モビリティ環境におけるスケーラブルな通信品質保証機構

A Scalable Communication System for QoS Support in Mobility Environment

萩原 茂明　　市川 隆浩*　　寺岡 文男
慶應義塾大学 大学院 理工学研究科

概 要

モビリティ環境に適応したスケーラブルな QoS 保証機構である QoSMoS を提案する。スケーラビリティの実現のため QoSMoS は Diffserv をベースとし、モバイル環境に適応するため QoSMoS は通信に先立って Admission control を行う。また、高速なハンドオーバを実現し、さらにハンドオーバ後も QoS を保証するため、QoSMoS はクロスレイヤ情報交換を利用してハンドオーバを事前に検知し、Pre-Admission control を行う。さらに突発的な輻輳に対処するため、QoSMoS は Traffic Manager (TM) を導入し、輻輳に関係するトラフィックの両 End Node に送信量の抑制を要求する。本論文は QoS の中で帯域保証に焦点を絞り、QoSMoS を FreeBSD 上に実装した。そして処理のオーバヘッドおよび帯域保証の有効性を検証した。その結果、処理のオーバヘッドは無視できる程度であり、帯域保証が実現できることがわかった。

1　はじめに

インターネットにおけるストリーミングアプリケーションの普及が進んでいる。たとえば YouTube に代表される動画配信や Skype に代表される音声通話などである。さらに近い将来には地上デジタル放送もインターネットを通じて配信されるようになるであろう。一方インターネットにおけるモビリティ環境では、Mobile IPv4/v6 などのモビリティサポートプロトコルの標準化や無線通信環境の広帯域化、携帯端末の高機能化により、モビリティ環境においてもストリーミングアプリケーションを利用したいという要求が高まっている。すなわちモビリティ環境における通信品質保証への要求がますます高まるものと思われる。

インターネットにおける通信品質保証の方法には大きく分けて Intserv と Diffserv がある。Intserv はフローごとの通信品質保証を実現するため、ルータはフローごとの状態を保持する必要がある。そのためスケーラビリティに問題があるとされている。Intserv をモビリティ環境に適応させた研究 [1] もあるが、スケーラビリティやハンドオーバへの対処が不十分である。一方 Diffserv はフローを少数のクラスに分けてクラスごとの品質保証を実現するため、ルータはフローごとの状態を保持する必要がなくスケーラビリティに優れていると言われているが、フローごとの品質は厳密には保証できない。Diffserv では、Diffserv を提供する Domain (DS Domain) とユーザ間で SLA (Service Level Agreement) と呼ばれる契約を結び、DS Domain は各ユーザとの SLA に基づいてネットワーク資源を準備する。このように Diffserv は通信経路が静的に決まる環境を想定しているが、モビリティ環境においては移動ノードのインターネットへの接続点は動的に変わるので、SLA による静的な制御では不十分である。さらに移動ノードがハンドオーバした後も品質保証を実現する必要がある。また、移動ノードがインターネットに接続するための無線アクセスネットワーク部分は本来 DS Domain には含まれないが、この部分における品質保証も重要となる。Diffserv に基づき、モビリティ環境において品質保証を実現するための研究 [2, 3, 4, 5, 6] もあるが、上記の問題点をすべて解決しているものはない。

本論文はモビリティ環境に適応したスケーラブルな品質保証機構である QoSMoS を提案する。QoSMoS は通信に先立って Admission control を行うことにより、移動ノードからの品質保証要求に対応する。またクロスレイヤ情報交換を利用してハンドオーバを事前に検知し、Pre-Admission control を行うことで高速なハンドオーバおよびハンドオーバ後の品質保証を実現する。さらに突発的な輻輳に対処するため、Traffic Manager (TM) を導入し、輻輳発生時には関係する End Node に送信量の抑制を依頼する。本論文では品質保証の中で帯域保証に焦点を当て、QoSMoS を FreeBSD 上に実装した。本論文では QoSMoS の処理オーバヘッドや帯域保証の有効性についても述べる。

2　関連研究

論文 [1] では，RSVP を改良しハンドオーバに対応させたシグナリングプロトコル HO-RSVP を提案している．HO-RSVP はハンドオーバを検知した時に，ハンドオーバ後の

*現 NTT 東日本

パスに対して事前に Admission Control をするため，シグナリング処理を行う．しかし，保持情報の過多によるスケーラビリティの欠如という従来の RSVP の欠点は改善されていない．

