エンド・エンド測定に基づく アドミッション・コントロール方式

(課題番号 14350199)

平成14年度~平成17年度科学研究費補助金 (基盤研究(B)) 研究成果報告書

新潟大学附属図書館

1060034243

平成18年3月

研究代表者 間 瀬 憲 (新潟大学自然科学系教授)



近年の情報通信分野における発展は目覚しいものがあり、特にインターネットは日々 我々の生活に無くてはならないものへと成長を続けている。このインターネットの成長は アクセス回線のブロードバンド化、常時接続の一般化により更に加速している。

こうした中、従来は電話回線を用いることで提供されてきた音声通信を、IP ネットワークを通して行う VoIP(Voice over IP)が大変注目を集めている。VoIP とは、アナログの音声をデジタル化し、IP ネットワークを経由した音声通信を実現する技術である。

VoIP により、従来の電話サービスで用いられている回線交換方式よりも高い回線利用効 率を実現できる。しかし、IP ネットワークは本来、ベストエフォート型のトラヒックを前 提として設計されているため、リアルタイムに通信を行う必要のある VoIP では、通話品質 を一定に保つことが難しい。ネットワークが混雑した状況では、パケット損失、遅延、ゆ らぎの影響により、通話品質の低下が起こる恐れがある。この通話品質低下を防ぐ方法と して、IntServ/RSVP(Integrated Service / Resource reSerVation Protocol)等のシグナリングプロト コルを用いる方法がある。通話を開始する前に、サービスに必要なリソースをルータなど の中継機器が予約するため、通話時の音声品質が保証される。この方法では、ネットワー ク内の中継機器が全て、このプロトコルをサポートする必要があり、また、フロー毎の資 源予約情報を全て保持する必要があるため、スケーラビリティの面で難がある。もう 1 つ のアプローチとして、エンドポイントアドミッションコントロールがある。エンドポイン トアドミッションコントロールでは、実際に通信を行う両エンドポイントのみが制御を行 うため、中継機器に特別な機能を必要とせず、スケーラビリティに優れている。そこで、 本研究ではエンドポイントアドミッションコントロールによる品質保証に焦点を当てて研 究を行い、具体的なアドミッションコントロールの仕組み、ネットワーク設計法、エンド・ エンドのネットワーク・アプリケーション品質管理システムなど様々な成果を得た。

本研究の過程でお手伝い頂いた新潟大学工学部情報工学科阿達透技官、学生諸氏に感謝の意を表します。

研究代表者 間 瀬 憲 一

研究組織

研究代表者 間瀬憲一(新潟大学・自然科学系・教授) 研究分担者 柄沢直之(新潟大学・自然科学系・助手) (研究協力者 塩田茂雄(千葉大学・工学部・助教授))

研究経費

平成 14 年度		10,200	千円
平成 15 年度	a"	1,600	千円
平成 16 年度		1,400	千円
平成 17 年度	•	1,600	千円
計		14,800	千円

研究発表

(1) 学会誌等

- [1] 間瀬憲一: "インターネットにおけるスケーラブルなアドミッションコントロール方式," 電子情報通信学会誌, vol. 85, no. 9, pp. 655-661, Sept. 2002.
- [2] 間瀬憲一, 栗林孝行, 津野昭彦: "QoS 統計データを利用した動的サーバ選択法," 電子情報通信学会論文誌(B), vol. J86-B, no. 3, pp. 1-12, March 2003.
- [3] A. Bilhaj and K. Mase: "Endpoint Admission Control Enhanced Systems for VoIP Networks," IEICE Transaction on Communications, vol. E87-B, no. 4, pp. 948-956, April 2004.
- [4] K. Mase: "Toward Scalable Admission Control for VoIP Networks," IEEE Communications Magazine, vol. 42, no. 7, pp. 42-47, July 2004.
- [5] 八木敬宏, 塩田茂雄, 間瀬憲一: "ボトルネックリンク速度推定ツールの提案と精度検証," 電子情報通信学会論文誌(B), vol. J87-B, no. 10, pp. 1636-1647, Oct. 2004.

(2) □頭発表

国際会議(International Conferences)

- [1] K. Mase and Y. Toyama: "End-to-End Measurement Based Admission Control for VoIP Networks," 2002 IEEE International Conference on Communications (ICC2002) Conference Proceedings, vol. 2, pp. 1194-1198, May 2002.
- [2] X. Kong and K. Mase: "Dynamic Routing with Endpoint Admission Control," 2003 IEEE International Conference on Communications (ICC2003) Conference Proceedings, vol. 3, pp. 1728-1732, May 2003.
- [3] K. Mase and H. Kobayashi: "An Efficient End-to-End Measurement Based Admission Control for VoIP Networks," 2004 IEEE International Conference on Communications (ICC2004) Conference Proceedings, vol. 4, pp. 1877-1881, June 2004.
- [4] S. Shioda and K. Mase: "A New Approach to Bandwidth Requirement Estimation and its Accuracy Verification," 2004 IEEE International Conference on Communications (ICC2004) Conference Proceedings, vol. 4, pp. 1953-1957, June 2004.
- [5] Shigeo Shioda, Takahiro Yagi and Kenichi Mase: "A New Approach to the Bottleneck Bandwidth Measurement for an End-to-End Network Path," 2005 IEEE International Conference on Communications (ICC2005) Conference Proceedings, vol. 1, pp. 59-64, May 2005.
- [6] Shigeo Shioda and Kenichi Mase: "Performance Comparison between IntServ-Based and DiffServ-Based Networks," 2005 IEEE Global Telecommunications Conference (GLOBECOM 2005), vol. 1, pp. 529-534, Dec. 2005.

研究会

- [1] 孔翔, 間瀬憲一: "VoIPネットワークにおける迂回を考慮したエンドポイントアドミッションコントロール," 電子情報通信学会技術研究報告, CQ2002-60, pp. 53-57, July 2002.
- [2] 小林尚人, 間瀬憲一: "VoIPネットワークにおけるエンドポイントアドミッションコントロールの性能評価,"電子情報通信学会技術研究報告, CQ2002-61, pp. 59-64, July 2002.
- [3] 戸嶋宏樹, 間瀬憲一, 柄沢直之: "VoIP ネットワークにおけるアドミッションコントロール方式の実験的性能評価," 電子情報通信学会技術研究報告, CQ2002-88, pp. 23-28, Sept. 2002.
- [4] 八木敬宏, 塩田茂雄, 間瀬憲一: "ボトルネックリンク速度推定ツールの提案と精度検証," 電子情報通信学会技術研究報告, IN2002-199, pp. 47-52, Feb. 2003.
- [5] 八木敬宏, 塩田茂雄, 間瀬憲一: "UDP ペアパケットを用いたボトルネックリンク速度推定ツールの提案," 電子情報通信学会技術研究報告, IN2003-92, pp. 13-18, Oct. 2003.
- [6] 小林尚人, 間瀬憲一: "VoIP網におけるエンドポイントアドミッションコントロールとそのネットワーク設計方法の評価," 電子情報通信学会技術研究報告, NS003-225, pp. 5-8, Feb. 2004.
- [7] 鈴木裕和,高橋拓也,金子孝二,三浦輝典,間瀬憲一: "VoIP ネットワークにおける End-to-End QoS 監視システムの構築,"電子情報通信学会技術研究報告, CQ2005-10, pp. 11-14, April 2005.
- [8] 須田和彦, 鈴木裕和, 間瀬憲一: "VoIP ネットワークにおける過負荷リンクを考慮したエンドポイントアドミッションコントロールの性能評価," 電子情報通信学会技術研究報告, CQ2005-11, pp. 15-20, April 2005.
- [9] 高橋拓也, 鈴木裕和, 間瀬憲一: "VoIP ネットワークにおける End-to-End QoS 監視システムの構築," 電子情報通信学会技術研究報告, TM2005-52, pp. 61-66, Jan. 2006.
- [10] 須田和彦, 鈴木裕和, 間瀬憲一: "VoIP ネットワークにおける複数迂回経路を考慮したエンドポイントアドミッションコントロールの性能評価," 電子情報通信学会技術研究報告, TM2005-53, pp. 67-72, Jan. 2006.

全国大会

- [1] 戸嶋宏樹, 間瀬憲一, 柄沢直之: "VoIP ネットワークにおけるプローブフローを用いたアドミッションコントロール方式の実験的検討," 2002 年電子情報通信学会ソサイエティ大会, B-11-4, Sept. 2002.
- [2] 鈴木裕和, 間瀬憲一: "MOS 評価のための VoIP 実験システムの構築," 2003 年電子情報通信学会総合大会, B-11-10, March 2003.
- [3] 小林尚人, 間瀬憲一: "VoIPネットワークにおける確率的エンドポイントアドミッションコントロール," 2003 年電子情報通信学会総合大会, B-11-11, March 2003.

- [4] 鈴木裕和, 浅野博文, 大和田泰伯, 間瀬憲一: "Bluetooth 位置検出を用いた VoIP サービス に関する考察," 2003 年電子情報通信学会ソサイエティ大会, B-6-7, Sept. 2003.
- [5] A. A. Bilhaj, K. Mase: "Efficiency Evaluation for VoIP Networks with Enhanced Endpoint Admission Control," 2003 年電子情報通信学会ソサイエティ大会, SB-10-12, SE-22, 23, Sept. 2003.
- [6] 小林尚人, 間瀬憲一: "エンドポイントアドミッションコントロールを用いた VoIP 網のエンド-エンド品質の評価," 2003 年電子情報通信学会ソサイエティ大会, B-11-17, Sept. 2003.
- [7] 鈴木裕和, 間瀬憲一: "ユーザ品質評価のための VoIP ネットワーク品質管理手法の提案," 2004 年電子情報通信学会総合大会, B-11-19, p. 512, March 2004.
- [8] Lin Lu, Kenichi Mase: "An Evaluation of End-to-End Measurement Based Admission Control with Probability for Video Streaming," 2004 年電子情報通信学会総合大会, B-7-42, p. 251, March 2004.
- [9] 須田和彦, 間瀬憲一: "VoIP ネットワークにおける過負荷状態を考慮した EMBAC 方式," 2004 年電子情報通信学会ソサイエティ大会, B-11-16, p. 325, Sept. 2004.
- [10] 高橋拓也, 鈴木裕和, 金子孝二, 間瀬憲一: "VoIP ネットワークの End-to-End 品質測定手法の実現," 2005 年電子情報通信学会総合大会, B-11-31, Sept. 2005.

研究成果

1. エンド・エンド測定に基づくアドミッションコントロール方式

1.1 はじめに

この章では、音声品質を良好に保つためのアプローチである、エンド・エンド測定に基づくアドミッションコントロール方式(EMBAC: End-to-end Measurement Based Admission Control)について説明する。

IP 網には常にさまざまな情報が転送されておりサービスに必要な帯域が保証されていない。もし、ネットワークが混雑している時に VoIP(Voice over IP)による通話を行おうとすると、音声パケットにパケット損失、遅延、ゆらぎが生じてしまい、音声品質の劣化が起こる恐れがある。どんなにネットワークが混雑していても既存の VoIP 端末は通話を開始してしまうため、音質が悪く快適な通話を行えない場合がある[1][2]。このような問題を解決するためアドミッションコントロール方式では、新たな呼が生起した場合にネットワーク上のリソースの使用状況に応じて接続に制限を掛ける。これによって、低音質の通話を回避することができ、また一度接続を許可された呼の音声品質を良好に保つことが可能となる[3]-[5]。

アドミッションコントロール方式には、様々な方式が提案されているが、本稿では EMBAC に着目する[6]。 EMBAC では、通話を開始する前にエンドポイント同士で試験トラヒック(プローブフロー)の送受信を行い、ネットワークリソースの状態を把握する。プローブフローが VoIP フローを圧迫しないよう、優先度を低くする。その結果、通話を行うのに十分なリソースがある場合は呼接続が開始される。リソースが足りないと判断した場合、呼接続を中止する[7][8]。 EMBAC の流れを図 1-1 に示す[9]。

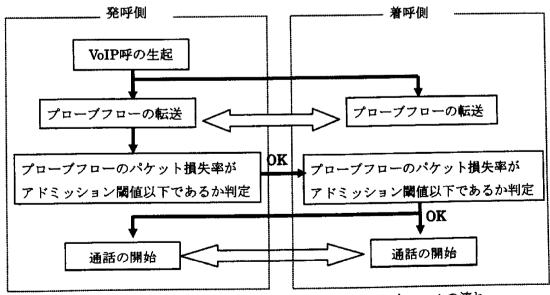


図 1-1 エンド・エンド測定に基づくアドミッションコントロールの流れ

アドミッションコントロールとは、プローブフローのパケット損失率によりネットワーク状態を推測し、VoIPトラヒックの量を制限する方式と言える。しかしプローブパケット損失率と平均 VoIPパケット損失率の間に線形的な相関が成立せず、制御感度が低いため、VoIPフローのトラヒックを正確には制御できないという問題がある。確率アドミッションコントロール方式(EMBAC-P)とは、プローブパケット損失率の他に、確率による呼接続制御を行い、アドミッションコントロールの制御感度を向上させる方式である。これに対して確率を使用しない確定方式をEMBAC-Dと呼ぶ。

1.2 確率アドミッションコントロール方式 (EMBAC-P)

EMBAC-P では図 1-2 に示すようにアドミッション閾値による判定で呼損と判断された呼をある一定の確率 (アドミッション確率) により受け入れ許可とする。これまで述べてきたアドミッション閾値だけでの制御を行った場合と比べ、過制御を減らす事ができる[10]。

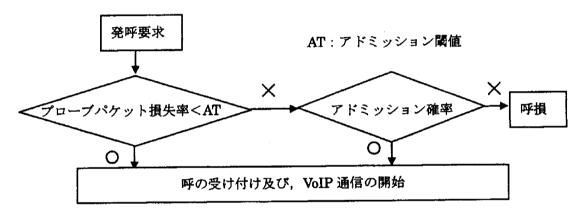


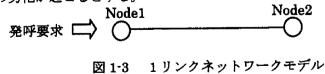
図 1-2 確率アドミッションコントロール方式

1.3 シミュレーションによる評価

1.3.1 シミュレーションモデルと条件

計算機シミュレーションにより EMBAC の性能評価を行った。シミュレータとして ns-2 を使用した[11]。

シミュレーションには図 1-3 に示す 1 リンクネットワークモデルを使用する。これは、ボトルネックリンクをモデル化したものである。音声通信はこのボトルネックリンクによる影響で QoS の劣化が起こるとする。



シミュレーション条件を表 1-1 に示す。想定する音声コーデックは ITU-T 標準の G.729 とする。このためエンドポイントでは音声信号が 8kbps で符号化される。符号化データに RTP/UDP/IP ヘッダ計 40[byte]をつけて 20[msec]の送信間隔でパケットは送信される。 20[msec]の送信間隔でパケットを生成するのでペイロードサイズは 20[byte]となり、最大 60[byte]のパケットが送信される。無音圧縮を行うため、無音区間ではパケットを送信しない。呼の到着率はポアソン分布に従い、呼の保留時間は指数分布に従うと仮定する。

