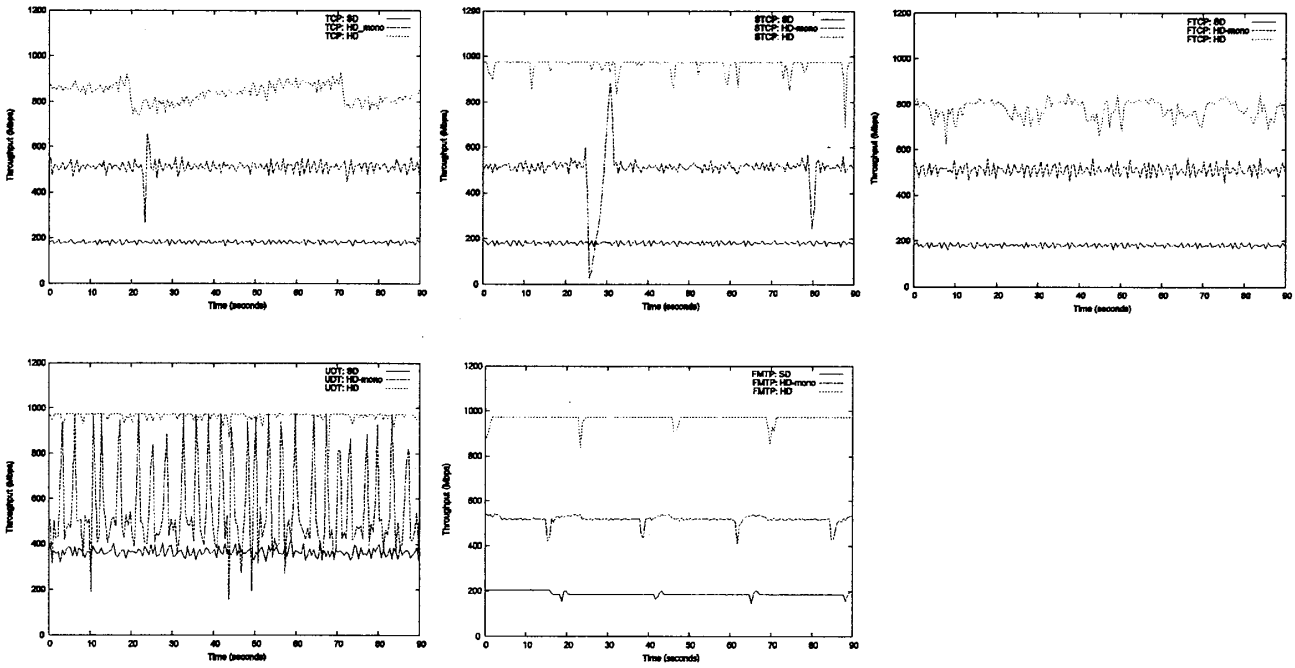


図 6 スループットの比較（RTT=10msec, loss=0）

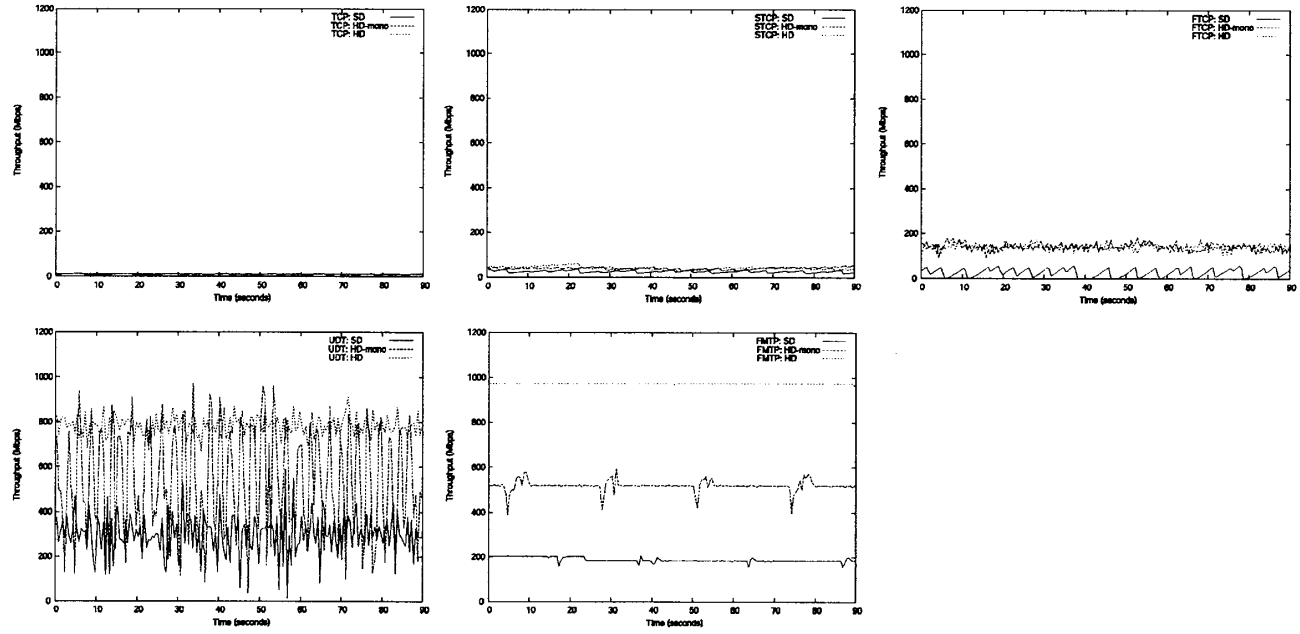


図 7 スループットの比較（RTT=100msec, loss=1/10000pkt）

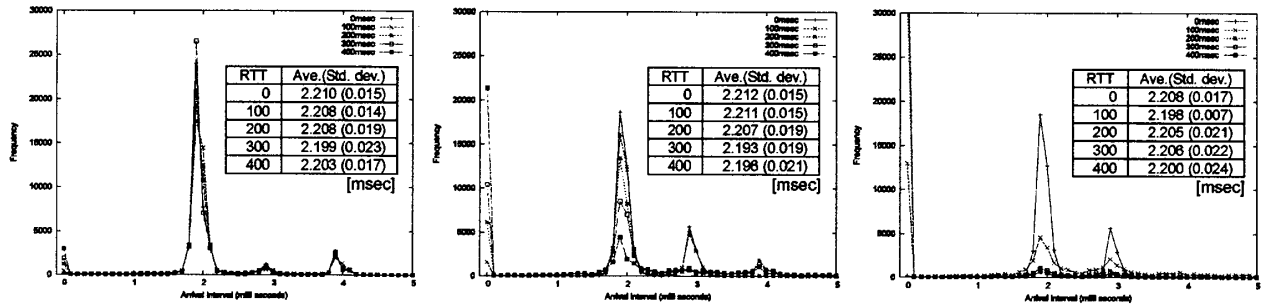


図 8 ACK パケットの到着時間間隔（左から loss=0, 1/10000pkt, 1/1000pkt）