そこで，スケーラビリティを考慮して，DiffServ に基づいた論文に着目する．論文 [2] では，Diffserv を拡張して，Edge Router の下流に複数台の Access Router を配置し，その中でのハンドオーバについての QoS 保証を提案している．また，論文 [3] では，階層的に Quality Agent (QA) を配置し，QA 間で ICMP メッセージをやりとりすることで QoS を実現している．論文 [4] では，ハンドオーバ前に Call Admission Control (CAC) を実行しておき、実際のハンドオーバ時にその要求を Activate することで，QoS 再開の時間短縮を図っている．しかし，これらの論文 [2, 3, 4] では，コアネットワーク部分の QoS 保証については考慮されていない．QoSMoS では，ハンドオーバによるコアネットワーク部分の経路変更も考慮した QoS 保証を提案している．

論文 [5] では，Diffserv をベースとして，ICMP シグナリングを利用することによってモビリティ環境に対応できるようにしている．また，新しく接続してきたモバイルノードに対する通信帯域を一時的に確保することにより，ハンドオーバ時の通信品質悪化を低減させることを提案している．また，論文 [6] では，hop-by-hop の Admission control を提案している．しかし，これらの論文 [5, 6] では，共に無線アクセスネットワーク部分の QoS を考慮していない．論文 [7] では，Intserv と Diffserv の双方で利用可能な Admission control protocol を設計している．しかし，ハンドオーバ時の QoS が考慮されていないという問題がある．QoSMoS では，無線アクセスネットワーク部分の QoS やハンドオーバ前後の QoS も考慮している．

3 QoSMoS のアーキテクチャ

3.1 基本アーキテクチャ

QoSMoS はスケーラビリティを考慮して，DiffServ を基にしたアーキテクチャになっている．QoSMoS では，DiffServ と同様に Edge Router (ER) が多くの処理をすることで，Core Router (CR) の保持情報を最小限にしている．図 1 に QoSMoS のアーキテクチャを示す．

QoSMoS のアーキテクチャは 1 つの Domain を構築する ER，CR，および Traffic Manager(TM) と，そのサービスを利用する End Node からなる．End Node は，Access Point(AP) に接続する Mobile Node (MN) も考慮している．QoSMoS は IPv6 を用いて Domain 内におけるハンドオーバのみを扱い，ER が Access Router の役割を果たすものとする．また，本研究における QoSMoS は，帯域のみを考慮した QoS を保証するものとする．

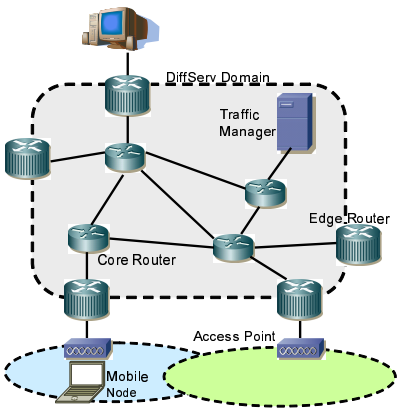

図 1: QoSMoS のアーキテクチャ

3.2 IPv6 ヘッダの利用

QoSMoS の CR では DSCP を利用してパケットのクラス分けが行われる．IPv6 Header には 8 bits の Traffic Class フィールドがあり，QoSMoS ではそのうちの 6 bits を DSCP を書き込むフィールドとして利用する．

更に，QoSMoS における ER の marking component の Classifier は，始点 IP アドレス，終点 IP アドレス，そして Flow Label の 3 つをフィルタリングの識別子としてクラス分けを行う．IPv4 における Classifier は，識別子として IP アドレス，ポート番号やプロトコルなどを利用してクラス分けを行っていたので，トランスポートレイヤが付加する TCP Header や UDP Header の内容まで参照する必要があったが，QoSMoS では IPv6 Header の内容のみを参照すればよい．

3.3 モビリティ環境への対処

QoSMoS では，動的なネットワーク接続環境に適応できるようにするために，シグナリングメッセージによる Admission Control と Pre-Admission Control を導入する．また，Pre-Admission Control のために，ER の Router Advertisement (RA) を拡張する．この拡張 RA により，隣接する ER の IP アドレスと，それに接続する AP の MAC アドレスの対応付けがわかる．

Admission Control は通信開始時にユーザからの要請によって開始される．そして Domain 内のルータが Admission Control によって得られた結果をユーザに返答する．Admission Control では，ユーザの要求する QoS 保証が可能かどうか判定し，可能な場合は ER の marking component の Classifier に対して動的に設定を行う．Admission Control の内容は，始点と終点の IP アドレスと Flow Label 及び利用する DSCP と要求帯域である．その返答は OK または NG という形である．