表 1-1 シミュレーション条件

呼と VoIP フロー条件

VoIP パケットサイズ	41[byte]~60[byte]	
VoIP パケット送信間隔	20[msec]	
有音率	30[%]	
有音区間の平均長	3[sec]	
平均保留時間	180[sec]	

ネットワーク条件

- · · · · · · · · · · · · · · · · · · ·	
各ノードでの VoIP 呼のバッファサイズ	注①
各ノードでのプローブフローのバッファサイズ	VoIP パッファサイズの半分
各ノードでの VoIP 呼への保証帯域(割合)	100[%]
各ノードでのプローブフローへの保証帯域(割合)	0[%]

プローブ条件

プローブパケットサイズ	40[byte]
プローブパケット送信間隔	20[msec]
プローブパケット送信時間	1[sec]

シミュレーション条件

データ取得開始時間	3600[sec]
シミュレーション終了時間	10800[sec]

注① EMBAC方式はVoIP通信のパケット損失について保証する方式であり遅延についての保証は行わない。音声通信を行う上で問題となるパケットの遅延はコーデックの性能、ホップ数による影響が大きい。本稿ではコーデック、ホップ数による遅延については、十分抑えられているネットワークを前提として評価を行う。最適なバッファサイズはキューイングにかかる最大遅延時間を元に求められる。本稿ではキューイングの最大許容遅延時間を 5[msec]とした。帯域を x[Mbps]とすると以下の計算式より最適バッファサイズ b[packets]は導出される[12]。

 $b = 10.4 \times x \cong 10 \times x$

1.3.2 EMBAC-D と EMBAC-P の性能評価

まず EMBAC-D の評価を行う。リンク帯域を 2.5[Mbps] とし、呼量を $295\sim310[erl]$ と変化させ、評価を行った。

図 1-4 に EMBAC-D におけるアドミッション閾値と平均 VoIP パケット損失率、図 1-5 にアドミッション閾値と呼損率の関係を示す。

アドミッション閾値が 100[%]とは、プローブパケット損失率が 100[%]でも接続許可をするという意味である。図 1-4 よりアドミッション閾値が 0~98[%]の範囲では平均 VoIP パケット損失率にあまり変化が見られない。このことから、プローブパケット損失率で VoIP パケット損失率をコントロールできる範囲は、極端に狭いということがわかる。例えばパケット損失率を 0.2[%]にするためには、アドミッション閾値をパケット損失率が急激に増加する、極めて不安定な領域で精密に制御する必要があるが、これは実質的に困難である。仮にアドミッション閾値を 80[%]と設定すると、パケット損失率は目標の 0.2[%]に比べて低すぎるため、目標とする平均 VoIP パケット損失率を満たすために過制御となってしまう。これは図 1-5 の呼損率についても同じ事が言える。これでは無駄な呼損が多く、接続性という意味において品質が良いとは言えない。

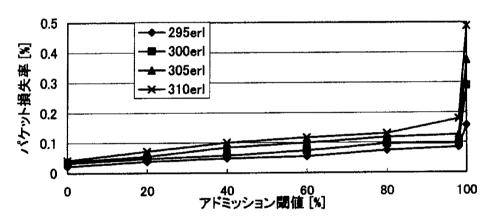


図 1-4 EMBAC-D の性能評価 (パケット損失率)

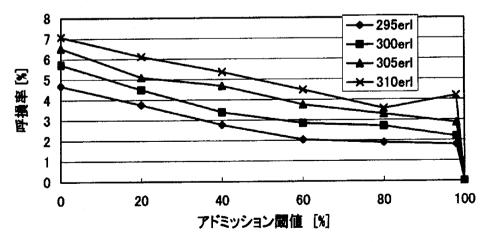


図 1-5 EMBAC-D の性能評価 (呼損率)

次に、接続拒否された呼についても一定の確率で接続許可を行う EMBAC-P を評価する。アドミッション確率にはアドミッション閾値と同じ値を使用した。つまり、アドミッション閾値が 30[%]のときはプローブパケット損失率が 30[%]を超えたとしても 30[%]の確率で接続許可とする。図 1-6に EMBAC-Pでのアドミッション閾値と平均 VoIPパケット損失率、図 1-7にアドミッション閾値と呼損率の関係を示す。図 1-6、図 1-7より、EMBAC-Pではアドミッション閾値 0~100[%]にかけて、滑らかに平均 VoIPパケット損失率と呼損率が変化していることが判る。図 1-8に確率だけでアドミッションコントロールした場合の評価を載せる。設定値は呼量 100[er]、リンク帯域 900[kbps]とした。結果より、確率だけでは、VoIPパケット損失率は図 1-6 のように滑らかに推移しないことから、プローブは必要だと言える。

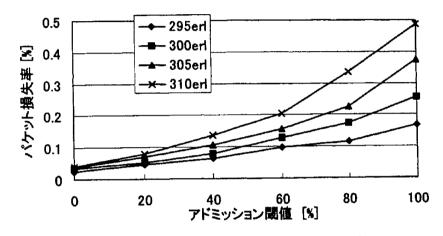


図 1-6 EMBAC-P の性能評価 (パケット損失率)

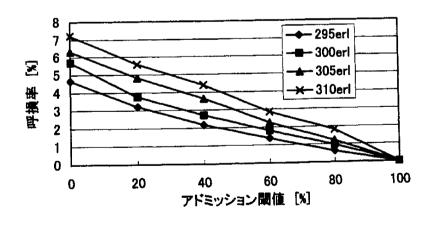


図 1-7 EMBAC-P の性能評価(呼損率)

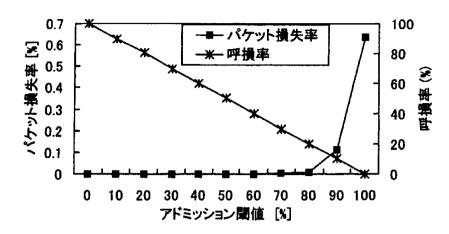


図 1-8 確率だけでアドミッションコントロールを行った場合

図 1-9 に目標平均 VoIP パケット損失率を 0.2[%]とした場合の呼損率の評価を示す。 EMBAC-P の方が EMBAC-D に比べ、同じ目標平均 VoIP パケット損失率の場合に呼損率 が少ない事がわかる。このことから、EMBAC-P が EMBAC-D よりも効率的なアドミッションコントロール方式であることが言える。

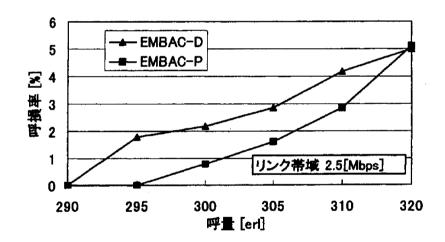


図 1-9 目標 VoIP パケット損失率 0.2[%]の時の呼量別呼損率

1.4 まとめ

1章では、音声品質を良好にたもつためのアプローチである、EMBAC 方式についての説明を行い、その性能評価を行った。その結果、プローブフローによる網状態測定は有効なものであり、この測定に基づき、呼損を発生させることで、網内の輻輳状態を抑制可能であることを示した。また、EMBAC-D は目標とする平均 VoIP パケット損失率を満たす為に過制御となってしまうことから、無駄な呼損が多く、接続性という意味において品質がよいとは言えない。これに対して EMBAC-P は、アドミッション閾値とパケット損失率、呼損率の間に線形的な相関が見られることから、コントロールがしやすい方式であると言える。このため、EMBAC-P の方が EMBAC-D に比べ、同じ目標平均 VoIP パケット損失率の場合に呼損率を低下させることができるため、効率的なアドミッションコントロール方式であると言える。これらのことは、パケット損失率の目標値を変化させた場合においても成り立つ。

参考文献

- [1] 間瀬憲一, "インターネット品質・トラヒック管理[I]", 電子情報通信学会誌, vol82, no.10, pp.1054-1162, Oct.1999.
- [2] 住本順一, 間瀬憲一, 横井忠寛, "インターネットの品質・トラヒック管理[II]", 電子情報通信学会誌, vol82, no.12, pp.1256-1263, 1999.
- [3] V. Elek, G. Karlsson, and R. Ronngren, "Admission control based on end-to-end measurement," INFOCOM 2000, 2000.
- [4] G. Bianchi, A. Capone, C. Petrioli, "Throughput analysis of end-to-end measurement-based admission control in IP," INFOCOM 2000, 2000.
- [5] L. Breslau, E. Q. Knightly, S. Shenker, I. Stoica, and H. Zhang, "Endpoint admission control: architectural issues and performance," pp. 57-69, Proc. SIGCOMM 2000, 2000.
- [6] S. Shioda, K. Nakamura, "A Measurement-Free Admission Control for the IP Telephony Service over Diffserv-Based Networks", ICC2005, CQ04-3, May 2005.
- [7] 外山雄一郎、間瀬憲一、"VoIP ネットワークにおけるエンド・エンド測定を用いたフィードバック制御型アドミッションコントロール方式", 2001 年電子情報通信学会ソサイエティ大会講演論文集, no.SB-7-4, pp.571-572, sep.2001.
- [8]K.Mase and Y.Toyama, "End-to-end Measurement Based Admission Control for VoIP Networks", ICC 2002. 2002.
- [9] 小林尚人, 間瀬憲一, "VoIP ネットワークにおけるエンドポイントアドミッションコントロールの性能評価", 電子情報通信学会誌, vol102, no.190, pp.59-64, July.2002.
- [10] 小林尚人, 間瀬憲一, "VoIP ネットワークにおける確率的エンドポイントアドミッションコントロール", 2003 年電子情報通信学会総合大会講演論文集, B·11-11, pp593, Mar.2003.
- [11] ns-2, http://www.isi.edu/nsnam/ns/
- [12] K.Mase and H.Kobayashi, "An Efficient Endto-End Measurement-Based Admission Control for VoIP Networks", ICC 2004, 2004.

2. 迂回中継を考慮したアドミッションコントロール方式

2.1 はじめに

この章では、ネットワーク上のトラヒック量が増大し、過負荷状態となった場合において、過負荷であるために接続拒否された呼の迂回中継を行うことで、接続性を向上させるための迂回ルーティングについての検討を行う[1]。

まず 1 章で用いた 1 リンクネットワークモデルを使用し、過負荷状態とした場合の EMBAC の問題点と改善策について説明する。この問題点とは、過負荷リンクではプローブフローでのネットワーク状態測定が不正確となり、接続許可と判断される呼が増えることでパケット損失が増大する点である。この問題点を改善するため、過負荷状態ではプローブフロー送受信の結果、接続許可と判断された呼について、確率的に接続許可と接続拒否の再振り分けを行う。この方式を確率呼損 EMBAC 方式 (EMBAC-RP) と呼ぶ。

次に、この EMBAC-RP 方式を用い、接続拒否された呼を迂回中継することで、ネットワークリソースを効率よく利用することを目的としたアプローチについて説明する。1リンクネットワークモデルでは、接続要求が過大となると通話中の呼を保護するために接続拒否される呼が多くなり、呼損率が高くなる。このため、このような高負荷なリンクからの溢れ呼を低負荷なリンクへと迂回させる迂回ルーティングが効果的であると考えられる。また、迂回路が複数存在する場合には、使用経路の選択も課題となる。

2.2 確率呼損 EMBAC 方式

確率呼損 EMBAC 方式の流れを図 2-1 に示す。この方式では、過負荷時において、プローブパケットが損失なく送受信された時、生起呼量に応じた呼損化確率(RP)に基づき、接続を確率的に拒否する。例として RP=10[%]の場合は、プローブの損失が全く起こらなかった呼に対して、10[%]の確率で接続を拒否に変更とする。

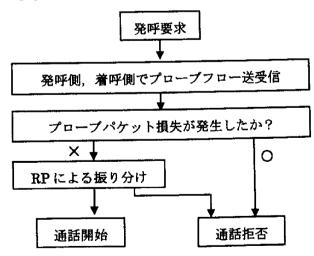


図 2-1 確率呼損 EMBAC 方式の流れ

ノード A、B から形成される 1 リンクネットワークモデルにおいて、アドミッション関値(AT)が 0[%]の時に目標パケット損失率を満たすことのできる最大の呼量を Amax[er]とする。ノード A から B への生起呼量 Aab が Amax 以下である時は RP を使用せずに最適な AT を設けることが可能であるが、そうでない場合は RP を設定する必要がある。 RP は式(1) より最適値を求めることが可能である。

$$RP[\%] = \begin{cases} 0 & (Aab \leq A \max) \\ 100 \times (1 - \frac{A \max}{Aab}) & (Aab > A \max) \end{cases}$$
 (1)

2.3 確率呼損 EMBAC 方式の評価

1リンクモデルを用いて、EMBAC-RP と EMBAC-P の性能評価を行った。シミュレータとして ns-2 を用いた[2]。VoIP 呼はポアソン到着をし、呼の保留時間は平均保留時間 180[sec]の指数分布に従うとする。またプローブフローの優先度を VoIP 呼よりも低く設定した。また、その他のシミュレーションパラメータを表 2-1 に示す。目標パケット損失率は 0.1[%] とした。帯域 1[Mbps]に対して、Amaxは 113[er]]と求まった。

パケット損失率と呼損率についての比較をそれぞれ図2-2、図2-3に示す。

帯域	1 [Mbps]	想定コーデック	G.729
パケット送信間隔	0.02[sec]	有音率	30[%]
パケットサイズ	40~60[byte]	データ取得開始時間	3600[sec]
プローブ送信時間	1[sec]	データ取得終了時間	10800[sec]

表 2-1 シミュレーションパラメータ

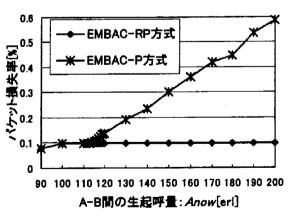


図 2-2 パケット損失率の比較

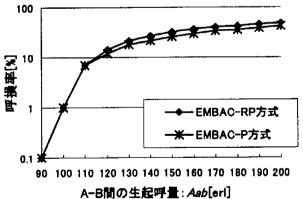


図 2-3 呼損率の比較

図 2-2 より EMBAC-P では Amax である 113[erl]以上の場合に、パケット損失率の制御ができない。この理由は過負荷状態でのプローブフローが不正確となり、容量以上の接続を許可とするためである。一方 EMBAC-RP では、過負荷となってもパケット損失率の制御は可能である。これは確率的に呼損を発生させることでプローブの不正確性を補うこと可能であることを示している。しかしこのために呼損率は EMBAC-RP の方が高くなる[3]。

2.4 確率呼損 EMBAC 方式を用いた迂回ルーティング方式

図 2-4 のようなリンクにおいて、A から B への生起呼量 Aab が Amax 以上である場合、AB リンクからの溢れ呼量を AOab[erl]と呼び。反対に AB リンクの負荷が小さく空き容量が存在する場合、AB 間の空き容量を ASab[erl]と呼ぶ。AOはその送受信間の生起呼量と、Amax を用い、式(2)によって算出可能である。式(2)での B とは、負荷が Amax の時の呼損率を意味する[4]。

AB 間が過負荷である場合、溢れ呼と呼ばれる接続拒否呼が多大となるために、接続性が低い。そこで 1 リンクモデルにノード C を付随した 3 ノードネットワークモデル(図 2-5) を考える。全てのリンク帯域は同一であり、AB 間を過負荷なリンク、AC 間および BC 間を空いたリンクとする。この場合、AB を結ぶ最短経路で接続拒否された溢れ呼を $A \rightarrow C \rightarrow B$ と迂回中継させることが可能となる[5][6]。

発呼要求が生起する毎に、送受信ノードはプローブフローを最短経路と迂回路へ 2 フロー送出する。最短経路で接続拒否され、迂回路が空いていると判断された時は、迂回路で呼接続する。ここで、ASの求め方を 3 通り提案する。式(3)を Min 法、式(4)を Max 法、式(5)を Dynamic 法と呼ぶ。 Max 法で設定される ASが最も大きく、Min 法が最も小さい。 Dynamic 法は Min 法と Max 法の間を、2 次式を用いて算出する方式であり、負荷が小さい場合には Max 法に近似し、負荷が大きくなるに連れて、Min 法に近似する特徴を持つ方式である。AC 間の空き容量 ASac[erl]と BC 間の空き容量 ASbc[erl]それぞれ算出し、比較した結果、小さい方を迂回路としての空き容量 ASとする。

$$AOab = Aab - A\max \times (1 - \beta) \tag{2}$$

$$ASac = A \min - Aac \tag{3}$$

$$ASac = A \max - Aac \tag{4}$$

$$ASac = \frac{(A\min - A\max) \times Aac^{2}}{A\min^{2}} + A\max$$
 (5)



図 2-4 1リンクモデル

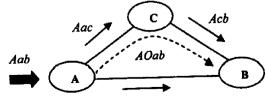


図 2.5 3 ノードネットワークモデル

迂回路のASが AB リンクからの溢れ呼量AOab よりも大きい場合には全ての迂回呼を許容できる。しかしAOab が過多となり、AS を超える場合は、迂回呼の流入により迂回路においても輻輳を招く。この場合、迂回路が過負荷状態であり、迂回路でのプローブフロー測定が不正確となるため、パケット損失率の制御ができなくなる。そこでこれを防ぐために、迂回路でも EMBAC-RP を使用する。迂回呼に対する呼損化確率 RP2 は式(6)より設定する[7]。

$$RP2[\%] = \begin{cases} 0 & (AOab \leq AS) \\ 100 \times (1 - \frac{AS}{AOab}) & (AOab > AS) \end{cases}$$
 (6)

2.5 3ノードモデルの性能評価

Min 法、Max 法、Dynamic 法の性能評価を行う。過負荷リンクである AB 間の生起呼量 Aab を 150[erl]とし、空いたリンクである AC および CB の呼量 Aac、Acb を共に 10[erl] ~90[erl]とした。 β は 1 リンクモデルでのシミュレーションにより、 0.1 と求まった。

呼損率の比較を図 2-6 に示す。1 リンクモデルの時と比較すると、迂回ルーティングを利用することにより、大幅に呼損率を低下させることが可能だと言える。また、迂回ルーティングに関する 3 方式を比較すると、Max 法が最も呼損率を低くでき、Min 法が最も高くなる。一方で Dynamic 法は Aac が小さい場合には Max 法に近似し、Aac が大きくなるに従い Min 法に近似する。これらは式(3)~(5)により算出される AS 値の違いが顕著に表れた結果と言える。

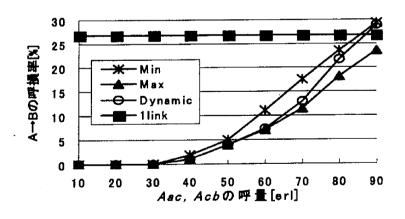


図 2-6 A→B接続要求に対する呼損率

パケット損失率の比較を図 2-7、2-8 に示す。A→C 呼、C→B 呼といったもともと空いた 経路へ流れている呼も、ノード C を通って迂回中継する呼も、Max 法ではパケット損失率 が目標値を大幅に超えることから、許容迂回呼量が過大だと言える。また Min 法では、パ ケット損失率が目標値よりも小さいが、その一方で呼損率が大きいため、迂回呼に対して、 過制御となっていると言える。Dynamic 法は、Aac、Acb が低負荷の時には Max 法と近似 し、高負荷の時には Min 法と近似するために、パケット損失率にも同様の性質が見られる。 つまり、Min 法のような過制御を避けることができ、さらに目標パケット損失率を満たす ことが可能であると言える。以上の比較から、パケット損失率がどの呼に対しても目標値 0.1[%]以下を常に満たし、さらには過制御を避けることができている Dynamic 法が最も有 効な方式であると言える。

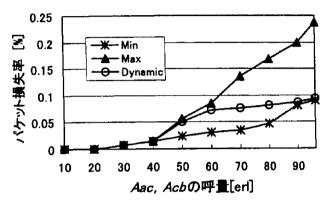


図 2-7 A→C 呼, C→B 呼のパケット損失率

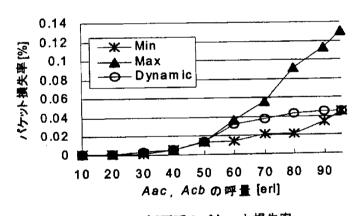


図 2-8 迂回呼のパケット損失率

2.6 迂回路が複数存在する場合のアドミッションコントロールの検討

前節での検討を発展させ、最短な迂回路が複数 (k 本) 存在する場合のアドミッションコントロールについての検討を行った。モデルは図 2-9 に示す 5 ノード完全グラフ型である。一対の送受信ノードに注目すると、このモデルでの最短な迂回路は、2 ホップ迂回路であり、3 本の 2 ホップ迂回路が存在することから、k=3 となる。このように、複数の迂回路が存在する場合には、3 ノードモデルと異なり、使用する迂回路の選択法を考慮する必要がある。迂回路選択法として以下の 2 つの方式を提案し、性能評価を行う。

●迂回路学習方式(Dynamic Routing Advanced Mode)[8][9]

直前の呼が選択した迂回路と、その経路で送受信されたプローブフローの状態を記憶する。プロープフローの結果、混雑していると判断された場合は、次に生起した呼はこの経路を除く(k-1)本の経路からランダムに選択する。反対に、混雑していないと判断された場合には、次に生起した呼もこの経路を選択し、プローブフローの送受信を行う。

●確率+学習方式

迂回路学習方式の機能を拡張した方式。過負荷経路でのプローブフローの不正確性を補うため、迂回路でのプローブ損失がない場合に、式(7)に示す経路選択確率 S(x)に基づき、迂回路の再選択を行う。ASaxb[erl]とはノード A からノード x を中継してノード B へと達する迂回路が持つ空き容量を意味する。空き容量の少ない経路では S(x)が大きくなり、経路変更が多発する。反対に空き容量が大きい経路では S(x)が小さくなるために経路切り替えは少なく、溢れ呼が多く流入する。またプローブフローにより混雑と判断された場合には学習方式と同様に経路変更を行う。また迂回呼の接続が空き容量以上とならないよう、式(8)に従い呼損化確率 RP2 を設定する。ここで AB 間において、迂回可能な最短ホップ集合を $De=\{d_1, d_2, \dots, d_k\}$ と定義し、図 $2\cdot 9$ のモデルにおいて、AB 間に注目すると、 $De=\{C,D,E\}$ となる。また、提案方式の流れを図 $2\cdot 10$ に示す。

$$S(x)[\%] = \left\{1 - \frac{ASaxb}{AOab}\right\} \times 100 \tag{7}$$

ただし、ASaxb > AOab の場合 S(x)=0[%]

$$RP2[\%] = \left(1 - \frac{\sum_{i=1}^{k} ASad_{i}b}{AOab}\right) \times 100 \qquad (8)$$

ただし、 $\sum_{i=1}^{k} ASad_i b > AOab$ の場合 RP2=0[%]

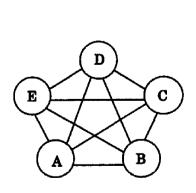


図 2-9 5 ノードモデル

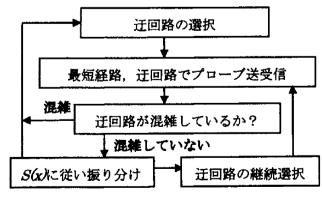


図 2-10 提案方式の流れ

2.7 5ノードモデルでの性能評価

各方式での呼損率、パケット損失率を評価する。ネットワーク内に 1 つの過負荷リンクが存在し、その他のリンクには空きがあると想定し、この過負荷な接続要求に対する呼損率を比較する。またパケット損失率は、1回のシミュレーションにつきネットワーク内で最もパケット損失率が高いリンクを採用し、20回のシミュレーションによる平均値とする。

Aから Bへの生起呼量が 180[erl]、AE、BE 間にはもともと 90[erl]の負荷があり、他のリンクの生起呼量を 10~90[erl]の負荷を与えた場合の性能比較を行う。この時、迂回路の中では最も混雑しているノード E を経由する迂回路を避け、空き容量の多いノード C 及びノード D を中継する迂回路を使用することが効果的な迂回となる。単一迂回路とは、迂回路を切り替えないモデルを指し、ノード C を経由する迂回路のみを使用する場合とする。

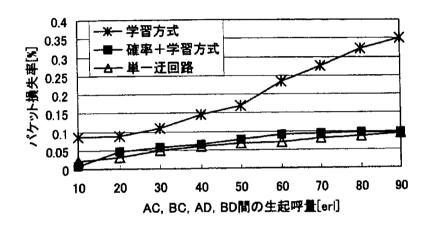


図 2-11 網内最大パケット損失率の推移

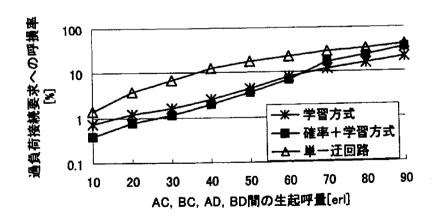


図 2-12 過負荷接続要求に対する呼損率の推移

パケット損失率については、ノード C 及びノード D を経由する迂回路に元々流れている 呼量、つまり AC、BC、AD、BD 間の生起呼量を横軸にとったグラフとして図 2-11 に示す。 また、呼損率の比較を図 2-12 に示す。まず過負荷リンク AB を除き、網内でパケット損失率が最大となるリンクは、リンク AE、BE のどちらかであった。学習方式は網内負荷が小さい場合には、パケット損失率を目標値 0.1[%]以下に制御できるが、負荷が大きくなるに連れて、パケット損失率の劣化が生じる。この理由としては、プローブが不正確であるために、空き容量を越えた通信を許容している点、そして、経路の変更をプローブフローの 結果にのみ基づいているために、最も混雑した経路を迂回路として多用する点が考えられる。

確率+学習方式では経路選択確率 S(x)と迂回呼損化確率 RP2 の設定により、低負荷リンクを迂回路として多用させることで、呼量によらず全リンクにおいてパケット損失率の目標値を満たすことが可能である。また経路選択確率 S(x)を用いることで、網内トラヒックに偏りが大きい時には、負荷の小さな経路を迂回路として多用するために、呼損率の面で有効である。反対に網内負荷の偏りが小さく、空き容量の総和が溢れ呼量よりも小さい場合には、RP2 により制御をかけることで無理な迂回を抑え、網内の品質保証を行うことができると言える。このために呼損率は、網内トラヒックが増えるに従い、学習方式よりも高くなると考えられる。

また、学習方式、確率+学習方式ともに、迂回路の変更を行わない単一迂回路モデルと 比較すると、呼損率が低くなっている点が、経路変更を行う利点であると言える。

2.8 まとめ

過負荷なリンクが存在する場合の EMBAC についての検討を行った。網内負荷を測定するためであるプローブフローは、過負荷なリンクでは不正確となりパケット損失率の劣化を生じさせる。そこでこの問題点を補うための確率呼損 EMBAC 方式を提案し、パケット損失率の面で有効であることを示した。

一方、プローブフローを最短経路と迂回路の 2 経路に流す迂回ルーティングは、過負荷 リンクからの溢れ呼を、迂回路へと流すことができるため、呼損率の大幅な低下が可能で ある。また、迂回呼が増大すると、全ての迂回呼を許容することができなくなる。このた めに、迂回呼に対しても確率呼損 EMBAC 方式を使用することで、網内のどのリンクにお いてもパケット損失率を目標値以下に制御可能であることを示した。

迂回路が複数存在する場合には、確率+学習方式を用いることで、迂回路を切り替えて 利用できるために、呼損率をさらに低下させることができる。また、網内のどのリンクに おいてもパケット損失率を目標値以下に制御可能であることを示した。

以上のことから、迂回中継を考慮した EMBAC 方式の有効性を示すことができた。

参考文献

- [1] Marco Conte, "Dynamic Routing in Broadband Networks", KLUWER ACADEMIC PUBLISHERS, 2003.
- [2] ns-2, http://www.isi.edu/nsnam/ns/
- [3] 須田和彦, 間瀬憲一, "VoIP ネットワークにおける過負荷状態を考慮した EMBAC 方式", 2004 年電子情報通信学会ソサイエティ大会講演論文集, B·11·16, Sep. 2004.
- [4] 藤木正也,雁部頴一,"通信トラヒック理論",丸善株式会社,Sep. 1980.
- [5] H. Tamura, K. Kawahara, Y. Oie, "Performance analysis of flow loss probability and link utilization in MPLS networks for Traffic Engineering", IEICE Transaction on Communications, Vol.E87-B, No.3, pp.579-586, Mar. 2004.
- [6] H. Tamura, K. Kawahara, Y. Oie, "Performance Evaluation of Detour Path Selection Scheme for MPLS Traffic Engineering", IEEE PacRim2005, pp.109-112, Aug. 2005 [7] 須田和彦, 鈴木裕和, 間瀬憲一, "VoIPネットワークにおける過負荷リンクを考慮したエンドポイントアドミッションコントロールの性能評価", 2005 年電子情報通信学会技術研究報告, CQ2005-11, Apr. 2005.
- [8] 孔翔, 間瀬憲一, "VoIP ネットワークにおける迂回を考慮したエンドポイントアドミッションコントロール", 2002 年電子情報通信学会技術研究報告, CQ2002-60, Jul. 2002.
- [9] X.Kong, K. Mase, "Dynamic Routing with Endpoint Admission Control for VoIP Networks", ICC2003, CQ11-1, May 2003.