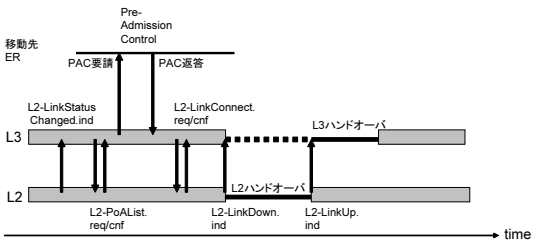

図 2: ネットワーク層主導ハンドオーバと Pre-Admission Control

また QoSMoS では，サブネットが異なる基地局間における移動時において移動透過性を実現させつつ，移動先予測に基づき事前に移動先経路において Pre-Admission Control を行う．それにより，QoS 保証された通信が途切れないことを可能にする．もし，移動先経路において移動前の QoS が保証できないときでも，ユーザに前もって知らせることにより対応ができるようにする．

QoSMoS では，モビリティサポートプロトコルとして Location Independent Networking for IPv6 (LIN6) [8, 9] を利用し，移動透過性を実現する．そして，Pre-Admission Control を行うための移動予測を可能とする機構として inter-Layer Information Exchange System (LIES) [10] を用いる．LIES を用いることにより，ネットワーク層とリンク層の連携が可能となり，ネットワーク層主導ハンドオーバを行うことができる．その過程の中で，リンクの切断予測と移動先予測に基づく Pre-Admission Control を実行する．

ネットワーク層主導ハンドオーバと Pre-Admission Control の流れを図 2 に示す．

このように，リンク層での通信品質が低下したときなどに (例えば電波強度がある閾値を下回ったときに) リンク層からネットワーク層に対して動的に通知され，ネットワーク層がハンドオーバの必要性を判断することから処理が開始される．そして接続可能な基地局のリストを L2-PoAList を用いてリンク層からネットワーク層に通知する．接続可能な基地局が複数存在する場合には，できるだけ同一ネットワーク内にあり通信品質の高いものを選ぶ．そこで，選択した移動先基地局が接続する ER に対し，Pre-Admission Control 要請を行う．Pre-Admission Control の内容は，移動前の通信開始時に行った Admission Control の内容と QoS 要求の点では同じであり，経路情報が移動先のものとなる．要請の返答は OK または NG という形である．Pre-Admission Control 完了後にハンドオーバが開始される．

このようなネットワーク層主導ハンドオーバをユーザ区間で制御するための機構として，本研究室で実装された Pre-Handover Daemon (prehod) [11] を利用する．prehod はカーネル空間の LIES とユーザ空間をブリッジするソケットを生成し，そのソケットを通じて L2 への request の送信，または L2 からの confirm や indication の受信を行う．この prehod が持つ機能を利用して，Pre-Admission Control 要請を行う．

3.4 無線区間の QoS 保証

従来の DiffServ ではドメイン内の通信帯域のみを制御していた。しかし，モビリティ環境を考慮する QoSMoS では，ER とそれに接続する MN 間の通信帯域も制御する必要がある．そこで QoSMoS では，ER が AP に現在の使用状況を SNMP で問い合わせることで，ER と MN 間の通信帯域も制御する．

3.5 突発的な輻輳への対処

QoSMoS は動的な通信環境における突発的な輻輳の発生時，輻輳に関係するトラフィックの両 End Node に対して送信量の抑制を要求する．しかし，CR と ER でその処理を行うとそれぞれが保持する情報量が多くなるので，QoSMoS はドメイン内に Traffic Manager(TM) を導入する．

TM は，Admission Control や Pre-Admission Control の時に，サービスをうける通信フロー情報の報告を ER から受ける．その報告により，TM はその通信フローの始点と終点の IP アドレス，始点と終点それぞれが接続する ER の IP アドレス，Flow Label を保持する．

輻輳発生時，輻輳を検知した ER または CR は，輻輳を発生させているパケットの始点 IP アドレスと Flow Label を送信する．TM はその情報から輻輳に関係するトラフィックの両 End Node を特定し，ER 経由で両 End Node に送信量の抑制を要求する．

4 QoSMoS の動作

4.1 通信開始時の動作

通信開始時の動作を図 3 に示す．図 3 の場合，MN が initiator で CN が responder であるとする．initiator とは通信開始要求を出す端末を指し，responder とはその要求を受ける端末を指す．また，通信経路が対称であることを前提として考える．

最初に，アプリケーション同士で通信開始要求が行われるとする (図 3 (1))．このときに，お互いが利用する Flow Label などの情報が交換されるとする．次に，initiator である MN は接続する ER1 に対して Admission Control 要請を行う (図 3 (2))．Admission Control 要請を受けた ER1 は CN への経路について Admission Control を行う (図 3 (3))．Admission Control Message が ER3 まで到達すると，ユーザの要求内容に対する許可，不許可を返答する (図 3 (4))．最後に，initiator 側の ER である ER1 は TM に対して通信情報を通知する (図 3 (5))．以上が完了すると，MN と CN は要求した QoS が保証された通信を行うことができる．