3 リンク設計によるAT決定方式とその評価

3.1 はじめに

最適なアドミッション閾値(AT)は各エンドポイントが保持する。通信品質を目標値(平均パケット損失率、呼損率)に制御するためには、各エンドポイントがそれぞれ最適な AT を使用しなければならない(図 3.1)。しかし、通信を行うネットワークの帯域、呼量、ホップ数により最適な AT は違ってくる。各エンドポイントの最適 AT をどう決定するかが、問題となる。この章ではリンク設計により AT を決定する方法を提案し評価する。以降、本報告では目標平均パケット損失率と目標呼損率を単に目標値と呼ぶ。

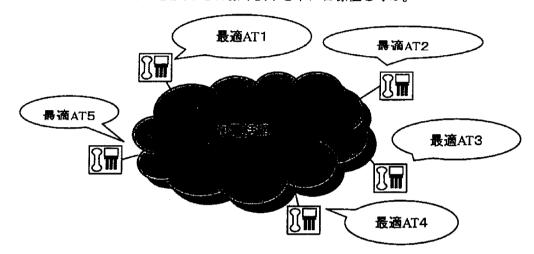


図 3.1 各エンドポイントと最適 AT

3.2 アドミッション閾値の決定方法

エンドツーエンドでの通信品質の目標を各リンクに分割する。リンク数がn個ある場合のエンドツーエンドでの品質目標値をbとすると各リンクでの品質目標値はb/nとなる(図3.2)。リンクごとにシミュレーションを行い。目標値を満たすように、帯域とリンクアドミッション閾値を用意する。リンクアドミッション閾値とは各リンクで目標値を満たす最適なアドミッション閾値のことをいう。

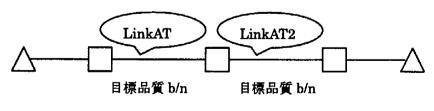
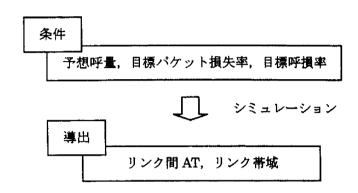


図 3.2 品質分割と AT

品質目標値の分割によりエンドツーエンドでの品質は各リンクの品質の和と考える事ができる。

1リンクでのシミュレーションでは、リンクへの予想呼量と目標パケット損失率、目標 呼損率より、最適なリンク AT とリンク帯域を導出する。リンク帯域は目標を満たす中で最 小量のものを選択する。そうすることで、最適なリンク AT とリンク帯域は1つに決まる。



3.3 シミュレーションモデルと条件

図 3.3 に評価に使用したシミュレーションモデルを示す。太くなっているリンクがリンク 設計を行ったリンクで2リンクをくっつけた形のモデルとなる。細いリンクはパケット損 失の起こらない理想的なリンクとした。

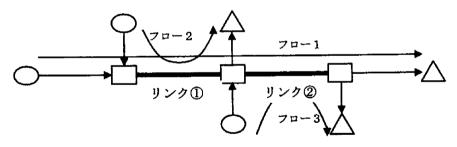


図 3.3 2リンクモデル

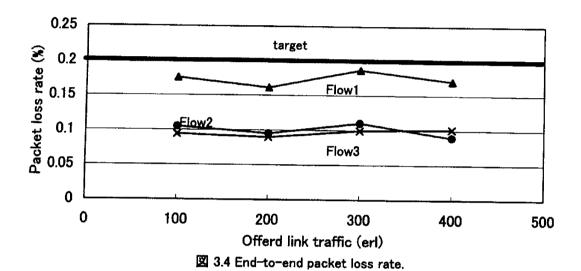
リンク①の呼量=フロー1の呼量+フロー2の呼量 リンク②の呼量=フロー1の呼量+フロー3の呼量 となる。

目標値は以下のように設定した。

1リンクでのパケット損失率	0.1[%]
1リンクでの呼損率	1[%]
フロー1 (2リンクを跨ぐ) のパケット損失率	0.2[%]
フロー1の呼損率	2[%]

3.4 シミュレーション結果

図 3.4、図 3.5 はリンク①とリンク②へかかる呼量を同じとした場合のパケット損失率と呼損率である。この場合、リンク①とリンク②の最適 AT は等しいので、その値を全エンドポイントで使用する。この評価では呼量についてフロー1と他フローの比率を1:2とした。つまりリンク①(=リンク②)へかかる呼の 1/3 がフロー1 からの呼となる。図 3.4 ではリンクへかかる呼量がどうあれ、全フローがエンドツーエンドでの目標パケット損失率の 0.2%を満たしている。図 3.5 についても同様に目標呼損率を全フローが満たしている。リンク間呼量が同じ場合はリンクへの品質分割によりエンドツーエンドで目標値を満たすことが可能であることがわかる。



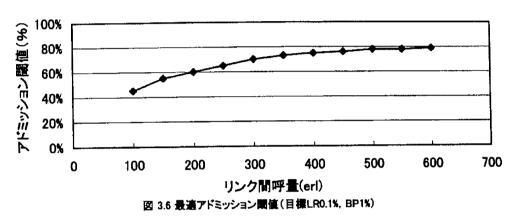
(Percentage of Floww 1=1/3)

2.5 Blocking probability target 2 £^{1.5} Flow 1 Flow2 1 Flow3 0.5 0 0 100 200 300 400 500 Offerd link traffic (erl)

図 3.5 End-to-end blocking probability. (Percentage of Floww 1=1/3)

図 3.4、図 3.5 で、呼量がリンク①リンク②で均等な場合では目標値を満たす制御ができるという結果となった。

リンクにかかる呼量が不均等な場合は、呼量差が大きくなるとアドミッション閾値が不適当な状態になり得る。これは図 3.6 のようにリンク間呼量が大きくなるに従って最適なアドミッション閾値も大きくなるために起こる。例えばリンク①が 100[erl]リンク②が 600 [erl]の場合には各リンクの最適アドミッション閾値に 35[%]近くの差が出てしまい、どちらの閾値をエンドポイント用に選んだとしても、他方のリンクを通るフローに関してはアドミッション閾値が不適当となる。



リンク間呼量にばらつきがある場合の補正方法として、帯域を増やす方法を考察する。 下記の2通りの方法でATを選択する。

- ① ネットワーク内で1番低負荷のリンクにより求めた AT を全エンドポイントが採用する (ATmin 方式)
- ② ネットワーク内で1番髙負荷のリンクにより求めたATを全エンドポイントで採用する (ATmax 方式)

ATmin 方式と ATmax 方式それぞれにおいて AT が不適当なリンクは目標値を満たすよまで帯域を増加させる。図 3.3 のリンク①リンク②にそれぞれ掛かる呼量の合計を 700(erl) とし評価を行った。結果は図 3.7 のようになる。リンク①とリンク②の間の呼量に格差がある場合のリンク①とリンク②の合計帯域を、リンク①、②両方とも 350(erl)の場合を 100%とし表した。独立な場合とはリンク①、リンク②に関係がまったく無い場合であり、図 3.3のフロー1が無い場合を指す。つまり、各リンクが最適な AT を使用できた場合の帯域である。

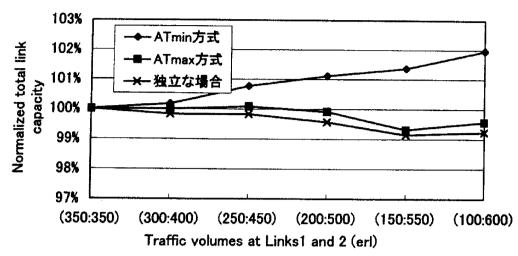


図 3.7 Comparison of admission threshold selecting approaches.

図 3.7 より、ATmax 方式を使用した方が増加帯域を少なくする事ができる事がわかる。 つまり高負荷な大規模リンクを先に設計して低負荷なリンクがあわせる方式が帯域増加の コストの面で有効であると言える。

3.5 まとめ

簡単なネットワークモデルを対象に各リンクに加わる呼量に基づきエンドエンドの通信 品質の目標値(平均パケット損失率、呼損率)を制御するための最適なアドミッションコントロール関値(AT)の設計方法を示した。

4 ソフトフォン試作

4.1 はじめに

この章では、本研究において作成した LinuxOS 端末上で動作する電話機であるソフトフォンについて説明する。

第1章において、EMBACの評価を行った。これはシミュレーションによる評価であり、これらのアプローチを実装し、実験による評価は行われていない。実装には、パソコン上で動作する電話機であるソフトフォンにそれらの機能を追加する必要がある。しかし、これらのソフトフォンは実用面を重視して設計されているので、多機能で複雑な構造をしている。それゆえにプログラムが膨大になり、機能の追加が困難あるいはできないことがある。そこで本研究では、EMBACなどの機能を追加しやすくするために、シンプルな機能、構造をしたソフトフォンを作成した。

ソフトフォンを作成し、機能の追加を行う前段階として、ソフトフォンの基本性能の評価実験を行った。作成したソフトフォン同士でセッションの確立・切断を確認した上で、音声品質評価を行った。また、実験ネットワークに擬似的に遅延揺らぎを発生させた場合に、それに対応した機能として実装したジッタバッファの性能評価、および他のソフトフォントとの相互接続性の実験を行った。なお、音声品質評価指標としては客観的評価 PESQ を用いた。

実験を行った結果、ソフトフォンは PESQ による評価値としてはクラス B(携帯電話以上の品質)であり、快適に通話を行えることを確認した。また、遅延揺らぎを発生させた場合、揺らぎを吸収する役割であるジッタバッファの機能を確認し、MOS 値を上昇させることができた。 SIP を実装した他のソフトフォントの相互接続では、セッションの確立・切断ができることを確認した。

作成したソフトフォンによるセッションの確立・切断および、通話を行った上で音声品質評価を行った上で、ソフトフォンの改良を行った。作成したソフトフォンに EMBAC 機能を実装した。実装においては、あらかじめアドミッション閾値を設定しておき、プローブの送受信を行った上で、送受信結果と閾値を比較し、接続可否の判定を行う。本報告では、EMBACを実装し、その動作確認を行った。

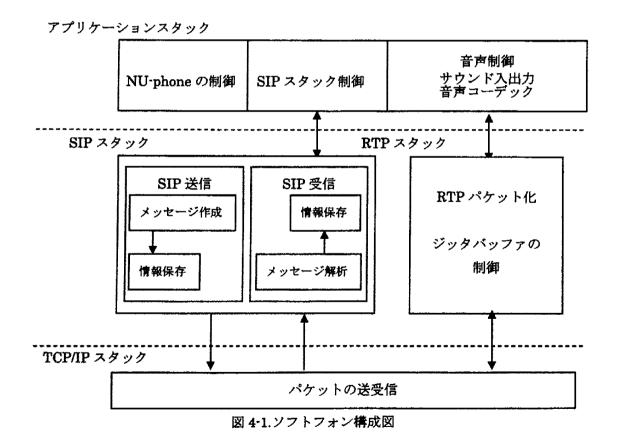
4.2 ソフトフォン NU-phone(Niigata University-phone)

本研究において作成したソフトフォンは、シグナリングプロトコルとして SIP(Session Initiation Protocol)[1]、音声に関して、コーデックとして G.711 μ -law[2]、転送プロトコルとして RTP(Real-time Transport Protocol)[3]を用いた。SIP は RFC3261 で規定されており、インターネット上でマルチメディアのセッションを開始、終了を管理、制御するために開発されたプロトコルである。また、メッセージが HTTP に似ているため WWW との親和性も高く、次世代のインターネット電話機能として注目されている。

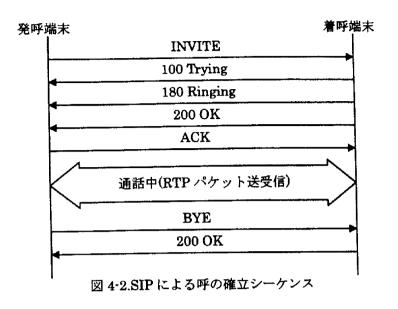
音声を送受信するとき、コーデックとして $G.711 \, \mu$ -law を用いている。 $G.711 \, \mu$ -law は現在使用されているコーデックでは最も音声品質が高く、 $300 \, \text{Hz}$ から $3.4 \, \text{kHz}$ の帯域信号を $8 \, \text{kHz}$ で標本化し、各標本を $8 \, \text{ビットで表すので音声データ量は } 64 \, \text{kbps}$ となる。この音声データの転送で使用した RTP はヘッダにタイムスタンプ、シーケンス番号などの情報が含まれ、パケットの整合性が取れるようになっている。

ソフトフォンの構造は図 4-1 に示すように、3 層構造からなる。TCP/IP スタックでは、パケットの送受信を行い、SIP スタックでは、SIP メッセージの作成、受信、解析を行う。SIP メッセージは単純な通話を行うのみ最低限必要なメッセージのみを実装しており、その SIP メッセージのシーケンスを図 4-2 に示す。SIP スタックと並んで、RTP スタックがあり、ここでは、受信した音声パケットを RTP ヘッダに基づいてパケットの並べ替えや、誤り訂正処理を行う。アプリケーションスタックでは、下位スタックを管理し、アプリケーション全体の制御を行う。また、NU-phone の状態管理やコーデック処理、サウンド制御を行う[4]。

プログラミング言語として、スクリプト言語である Ruby[5]を用いた。一般的には C 言語により実装されることが多いが、Ruby ではテキスト処理を高速に行えるため、テキストベースのシグナリングプロトコルを実装する上で Ruby の機能を最大限活用できる。



- 24 -

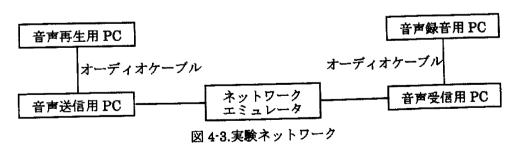


4.3 音声評価実験

前章では、NU-phone の概要について述べた。本章では、NU-phone の音声品質の評価を行った[6][7]。また、IPネットワークを用いる VoIP において、ジッタが発生するため、これを抑制する揺らぎ吸収バッファ(ジッタバッファと呼ぶ)の有効性の評価も行った。これらの結果をフリーソフトフォンである Linphone と Windows Messenger とで比較を行った。

4.3.1 実験環境と測定条件

本研究で使用した実験環境を図 4-3 に示す。ソフトフォンをインストールした PC に音声録音用、再生用のパソコンがオーディオケーブルにより接続されている。ソフトフォンがインストールされている PC はネットワークエミュレータに接続されている。ネットワークエミュレータは、擬似的にパケット損失や遅延、揺らぎなどを発生させることができる装置であり、実際のネットワークで起こり得る環境を作り出すことが可能である。本実験では、ジッタバッファの検証のため、遅延揺らぎのみを設定している。また、遅延を発生させないときには、遅延の設定をオフにすることにより単にブリッジとして動作する。



ソフトフォンにおいて SIP メッセージは、SIP サーバを介さずに P2P で送受信し、音声 パケットも RTP を用いて P2P で送受信される。

あらかじめソフトフォン同士でセッションを確立して通話状態にしておき、音声再生用PCでは、表 4·1 に示す 4 種類の評価用音声を再生する。これらの音声を、実験ネットワークを介して受信側PCで音声を再生し、音声録音用PCで音声を録音する。音声の再生と録音を 10 回行った。音声の送受信が 1 回終わるとセッションを切断し、また新たにセッションを確立する。これを 4 つの評価用音声に対して行い、再生音と録音した音声とを PESQ 評価ツールを用いて PESQ MOS を算出した。

表 4・1.評価用音声リスト

表 4-2.オーディオケーブルによる劣化

サンプル音声	性別	再生時間(秒)
1	男性	10
2	女性	10
3	男性	10
4	女性	10

サンプル音声	MOS 値
1	4.40
2	4.41
3	4.46
4	4.45

4.3.2 実験環境による音質低下

PESQ MOS 値は最高で 4.5 である。そのため、評価用音声同士を PESQ MOS 評価ソフトで評価すると 4.5 になる。実験環境では、音声ファイル(ディジタル)はオーディオケーブルを通し、オーディオケーブルを通った音声(アナログ)がソフトフォンの入力となる。すなわち、IP 電話として音声を送信するよりも D/A 変換が 1 回多いことになる。受信側でも同じことが行われ、ソフトフォンの出力からオーディオケーブルを通った音声が A/D 変換されて評価対象音声ファイルとなるため、A/D 変換が 1 回多い。そこで、A/D、D/A 変換によりどれだけ MOS 値が下がるのか測定した。音声再生用 PC の出力と録音用 PC の入力をオーディオケーブルで接続し、それぞれの音声に対して 10 回測定し、平均を算出した。結果を表 4・2 に示す。表 4・2 より、オーディオケーブルによって MOS 値が最大で 0.1 下がった。この要因として A/D、D/A 変換やオーディオケーブルの伝送中にノイズが乗ったことが考えられる。したがって、図 4・3 では、再生端末と録音端末で 2 回の変換が行われるので、MOS の最大値は 4.3 となる。しかし実際に通話すると、PESQ MOS 値が示す品質よりも音質がよく聞こえる。

4.3.3 NU-phone の音声品質評価実験結果

図 4-3 の実験環境において、ネットワークエミュレータの設定を行わずにブリッジとして動作させたときの実験結果を示す。図 4-4 にソフトフォンの評価結果を示す。NU-phone同士で音声送受信を行った場合の MOS 値は 3.89 である。これは携帯電話並みの音声品質の MOS 値 3.6 以上である[8]。そして、既存のアプリケーションである Windows Messenger や Linphone と比較しても、NU-phone の方が高い品質である。

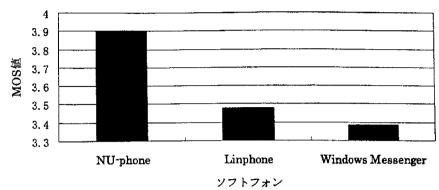
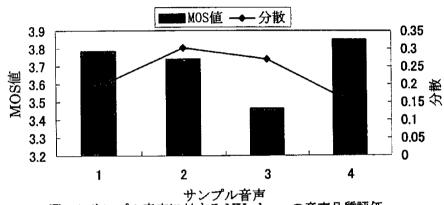


図 4-4.ソフトフォンの音声品質評価結果



ザンノル音戸 図 4-5.サンプル音声に対する NU-phone の音声品質評価

4.3.2 で述べたように、実験ネットワークによる影響のため、録音された音声の MOS は最大で 4.3 である。図 4-4、図 4-5 より MOS 値は 4.3 より低下している。その主な要因として音声パケットの送信間隔が挙げられる。コーデックとして G.711 μ-law を用いているため送信間隔は 20 ミリ秒である。NU-phone でも 20 ミリ秒で送信しているが、その送信間隔が大きくずれて送信することがある。この送信間隔のずれが MOS 値に影響があると考えられる。同様のことが Linphone、Windows Messenger においても言える。Linux では、タイマ割り込みは 10 ミリ秒で設定されているが、パケットは 20 ミリ秒間隔で送信される。そのためパケット送信以外の処理が割り込みを行った場合には、パケット送信の間隔がずれてしまう。これを改善するためには、Linux のタイマ割り込みも時間を再設定する必要がある。

そこで、Linux のタイマ割り込みを 1 ミリ秒に短縮し、同様の実験を行った。図 4-6 に割り込み間隔が 1 ミリ秒のときの MOS 値と 10 ミリ秒の時の MOS を示す。結果より、割り込み間隔が 1 ミリ秒のときの MOS 値が 0.2 高くなっている。処理の割り込み間隔を短くすることによって、他の処理による送信間隔のずれが非常に小さくなる。音声パケットのモニタリングを行い、20 ミリ秒の間隔で音声パケットが送信されていることを確認した。

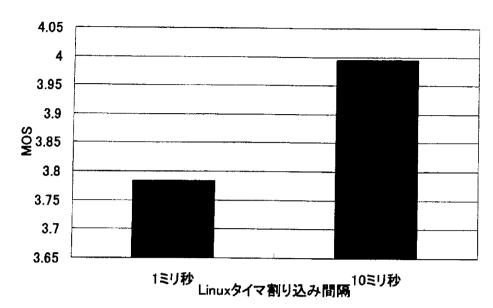


図4-6.Linuxのタイマ割り込み間隔によるMOSの違い

パケットの送信間隔のほかに、コーデックを使用したことによるデータの不完全性も考えられる。音声データを送信する際、圧縮処理してから送信している。受信端末で圧縮された音声データを受信し、元のデータに伸張するときに圧縮によりデータの欠損が生じ、完全なデータ再現が不可能になることがある。そのため、受信端末で再生された音声が元の音声と一致せず、MOSの低下につながった。

4.3.4 ジッタバッファの機能評価

IP ネットワーク上でバースト的なパケット損失や遅延が発生すると音声品質が大きく低下する。パケットの到着間隔にばらつきが発生する揺らぎ(ジッタ)が最も影響を受けやすい。到着間隔によってはパケットの到着順序が入れ替わることがある。ジッタバッファは受信したパケットをある長さの保存領域(バッファ)に一時保存しておき、パケットの順序を正しく並べ替え、再生するようにする[9]。このバッファのサイズは、小さすぎると遅れて受信したパケットは破棄され、音声が途切れて聞こえる。一方でバッファのサイズが大きいと、パケットが到着してから再生するまでの時間がなくなり、ネットワーク遅延にバッファからパケットを取り出す待ち時間がかかるため通話に影響が出る。そのため、ジッタバッファの大きさは、ネットワークの設計時あるいは、使用しているときの状態に応じて設定する必要がある。NU-phone に追加したジッタバッファの長さは 200 ミリ秒とした。これは、ネットワークの品質を定めた[8]より、最も品質の悪いクラス C で遅延が 200 ミリ秒以上と定義されている。このような環境でも使用できるようにするためジッタバッファの長さを200 ミリ秒とした。

そこで、NU-phone にもジッタバッファの機能を付加し、バッファがある場合とない場合で MOS 値が改善されるか測定、検証を行った。ネットワークに擬似的な遅延を発生させ

るために、ネットワークエミュレータを用いた。遅延の発生規則として、平均で20ミリ秒、 標準偏差が40ミリ秒の正規分布に従うように設定した。これにより、遅延は0ミリ秒から 140ミリ秒の間に99.73%発生する。

図 4-7 より、ジッタバッファがある場合では、MOS 値を大きく改善することができた。 これはジッタバッファが機能していることを示している。

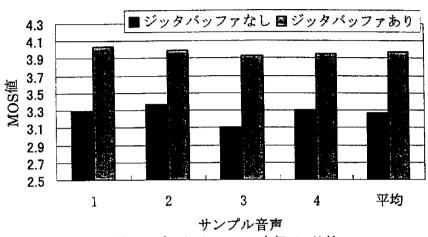


図 4-7.ジッタバッファの有無での比較

4.3.5 他のソフトフォンとの比較

前章では、NU-phone の性能評価として、PESQ MOS 値が携帯電話以上の品質であることがわかった。他のソフトフォンも MOS 値は低いものの、十分会話が成立する品質であった。しかし、現在の IP 電話ではシグナリングプロトコルの実装がメーカーにより異なるので、使用する IP 電話機やシグナリングサーバの組み合わせによっては通話できないなどの相互接続が問題となる。そこで、本章では Windows Messenger や Linphone との相互接続性を検証する。

4.3.6 相互接続性

ソフトフォンと呼ばれるソフトウェアの中で、シグナリングプロトコルとして SIP を用いているソフトウェアとの相互接続性を検証する。本研究では、Linux 上で動作する Linphone や Kphone、Windows 上で動作する Windows Messenger と NU-phone でのセッションの確立、切断を行った。

表 4-3 より、全てのソフトフォンの間で SIP によるセッションの確立、切断ができることが確認できた。会話可能であるかの判断は、マイクを使用して実際に会話を行い、会話が成立するかどうかを判断基準とした。表 4-3 より、NU-phone においては全てのソフトフォンに対してセッションの確立、切断が可能であり、また会話も行うことができた。 Linphone と Windows Messenger では、セッションの確立は可能であるが、会話を始めると Windows Messenger からセッションの切断を要求する BYE を送信するため、通話が行えなくなる。Windows Messenger がどのような実装をしているのか不明であったため、こ

の原因を追究することができなかった。

これらの表 4-3 で使用したソフトフォン以外に Kphone も使用したが、これに関しては全てのソフトフォンとセッションを確立できたが、会話が困難であった。その要因として、Kphone のパケット送信間隔が挙げられる。NU-phone では、20 ミリ秒ごとにパケットを送信しているが、Kphone では同じコーデックを使用しているものの、送信間隔が 113 ミリ秒から 137 ミリ秒で送信される。Kphone が音声パケットを受信する場合は、他のソフトフォンから送信されたパケットは、音声として認識できる品質であったであったが、安定して音声が再生されず、通話開始から 1 分ほどで会話が困難になった。

	NU-phone	Linphone	Windows Messenger
NU-phone	0	0	0
Linphone		0	Δ
Windows			
Messenger			

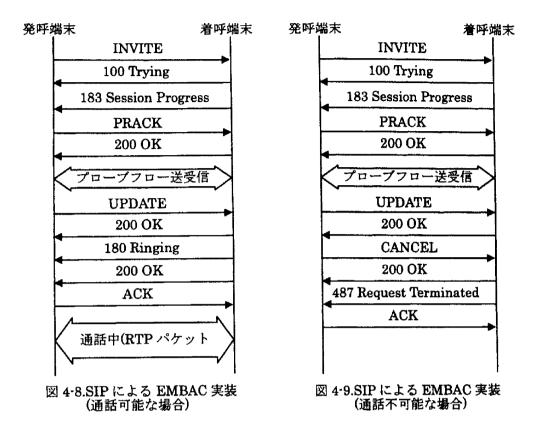
表 4-3.相互接続表

4.4 EMBACの実装

4.3 節で、セッションの確立・切断および、音声品質の評価を行った。その上で、本節では、EMBAC機能の追加を行う。EMBACは音声通話を行う前に、プローブフローを流す。しかし SIP では、通話を行う直前、すなわち着呼側が「200 OK」レスポンスに自分の IP アドレスやコーデック情報を含める。そのため、発呼側は通話を行う直前まで着呼側の IP アドレスやポート番号を知ることができないために、プローブフローの送受信を行えない。そこで、SIPによる通常のセッション確立までのメッセージのやり取りを変更した。図 4.7、図 4.8 に EMBAC を実装する際の SIP メッセージのやり取りを示す。

プローブフロー送受信を行う前に、着呼側が「183 Session Progress」レスポンスを創始する。これは信頼できる暫定レスポンスを示し、このメッセージのボディ部分に IP アドレスやコーデックなどの情報を示す記述言語である SDP(Session Description Protocol)[10]を付加することにより、発呼側は事前に着呼側の IP アドレスを知ることができる。「183 Session Progress」を受信すると、受信した旨を示す「PRACK」メソッドを送信し、プローブの送受信を行う。プローブ送受信後に、「UPDATE」メソッドを用いることにより、送受信結果をお互いに知ることができる。得られた結果とアドミッション閾値と比較し、通話可能ならば図 4-8 のような処理を行う。通話不可能ならば、図 4-9 のように「CANCEL」メソッドを送信し、セッション確立の中断を行う。

図 4-8、図 4-9 のように実装した NU-phone の動作確認を図 4-3 の実験ネットワークを用いて行った。プローブフローのみにパケット損失が起きるようにネットワークエミュレータの設定をした。NU-phone には、あらかじめアドミッション関値を設定しておく。このような状態で、NU-phone の動作確認を行い、実際に動作したことを確認した。



4.5 まとめ

本研究室で提案している EMBAC を実装するためにオープンソースソフトウェアのソフトフォンである Linphone を用いて実装してきたが、Linphone の仕様により実装が困難であった。そこで、機能の追加や拡張を容易にできることを目標としてソフトフォンを作成した。まずは、ソフトフォンの接続を確認した上で、音声品質の測定、評価を行った。その結果、作成したソフトフォン(NU-phone)はクラス B(携帯電話以上、固定電話以下の品質)であり、他のフリーのソフトフォンよりも高い品質であることを確認した。しかし、音声パケットの送信間隔にばらつきが見られ、送信間隔の短く、リアルタイム通信である VoIPでは品質に影響を及ぼす。主な原因として、Linux のタイマ割り込みを挙げた。Linux のタイマ割り込みを変えたときの MOS の比較を行った結果、タイマ割り込み間隔を短くすることにより、パケット送信間隔を一定し、MOS 値を高くすることができた。

実際のIPネットワークで使用すると、音声パケットの到着間隔にばらつきが発生し、到着したパケットをすぐに再生しようとすると、パケットの受信とパケットの再生のタイミングにずれが生じ、音質が低下する。この対策として、ジッタバッファ機能を追加する必要がある。そこで、NU-phone にジッタバッファ機能を追加し、音声品質評価を行った。その結果、MOS 値を高く維持することができた。これにより実装したジッタバッファが機能していることを確認した。

相互接続実験では、NU-phone はシグナリングプロトコルとして SIP を用いているソフトフォンとの接続実験を行った。NU-phone は SIP により他のソフトフォンとのセッションの確立、切断ができることを確認した。また、音声通話も会話が成立できることも確認した。

これらのソフトフォンの基本的な性能を評価した上で、EMBAC機能の追加を行い、動作確認を行った。ネットワーに故意にパケット損失を発生させ、NU-phoneで通話する前に行われるプローブフローの送受信により通話可否の判定を行った。あらかじめ設定している閾値と比較し、試験フローの結果から通話可否判定を行った。その結果、閾値に基づいた通話可否判定を行うことができたことを確認した。今後、さらなる性能評価を行う必要がある。

参考文献

- [1] J.Rosenberg et al., "SIP:Session Initiation Protocol", RFC3261, June.2002
- [2]ITU-T, "G.711 Pulse code modulation(PCM) of voice frequencies", 1988
- [3]H.Schulzrinne et al., "RTP: A Transport Protocol for Real-Time Applications", RFC3550, January, 1996
- [4]マッキーソフト株式会社, "基礎からわかる TCP/IP SIP による VoIP プログラミング", オーム社, 2004
- [5]http://www.ruby-lang.org
- [6]中岡謙, 加藤正美, "客観的音声品質評価法 PESQ による VoIP 端末の音声品質評価実験", 電子情報通信学会技術研究報告, CQ2002-57, 2002
- [7]北脇信彦, "IP 電話の知覚 QoS 評価法", 電子情報通信学会技術研究報告 CQ2002-56, 2002
- [8]総務省, "「IP ネットワーク技術に関する研究会」報告書", 2002
- [9]川西素春, 後藤雅徳, "基礎から理解する VoIP 技術", OPEN DESIGN 2003 年 3 月号 第 10 巻第 3 号(通巻第 69 号), 2003
- [10]M.Handley et al, "SDP: Session Description Protocol", RFC2327, April 1998

5 VoIP 品質管理システム

5.1 はじめに

この章では、VoIP ネットワークにおいてエンドツーエンドでネットワーク品質を監視するシステムについて説明する。

VoIP では、ネットワークのサービス品質(QoS)の低下が直接音声品質の低下に結びつくため、ユーザのネットワークを常に監視し、ネットワーク QoS の低下が起こった場合には対策を迅速に対応する必要がある。VoIP ネットワークにおいて重要な QoS パラメータとして、パケット損失、遅延時間、遅延揺らぎ、スループットが挙げられる[1]。これらの QoS パラメータを測定するためにこれまでの QoS 監視手法では、バックボーンネットワークの通信経路上において専用ハードウェアを設置し評価する手法が一般的である。しかし、この手法では実ユーザのエンドツーエンドでの品質評価とはならず、ユーザが感じる品質とは異なるものとなる。また、バックボーンネットワーク内でのネットワーク品質の低下や通信機器の障害の評価は行えない。このような問題を解決するため、VoIP 通信を行う端末にネットワーク品質測定機能を内蔵し、その情報を品質管理サーバに送信し、蓄積を行う方式[2][3]を説明する。

5.2 QoS 監視システム概要

QoS監視システムは以下の要素で構成される。

(1) 品質測定機能内蔵 VoIP 端末

品質測定機能内蔵 VoIP 端末は、VoIP 端末の基本的な機能とともに、プローブフローを 用いたアクティブ型測定機能を持つ。あらかじめ端末内に保存された擬似音声ファイルを プローブフローとして使用することにより、パケット損失率を求めるとともに、音声客観 品質評価も行えるようにする。さらに接続時間も測定する。呼制御プロトコルとして SIP を用いる。

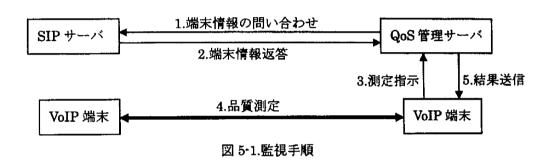
(2) QoS 管理サーバ

QoS 管理サーバはネットワーク QoS の測定指示を出し、測定により得られた結果を蓄積するサーバである。QoS 管理サーバではクライアントである VoIP 端末との間で、SIP メッセージをやり取りすることで測定指示の送信および測定結果の受信を行う。これは VoIP 端末と共通のスタックを利用することで、VoIP 端末の状態を SIP メッセージにより得られるメリットがある。たとえば、QoS 管理サーバが VoIP 端末に測定指示を送信したとき、VoIP 端末が通話中であると、「486 Busy Here」レスポンスを QoS 管理サーバに返答する。これにより、QoS 管理サーバは VoIP 端末の状態を簡単な手続きにより得ることができる。また、品質測定は QoS 管理サーバ内で作成されたシナリオに基づいて、定期的に実行する。

(3) SIP サーバ

プロクシサーバおよび登録サーバを使用する。VoIP 端末間の呼の発着にはプロクシサーバ機能により、SIP URI による接続が行われる。また、登録サーバに用いられるデータベースには QoS 管理サーバがアクセスすることが可能であり、VoIP 端末の情報を得ることができる。

これらの構成要素をもとに、QoS 監視システムは図 5·1 のように動作する。VoIP 端末は 起動時に SIP サーバに SIP アドレスを登録する。QoS 管理サーバは SIP サーバに登録され ている VoIP 端末のリストを定期的に取得する。次に取得したアドレスのリストを元に試験 対象とする端末の選定や、品質測定スケジュールを決定し、試験シナリオを作成する。シ ナリオに従い、選定された VoIP 端末に品質測定指示を行う。VoIP 端末は品質測定終了後、 結果を QoS 管理サーバに送信する。



5.3 QoS 監視システムの実装

5.3.1 QoS 管理サーバ

QoS 管理サーバの実装にはプログラミング言語として Ruby、データベースサーバとして MySQL[4]を用いた。 QoS 管理サーバにおける状態遷移を図 5-2 に示す。

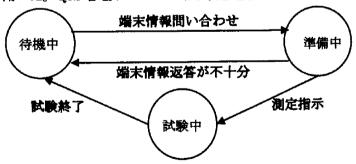


図 5-2.QoS 管理サーバの状態遷移

QoS 管理サーバにおける状態遷移を図 5-2 に示す。まず、待機中の状態において SIP サーバへ端末情報の問い合わせを行い、端末情報の一覧を得るとシナリオを作成する準備中の状態に移る。次に、得られた端末一覧からシナリオを作成し、そのシナリオに従い測定

を開始する。測定開始により試験中の状態に移る。シナリオどおりに試験が終了すると、 待機中に状態を移し、品質測定を終了する。

提案した品質測定システムを用いて品質測定を行う上で、測定対象とする VoIP 端末の SIP アドレスの一覧を取得してから測定シナリオを作成する。この測定シナリオでは、測定 を行う VoIP 端末の組み合わせと擬似音声を送受信する音声の送信時間を決める。 シナリオ が完成すると、選択された VoIP 端末に測定指示を送信する。指示を送信するときには SIP メッセージとして SUBSCRIBE メソッド[5]を用いた。 これは送信先に対してアクションを 要求する SIP メッセージである。VoIP 端末で測定が終了すると、NOTIFY メソッドを用い て QoS 管理サーバに測定結果を送信する。これは SUBSCRIBE と対をなすメッセージで あるり、SUBSCRIBE により要請されたアクションの結果を通知するメッセージである。 本システムにおいては測定指示に対する結果送信を意味する。図 5-3 に SUBSCRIBE、 NOTIFY メソッドの例を示す。

SIBSCRIBE sip:kkaneko@192.168.3.5 SIP/2.0

Via: SIP/2.0/UDP 192.168.3.20:5060;branch=x9hG4bK510

Max-Forward: 70

From: qosadmin <sip:qosadmin@192.168.3.20>;tag=FROM80563 To: kkaneko <sip:kkaneko@192.168.3.5>

Call-ID 267138685887 Cseq: 98 SUBSCRIBE

Expires: 300 Content-Type: text/plain Content-Length: 46

Body: kkaneko@192.168.3.5,koji@192.168.3.50,10

(a)SUBSCRIBE メソッド

NOTIFY sip:qosadmin@192.168.3.20 SIP/2.0 Via: SIP/2.0/UDP 192.168.3.20:5060;branch=x9hG4bK510

Max-Forward: 70

From: kkaneko <sip:kkaneko@192.168.3.5>;tag=to-6144

To: qosadmin <sip:qosadmin@192.168.3.20>;tag=FROM80563 Call-ID 267138685887

Cseq: 99 NOTIFY Expires: 300 Content-Type: text/plain Content-Length: 82

result:kkaneko@192.168.3.5,koji@192.168.3.50,0,0,10,Fri Mar 25 18:31:14 **JST 2005**

(b)NOTIFY メソッド

図 5-3.測定指示,結果送信メソッドの例

5.3.2 VoIP 端末

VoIP 端末として、第4章で作成した Linux 上で動作するソフトフォン「NU-phone」を 使用した。NU-phone に品質測定機能を加えた。このときの NU-phone の動作状態遷移図 を図 5-4 に示す。また、次の 2 つの場合には常に実際の通話を完全に優先して実行する。

・通話中に品質測定要求を受信した場合

・品質測定を実施中に発呼または着呼した場合

すなわち、通話中に品質測定要求を受信した場合には、即座に測定できないことをサーバ へ通知し、通話状態を維持する。品質測定中に発呼、または着呼した場合には品質測定を 即座に終了させ、サーバへ品質測定が中止されたことを通知する。

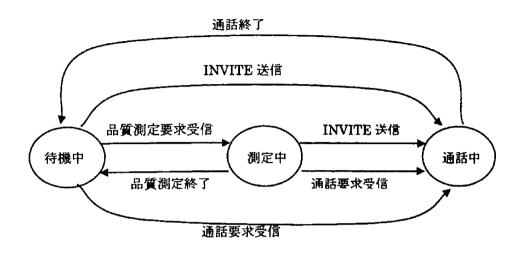


図 5-4.ソフトフォンの状態遷移

5.3.3 SIP メッセージの流れ

5.3.1、5.3.2 項で示した QoS 管理サーバ、VoIP 端末の状態遷移などを踏まえ、実際にやり取りされる SIP メッセージについて述べる。5.3.2 で述べたように、本システムでは通話を完全に優先する設計を行う。そのため、SIP メッセージのやり取りは

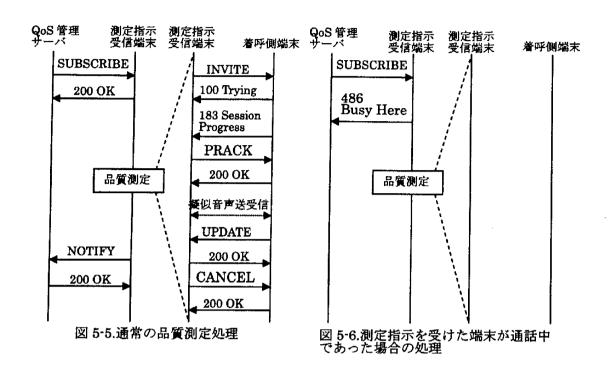
- ・通常の品質測定処理
- ・品質測定指示を受信した端末が通話中であった場合
- ・品質測定時における着呼側端末が通話中であった場合
- ・品質測定中に通話の発呼もしくは着呼要求があった場合

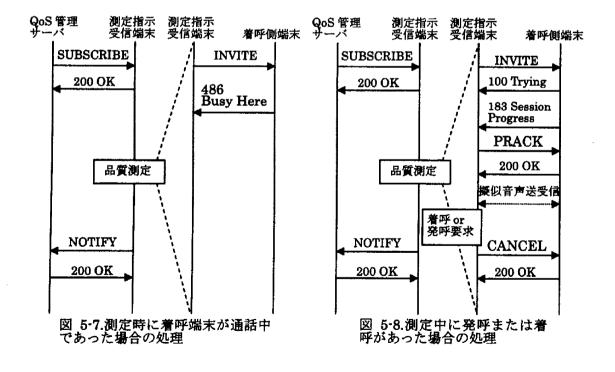
の4つの場合が考えられる。これらを実現する SIP メッセージの流れを図 5-5 から図 5-8 に示す。図 5-5 に示すように、VoIP 端末間での品質測定結果の送信には UPDATE メソッド[6]を用いる。これにより、着呼側端末で得られた品質測定情報が測定指示を受けた VoIP 端末に集約し、QoS 管理サーバへ登録することが可能である。図 5-9 に UPDATE メソッドの例を示す。

図 5-6、図 5-7 は、発呼端末あるいは着呼端末が話中であるときの SIP メッセージのやり とりである。このとき、通話を優先させるため、話中である「486 Busy Here」を返信する。 また、着呼端末が話中のときは、発呼端末が NOTIFY メソッドにより QoS 管理サーバに測 定できないことを報告する。

図 5-8 は、測定中に発呼あるいは着呼要求がどちらかの端末が送受信した場合の SIP メッセージのやりとりを示す。要求があった端末が CANCEL メソッドを送信することで測定

を中止する。発呼端末は測定中止になったことを NOTIFY メソッドにより QoS 管理サーバ に報告する。





UPDATE sip:kkaneko@192.168.3.5 SIP/2.0

Via: SIP/2.0/UDP 192.168.3.5:5060;branch=z9hG4bK-264

Max-Forwards: 70

From: koji <sip:koji@192.168.3.50>;tag=to-665

To: kkaneko <sip:kkaneko@192.168.3.5>;tag=from-254469 Call-ID: 794939093181

CSeq: 101 UPDATE

Contact: <sip:koji@192.168.3.50>

Content Type: text/plain

Content-Length: 50

Body: 0,kkaneko@192.168.3.5->koji@192.168.3.50

図 5-9.UPDATE メソッドの例

5.4 QoS 管理システムの評価実験

5.4.1 測定条件

5.2 節で述べた方法により実装した VoIP 端末、および QoS 管理サーバの動作確認を実験 ネットワークにて行った。実験ネットワークを図 5-10 に示す。ルータ間の帯域を 1Mbps に設定し、両端のネットワークに VoIP 端末、負荷発生装置を設置している。QoS 管理サー バは測定指示を受ける VoIP 端末を設置しているネットワークに接続している。

2 台の VoIP 端末間で擬似呼を生起し、10 秒間の擬似音声を送受信する。このとき、VoIP 端末では、擬似呼の接続時間および、音声パケットのパケット損失率を測定する。負荷発 生装置では、VoIP を想定した負荷を発生させる。この負荷において、到着率はポアソン分 布に従い、呼量は 5erl、発生時間は 1800 秒間である。さらに負荷発生から 600 秒後、さら に負荷を加える。この負荷も到着率はポアソン分布に従い、呼量は 6erl、発生時間は 600 秒間である。すなわち、実験開始後 600 秒から 1200 秒の間は 11erl の負荷がかかる。QoS サーバからの測定指示は、150sec 間隔で発呼側 VoIP 端末に送信される。

擬似音声フローおよび背景フローの条件を表 5.1 に示す。

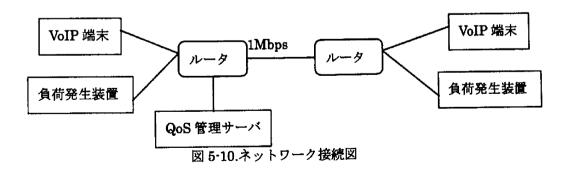


表 5-1.測定条件

擬似音声フロー条件

シグナリングプロトコル	SIP
VoIP パケット転送プロトコル	RTP
VoIP パケットサイズ	214[byte]
擬似音声送信時間	10[sec]

背景負荷条件

呼量	5~1lerl
	測定開始から終了まで 5erl
	測定開始から 600~1200sec では
	さらに 6erl 加える
パケットサイズ	214[byte]

測定条件

測定時間	1800[sec]
測定指示間隔	150[sec]

5.4.2 測定結果

図 5-11 から図 5-13 に品質監視システムにおける測定対象の測定結果を示す。図 5-11 より、背景負荷の増加により、ネットワークが輻輳状態となりパケット損失率が増加する。600 秒から 1200 秒の輻輳状態の時に、パケット損失率が減少しているときがある。これは、試験フローを送受信しているときに、背景フローのパケット損失が増加しているためである。そのため、輻輳状態にもかかわらず、試験フローのパケット損失率が小さい。 擬似音声のパケット損失率が増加すれば、元の音声を復元することができない。そのため、図 5-12 に示すように PESQ MOS 値が減少する。パケット損失率が増加すると、MOS 値が減少している。その一方で、パケット損失率が小さくても MOS 値が小さくなることがある。これは、MOS が人間の音の聞き方を反映しているためである。人間が音を聞く場合、連続して音が途切れるよりも、継続してブツブツといった音の途切れが発生するほうが、不快に感じることが知られているためである。この特性を考えると、パケット損失が小さくても MOS 値が小さくなるのは、送受信している間は継続してパケット損失が発生していることがわかる。そして、パケット損失が大きいが MOS 値が大きくなるのは、パケット損失の発生の代方がバースト的に連続して発生していることがわかる。

図 5-13 の擬似音声の送受信を行うまでの接続時間においては、背景負荷の増加に伴い、接続時間も長くなる。これは、パケット損失率と同様にネットワーク QoS を把握する要素の1つとなる。

この実験では、10 回行った結果の平均をとった。このとき、パケット損失率が非常に大きくなることがあった。これは、帯域 1Mbps では 10 フロー入るところを、10 フロー以上の負荷をかけている。そのため、SIP メッセージも損失することがある。SIP メッセージの

損失が生じると、SIP メッセージである INVITE や PRACK の再送処理が発生する。再送により、擬似音声を送受信するタイミングがずれ、送受信を始めた数秒間はパケットを受信できない状態になっていた。

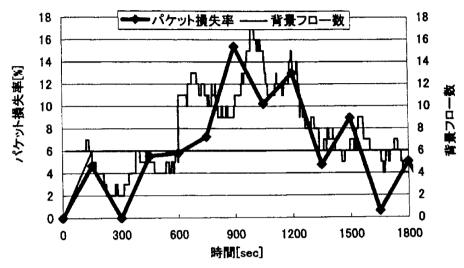


図 5-11.擬似音声フローのパケット損失率

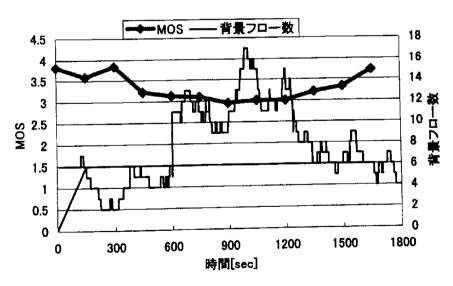
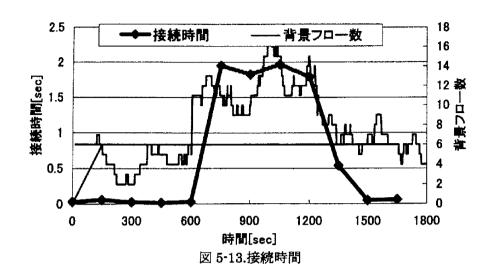


図 5-12.擬似音声の MOS



5.5 まとめ

提案したエンドユーザ間のネットワーク品質を直接測定する QoS 監視システムの実装の詳細と、実装した QoS 監視システムについて述べた。まず、QoS 監視システムの実装にあたり、第4章において作成したソフトフォン NU-phone に品質測定機能を追加し、QoS 監視システムを実験ネットワークにおいて評価した結果、ネットワークの状況に即したエンド・エンドのデータを得られることを確認した。また、輻輳状態においては、擬似音声のパケット損失率が増加すると同様に、接続時間も増加することがわかった。

本稿で提案したシステムを使用することにより、測定機器の設置場所を設計することなくエンドロエンドのデータを測定することができる。しかし、品質測定を行う上で、測定ポイントの選定や測定時間などの測定計画を設定する必要がある。測定計画が的確でなければ、実際にネットワークで起きている必要なデータを得られないためである。このため、ネットワークの状態の測定を的確に行うための測定計画作成アルゴリズムの検討を行う必要がある。

参考文献

- [1] 間瀬憲一, "マルチメディアネットワークとコミュニケーション品質", **電子情報通**信 学会, 1998
- [2] 鈴木他, "ユーザ品質評価のための VoIP ネットワーク品質管理手法の提案", 2004 年 電子情報通信学会総合大会, B-11-19, p. 512, 2004
- [3] 高橋他, "VoIP ネットワークの End-to-End 品質測定手法の実現", 2005 年電子情報通信学会総合大会, B-11-31, p. 435, 2005
- [4] http://www.mysql.com
- [5] A.B.Roach, "Session Initiation Protocol(SIP) Specific Event Notification", RFC3625, June 2002
- [6] J.Rosenberg, "The Session Initiation Protocol(SIP) UPDATE Method", RFC3311, June 2002

6 ボトルネックリンク速度推定法

6.1 背景

インターネットの実態は管理組織の異なる様々なネットワークの集合体である。インターネットの利用者は特定のインターネットサービス事業者とのみ契約するものの、インターネット利用中の通信経路は、通常、契約事業者以外のネットワークにもまたがって形成される。通信中に何らかのトラブルが発生し、利用者がその契約事業者に調査を依頼した場合、トラブルの内容によっては、契約事業者はエンドツーエンドの通信経路上に存在する他事業者のネットワークの正常性を調査する必要が発生する。しかし、他事業者のネットワークを直接覗くことは、通常、不可能であるため、何らかの間接的な方法でその正常性を判断せざるを得ない。このような背景から、ネットワークの内部状態を間接的に計測・推定する技術が最近注目されている。

本章では、上記技術のうち、端末間の通信経路上のボトルネックリンクの物理速度(帯域)を遠隔推定する技術に着目する。リンクの物理速度を推定する技術は、上述したようにネットワークのトラブルシューティング上重要であると同時に、例えば、ネットワーク内に分散する複数のプロキシー/コピーサーバから最適なサーバを選択する際に、または動画像等をストリーミング配信する場合の配信レイトを設定する際に利用することができる。そのため、これまでに、[1]-[3]等、数多くの推定原理・ツールが提案されているが、いずれも推定精度等に幾つかの問題点を抱えている。本章では、これら既存手法の長所を踏まえながら、より優れた推定精度を有すると考えられる手法を新たに提案するとともに、実際にLinux上で動作するツールを作成し、幾つかのネットワーク実験を通して、提案手法の有効性を示す。

6.2 Packet-Pair推定法

通信経路上のボトルネックリンクの物理速度を測定する原理の代表例に「Packet-Pair推定法」がある。本節ではPacket-Pair推定法について簡単に説明する。

6.2.1 基本原理

時刻0に、二つの同サイズのパケットをあるホストから別のホストに向けて連続送信し、各パケットが途中ルータに到達する時刻を考える。仮に、クロストラヒックがないとすると、両ホストを結ぶ通信経路上のボトルネックリンクの手前で、後行パケットは先行パケットに続いてバッファリングされるため、ボトルネックリンク以降、両パケットの到着時刻は後行パケットのボトルネックリンクでの処理時間分、拡がる。従って、以下の式が(近似的に成立する。

$$T_n^2 - T_n^1 \cong \frac{s}{\min\{b_1, \dots b_{n-1}\}}$$
 (6-1)

ここで、 T_n^1 と T_n^2 は先行パケットおよび後行パケットのn番目のルータへの到達時刻、 b_k は通信経路を構成するk番目のリンクの物理速度(帯域)、またsはパケットサイズを表す。つまり、Packet-Pair推定法は、何らかの方法により T_n^1 と T_n^2 (もしくは式(6-1)と同様の関係が成立する物理量)を測定し、その差でパケットサイズを割ることで、ボトルネックリンクの物理速度を推定するものである。

6.2.2 Packet-Pair推定法の改良

Packet-Pair推定法の原理となる式(6-1)は、パケットのバッファ待ち遅延がない場合に限り成立する。実際には、パケットペアの到達時間の差分にはバッファ待ち遅延が含まれるため、Packet-Pair推定法にはこれに起因する推定誤差が発生する。

このため、Packet-Pair推定法の改良案が幾つか提案されてきた。Carterと Crovellaは Packet-Pair推定法によりボトルネックリンク速度の推定値が分布する範囲を複数回求め、例えば分布範囲がオーバラップする箇所を抽出し、最終的な推定値とするツール (Bprobe)を提案している[4]。LaiとBakerは、Packet-Pair推定法によるボトルネックリンク速度の推定値の分布密度関数を求め、分布密度関数の最大ピークの場所を推定結果として与えるツール (Nettimer) を提案している[3]。一方、Dovrolis等は、Nettimerに類似した推定技術を利用したクライアントーサーバ型のツール (Pathrate) を提案している[5]。これらは、Packet-Pair推定法で得られた複数の推定結果から真の推定値を抽出する手法に工夫をこらしたものであるが、いずれもヒューリスティックな方法論に頼っており、必ずしも明快な理論的な裏付けを持たない。

6.3 提案ツールの概要

本節では、今回提案するボトルネックリンク物理速度推定ツール(以下RateTracerと呼ぶ)の概要を述べる。

6.3.1 基本原理

RateTracerは、基本的には、(Pathcharの考え方を取り入れて) Packet-Pair推定法の原理を改良し、精度を高めた推定ツールである。以下、基本原理を説明する。時刻0に、二つの同サイズのパケットをあるホストから別のホストに向けて連続送信した場合の、各パケットの途中ルータへの到達時刻を考える。以下の変数を導入する。

- d_k : k番目のルータとk+1番目のルータ間の伝搬遅延+k番目のルータにおける (アドレス 検索等の) パケットのフォワーディング処理に伴う遅延 (固定値)
- $Q_{\mathbf{k}}^{1}$:先行パケットの \mathbf{k} 番目のルータでのバッファ待ち遅延(確率変数)
- Q_k^2 :先行パケットと後行パケットの間に挿入された別のパケットにより生ずる、後行パケットのk番目のルータでのバッファ待ち遅延(確率変数)

 \tilde{O}_{k}^{2} :先行パケットにより生ずる、後行パケットのk番目のルータでのバッファ待ち 遅延 (確率変数)

ルータはパケットをストア&フォワード処理し、一方、ルータ間のリンクにはパケットをス トア&フォワード処理する(レイヤ2以下の)ノードが存在しないとすると、以下の式が成 立する。

$$T_n^1 = \sum_{k=0}^{n-1} \left(\frac{s}{b_k} + d_k + Q_k^1 \right), \quad T_n^2 = \sum_{k=0}^{n-1} \left(\frac{s}{b_k} + d_k + Q_k^2 + \widetilde{Q}_k^2 \right)$$
 (6-2)

ここで、次が成立する[6]

$$\widetilde{Q}_{k}^{2} = \max\{0, T_{k+1}^{1} - d_{k+1} - T_{k}^{2}\}. \tag{6-3}$$

 ${\widetilde Q_k}^2 = \max\{0, T_{k+1}^1 - d_{k+1} - T_k^2\}.$ (6-3 Q_k^1 及び Q_k^2 はクロストラヒックの影響で発生する項であるが、その最小値は0に等しいこ とから.

$$\min\{T_n^1\} = \sum_{k=0}^{n-1} \left(\frac{s}{b_k} + d_k\right), \quad \min\{T_n^2\} = \sum_{k=0}^{n-1} \left(\frac{s}{b_k} + d_k + \widetilde{Q}_k^2\right)$$
 (6-4)

式(6·3)と(6·4)から、次が導かれる[Lai2000]

$$\min\{T_n^2\} - \min\{T_n^1\} = \frac{s}{\min\{b_1, \dots b_{n-1}\}}$$
(6-5)

式(6-5)は、クロストラヒックに影響されない(つまり、先行パケットにバッファ待ち遅 延が発生せず、先行パケットと後行パケットの間に別のパケットが挿入されない)場合に、 $T_n^1 \geq T_n^2$ はその最小値を取り、さらにその場合には、Packe-Pair推定法が仮定する関係式(式 (6-1)) が厳密に成立することを示している。次節で述べるように、本論文で提案するツー ルは、 $T_n^1 \geq T_n^2$ (に類似した量)をそれぞれ複数回測定し、複数回の測定結果の最小値を式 (6-5)に代入してボトルネックリンク速度を推定することで、通常のPacket-Pair推定法の精 度を改善させたものである。

6.3.2 実現方法

式(6-5)の原理に基づいてボトルネックリンク速度推定ツールを実現する方法には、幾つ かの選択肢が存在する。本研究では、利便性を考慮し、単体ホストで動作するツールとし て実装することとした。すなわち、試験パケットをペアで連続送信し、それぞれのパケッ トに対するネットワークからの応答 (ICMPメッセージ) の受信時刻の差から、式(6-5)によ りボトルネックリンク速度を推定する形態である。

単体ホスト上のツールとしての実装方法にも複数の選択肢が存在するが、本研究では以 下のようなTracerouteベースの方式[7]を採用した。まず、ツールを実装したホストから、 同サイズのUDPパケットをペアで連続送信する。UDPパケットのTTLはいずれも同じ値 (例えばn) に設定されており、従ってnホップ目のルータからそれぞれのパケットに対し てICMP Time Exceeded Message (エンドホストからはICMP Port Unreachable

Message) が返される。UDPペアパケットの連続送信をnに固定して例えば100回行い、(100回中の) 先行UDPパケット及び後行UDPパケットを送ってから、各UDPパケットに対する ICMPメッセージを受信するまでの時間の最小値をそれぞれ求める。次いで、得られた最小値を式(6-5)に代入し、ホストとn番目のルータ間のボトルネックリンクの物理速度を推定する。この動作をn=1から順次nの値を増やして繰り返すことで、ボトルネックリンクの物理速度のみならずボトルネックリンクの存在箇所も同時に特定することができる。

なお、試験パケットのサイズはパス MTU に設定することが望ましいが、パス MTU が判るとは限らないため、本ツールでは試験パケットのデフォルト値を (IP ヘッダ込みで) 1500 バイトに設定している。

6.4 提案ツールの特性評価

本章では、提案ツールの推定精度等を検証するために実施した、ネットワーク実験の概要とその結果について述べる。

6.4.1 実験概要

図6-1に示すように、千葉大学内のホストAと大学外に置かれた3台のノード(以下、C1、C2、C3とする)間の経路上のボトルネックリンク速度をPathchar、Nettimer、RateTracerにより測定した。C1は一般のインターネットサービスプロバイダーにADSL回線(下り12Mbps、上りは約700kbps)により接続されたホスト、C2はAbiline(Internet2の基幹伝送網:米国)内のルータ、C3はGEMnet(NTTが保有する国際実験ネットワーク)内のホストである。千葉大学内のリンク速度は全て100Mbpsであり、Router1とRouter2の間には、一部、ATM回線が使われている。一方、図6-1のSINET及びAbilene内の全てのリンク速度は100Mbps以上である。

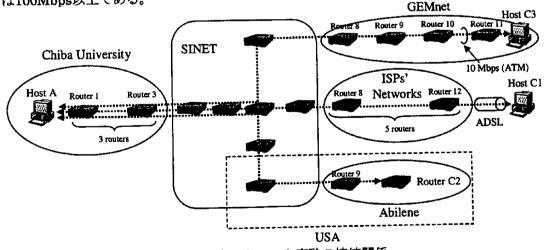


図6-1 ネットワーク実験の接続関係

また、GEMnet内の一部(Router 10-Router 11)に人為的にATMによる10Mbpsのボ

トルネックリンクを形成した。GEMnet内のRouter 9-Router 10間も110MbpsのATM回線であり、Router 11-Host C3間はFast Etherである。一般のサービスプロバイダー内のリンク速度は不明である。

6.4.2 実験結果:A-C1間

ホストA-C1間のボトルネックリンク速度の10回の推定結果の平均値を表6-1に示す。表には、経路上の各ルータとホストA間のボトルネックリンク速度の推定結果およびRTTも併せて記した。なお、Pathcharは各リンクの物理速度を直接計測するツールであるため、Pathcharの欄には、Pathcharで計測したそれぞれリンクの物理速度のうち最小のリンク速度をボトルネック速度として表示した。更に、Pathcharでは、6ホップ目以降一部の経路(Router 6-Router 7、Router 7-Router 8、Router 10-Router 11)の物理リンク速度が計測できなかったため、6ホップ目以降については、計測不能であったリンクを除く最小リンク速度を暫定的なボトルネックリンク速度としてカッコ付きの数値で示した。Nettimerについては、最終ホップを除く8ホップ目以降のボトルネックリンク速度は計測できなかったため、該当欄には数値を記載しなかった。また、「actual bottleneck link bandwidth」の欄には、各ホップまでのボトルネックリンク速度を表示したが、8ホップ以降は実際のボトルネックリンク速度が(100Mbps以下であるという点を除き)わからないため、「(\leq 100)」と表示した。

patk	Host A	Host A	Host A se Router 3	Host A an Rounts 4	Host A	Host A Bouner 6	Host A ⇒ Roster 7	Host A	Host A ⇒ Roster 9	Host A	Host A se Router 11	Host A ⇒ Router 12	Host A ⇒ Host C1
RateStraces (Mbps)	97.3	160.7	86.2	85.5	98.9	98.2	113.8	104.8	99.8	88.9	115.5	50.8	11.9
Pathcher (Mispa)	49.0	15.0	15.0	15.0	15.0	(15.0)	(15.0)	(15.0)	(15.0)	(15.0)	(15.0)	(15.0)	(3.0)
National [Mispa]	35.5	41.3	46.5	28.9	34.7	27.2	44.3			_	-		4.9
Actual bothstack link bandwidth [Mpu]	100	100	100	100	100	100	100	(≦100)	(≦100)	(≦100)	(≨100)	(≦100)	12
RTT(ou)	0.791	2.65	2,98	3.22	4.06	4.83	4.65	5.12	5.19	5.28	5.50	5.70	28.9

表6·1 ホストAールータC1間のボトルネックリンク速度の推定結果

ホストAーC1間のボトルネックリンクはADSL回線の部分であり。特に、 $A\rightarrow$ C1方向の経路上のボトルネックリンクの速度は12Mbpsである。RateTracerはこのボトルネックリンク速度を正確に推定しているが、PathcharとNettimerによる推定値は真値の半分以下の値である。

ホストAから8ホップ目 (Router 8) までの経路のボトルネックリンク速度は100Mbpsである。この8ホップ目までについて見ると、RateTracerによる推定結果は、Host A→Router 2を除き、おおむね正確である。これに比べて、他の二つのツールの推定精度は落ちる。特に、既に述べたように、Pathcharでは一部の経路 (Router6-Router7、Router7-Router8、Router10-Router11) のリンク速度が計測不能であり、Nettimerについても、最終ホップ

を除く8ホップ目以降は計測不能であった。

Nettimerが8ホップ目以降計測できない原因を分析するため、単純なPacket-Pair推定法によりHost A-Router 8間のボトルネックリンク速度を400回推定し、推定結果のヒストグラムを図6-2に示した。推定結果は広範囲に分布する。また、240Mbps付近にヒストグラムのピークが存在するが、それ以外には特徴的なピークは見当たらない。このように、バッファ待ち遅延の影響により真値は殆どマスクされてしまい、(Nettimerのように)ヒストグラムの情報からボトルネックリンク速度に関する情報を抽出するのは実際には困難であることが確認できる。

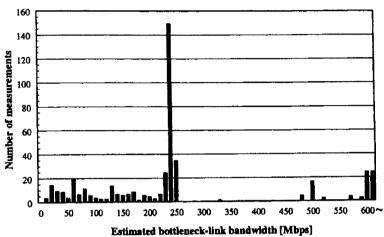


図6-2 Packet-Pair推定法による推定結果のヒストグラム (ホストAールータ8)

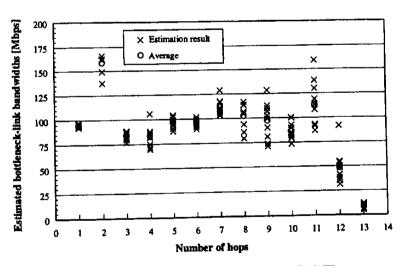


図6-3 RateTracerによる推定結果の散布図

図6-3には、各ホップまでのボトルネックリンク速度のRateTracerによる10回の推定結果の 散布図を示した。7ホップ目までは推定結果は比較的安定しているが、8ホップ目以降は推 定結果の分散が大きくなる。これは、ホップ数が増えると、クロストラヒックの影響を充分フィルタリングできなくなるためと思われる。なお、最終リンク (ボトルネックリンク)の物理速度の推定結果は安定している。

6.4.2 実験結果:A-C2間

表6-2はホストA-ルータC2間のボトルネックリンク速度の10回の推定結果をまとめたものである。このように、国際回線を含む経路についてもRateTracerの推定結果は概ね良好であった。Nettimerは全てのホップについて計測できたが、推定結果は真値のほぼ半分であった。Pathcharによる推定結果はこれらよりさらに落ちる。特に、PathcharではRouter7-Router8間のリンク速度が計測できず、それ以降のリンクについても測定が成功したのは10回中2~3回程度であった。

peth	Host A⇒ Router 1	Host A⇒ Router 2	Host A⇒ Router 3	Host A⇒ Router 4	Host A⇒ Router 5	Host A⇒ Router 6	Host A⇒ Router 7	Host A⇒ Router 8	Host A⇒ Router 9	Host A⇒ Router C2
RateTracer [Mbps]	106.7	149.3	93.1	121.5	98.2	101.6	97.2	116.2	98.7	101.7
Pathchar [Mbps]	31.0	15.0	15.0	15.0	15.0	15.0	15.0	(15.0)	(15.0)	(15.0)
Nettimer [Mbps]	36.9	40.8	40.6	24.4	29.1	33.0	45.2	48.5	45.6	46.3
Actual bottleneck link bandwichh [Mbps]	100	100	100	100	100	100	100	100	100	100
RTT (ms)	0.798	2.62	2.98	3.27	4.10	4.89	4.90	180.4	195.0	201.4

表6·2 ホストA-ルータC2間のボトルネックリンク速度の推定結果

6.4.3 実験結果:A-C3間

表6-3はホストAールータC3間のボトルネックリンク速度の10回の推定結果をまとめたものである。ホストAーC3間には最終ホップの一つ手前(Router 10ーRouter 11)に10MbpsのATM回線があり、これがホストAーC3間のボトルネックリンクとなる。

peth	Host A⇒ Router i	Host Ass Router 2	Host A⇒ Router 3	Host A⇔ Router 4	Host Am Router 5	Host A⇒ Router 6	Host A⇒ Router 7	Host A⇒ Router S	Host A= Router 9	Host A⇒ Router 10	Host A≕ Router II	Host A#
RateTracer [Mbps]	105.1	153.8	94.6	121.8	98.3	100.5	109.2	109.6	105.3	126.9	14.5	11.6
Pathchar (Mbps)	31.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0	7.1	7.1
Nettimer (Mbps)	34.9	41.9	42.0	26.3	27.7	30.0	26.6	31.3	45.6	44.3	4,4	23.5
ictual bottlennek link bandwidth [Mbps]	100	100	100	100	100	100	100	100	100	100	10	10
RTT (ms)	0.794	2.62	2.99	3.28	4.12	4.91	5.00	5.57	5.81	6.54	11.9	9.83

表6-3 ホストA-ルータC3間のボトルネックリンク速度の推定結果

各ツールによるホストA-C3間のボトルネックリンク速度の推定値を比較すると、 Nettimerの推定値(23.5 Mbps)の誤差が最も大きい(真値の倍以上)。一方、RateTracer は真値より若干大きい推定値(11.6 Mbps)、Pathcharは逆に真値より若干小さい推定値 (7.1 Mbps)を与える。ATMヘッダを考慮すると、実質的なボトルネックリンク速度は 9Mbps程度であり、その意味で、RateTracerよりはPathcharの推定値の方が若干正確である。また、ホストAーRouter12間についても、Pathcharの方がRateTracerよりも正確な推定値を与えている。このように、RateTracerは特にATM回線のリンク速度を若干大きめに 誤推定する傾向が認められる。ただし、Router 11およびその手前のルータまでの経路については、ホストAーRouter2間を除き、RateTracerが最も正確である。なお、ホストAーRouter 11間のRTTが、ホストAーC3間のRTTを上回っているのは、Router 11でのICMP 処理に時間がかかっているためと思われる。

6.5 まとめ

本章では、試験パケットを用いて、end-to-end経路上のボトルネックリンクの物理速度を推定するツール(RateTracer)を提案し、ネットワーク実験によりその有効性を検証した結果を示した。我々の提案は、Packet-Pair推定法の単純な改良であり、実装も容易である。また、幾つかの既存ツールに比べて、より優れた推定精度を持つことが確認できた。従って、提案ツールは、例えばトラブルシューティングの際の有望なツールとして充分期待できると思われる。今後の課題としては、利用する試験パケット量の軽減化、クライアントーサーバ型ツールとしての実装等がある。

参考文献

- [1] V. Jacobson, "pathchar," ftp://ftp.ee.lbl.gov/pathchar.
- [2] B. Mah, "pchar: A tool for measuring internet path characteristics," http://www.ca.sandia.gov/bmah/Software/pchar.
- [3] K. Lai, and M. Baker, "Nettimer: A tool for measuring bottleneck link bandwidth," Proc. USENIX Symposium on Internet Technologies and Systems, 2001.
- [4] K. Lai, and M. Baker, "Measuring link bandwidths using a deterministic model of packet delay," ACM SIGCOMM '00, pp.283--194, 2000.
- [5] R. Carter, and M. Crovella, "Measuring bottleneck link speed in packet-switched networks," Performance Evaluation, vol.27& 28, pp.297-318, 1996.
- [6] C. Dovrolis, P. Ramanathan, and D. Moore, "What do packets dispersion techniques measure?," IEEE INFOCOM, 2001.
- [7] W. Stevens, UNIX Network Programming, Networking APIs: Sockets and XTI Volume 1, Prentice Hall, 1998.

以下に文献をまとめて付録とし、成果をより詳細に示す。

•	間瀬憲一: "インターネットにおけるスケーラブルなアドミッション	/コントロ	ュール方
	式,"電子情報通信学会誌, vol. 85, no. 9, pp. 655-661, Sept. 2002.		付鍋1

- 間瀬憲一, 栗林孝行, 津野昭彦: "QoS 統計データを利用した動的サーバ選択法," 電子情報通信学会論文誌(B), vol. J86-B, no. 3, pp. 1-12, March 2003.・・・・・・・・ 付録 8
- A. Bilhaj and K. Mase: "Endpoint Admission Control Enhanced Systems for VoIP Networks," IEICE Transaction on Communications, vol. E87-B, no. 4, pp. 948-956, April 2004.

・・・・・ 付録 20

- K. Mase: "Toward Scalable Admission Control for VoIP Networks," IEEE Communications
 Magazine, vol. 42, no. 7, pp. 42-47, July 2004.
- 八木敬宏,塩田茂雄,間瀬憲一: "ボトルネックリンク速度推定ツールの提案と精度検証,"電子情報通信学会論文誌(B), vol. J87-B, no. 10, pp. 1636-1647, Oct. 2004.

…… 付録 36

- K. Mase and Y. Toyama: "End-to-End Measurement Based Admission Control for VoIP Networks," 2002 IEEE International Conference on Communications (ICC2002) Conference Proceedings, vol. 2, pp. 1194-1198, May 2002.
- X. Kong and K. Mase: "Dynamic Routing with Endpoint Admission Control," 2003 IEEE International Conference on Communications (ICC2003) Conference Proceedings, vol. 3, pp. 1728-1732, May 2003.
- K. Mase and H. Kobayashi: "An Efficient End-to-End Measurement Based Admission Control for VoIP Networks," 2004 IEEE International Conference on Communications (ICC2004)
 Conference Proceedings, vol. 4, pp. 1877-1881, June 2004.
- S. Shioda and K. Mase: "A New Approach to Bandwidth Requirement Estimation and its Accuracy Verification," 2004 IEEE International Conference on Communications (ICC2004)
 Conference Proceedings, vol. 4, pp. 1953-1957, June 2004.
- Shigeo Shioda, Takahiro Yagi and Kenichi Mase: "A New Approach to the Bottleneck Bandwidth Measurement for an End-to-End Network Path," 2005 IEEE International Conference on Communications (ICC2005) Conference Proceedings, vol. 1, pp. 59-64, May 2005.
- Shigeo Shioda and Kenichi Mase: "Performance Comparison between IntServ-Based and DiffServ-Based Networks," 2005 IEEE Global Telecommunications Conference (GLOBECOM 2005), vol. 1, pp. 529-534, Dec. 2005.