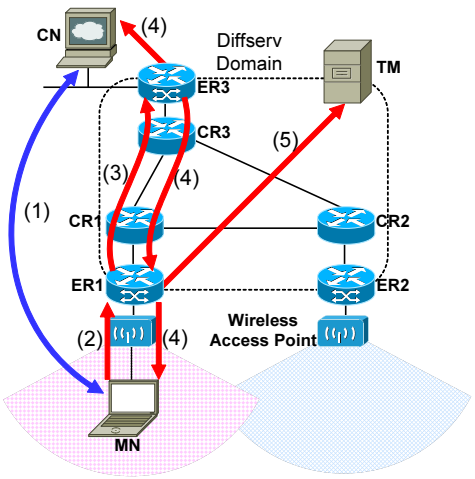

図 3: 通信開始時の動作概要

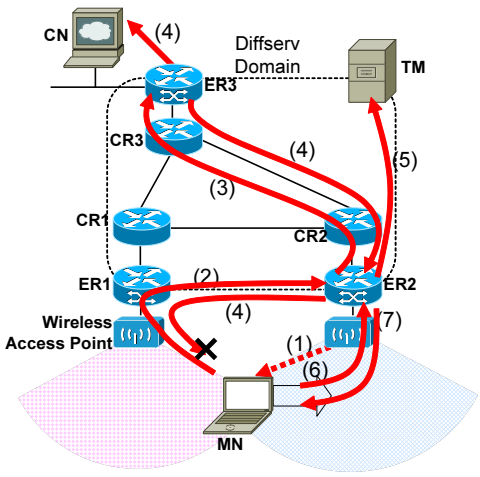

図 5: ハンドオーバ時の動作概要 2

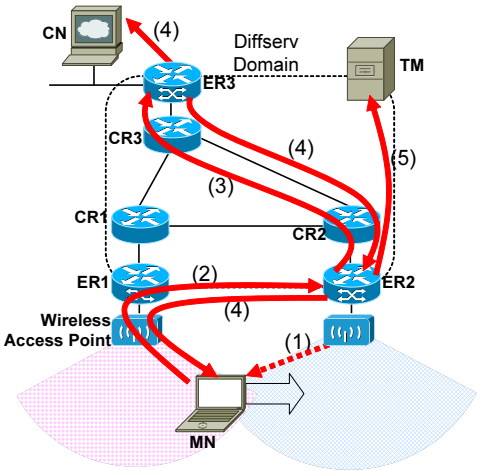

図 4: ハンドオーバ時の動作概要 1

4.2 ハンドオーバ時の動作

ハンドオーバ時の動作を図 4 に示す．MN は接続している基地局以外の基地局からビーコンを受信し (図 4 (1))，その電波強度が接続している基地局の電波強度よりも強いと判断した場合，その基地局が接続する ER2 に対してハンドオーバを開始する．ER1 より送信されている拡張した RA から ER2 のグローバルアドレスを知ることができるので，ER2 が接続するネットワークプレフィックスより移動先における LIN6 アドレスを生成し，ER2 に対して Pre-Admission Control 要請を行う (図 4 (2))．Pre-Admission Control 要請を受けた ER2 は CN への経路について Pre-Admission Control を行う (図 4 (3))．Pre-Admission Control Message が ER3 まで到達すると，ユーザの要求内容に対する許可，不許可を返答する (図 4 (4))．最後に，Pre-Admission Control 要請を受けた ER である ER2 は TM に対して通信情報を通知する (図 4 (5))．以上が完了すると，MN は L2 ハンドオーバ，L3 ハンドオーバを行い，MN と CN は移動先でも要求した QoS が保証された通信をスムーズに継続することができる．

ただし，図 5 に示すようにリンク層の判断により図 5 (4) のメッセージを受信する前に L2 ハンドオーバが行われてしまう場合が考えられる．その場合，MN は L2 ハンドオーバ，L3 ハンドオーバ完了後に ER2 に対して Pre-Admission Control を再要請する (図 5 (6))．ER2 は Pre-Admission Control 結果を保持しており，MN からの再要請があると保持した内容を再送する (図 5 (7))．これにより，ユーザは予期せぬハンドオーバが発生しても Pre-Admission Control が完了したことを知ることができる．

また，(図 4 (2)) の送信前に L2 ハンドオーバが行われてしまう場合は，L2 ハンドオーバ，L3 ハンドオーバ完了後に再度 Pre-Admission Control を要請する．

4.3 輻輳発生時の動作

輻輳発生時の動作を図 6 に示す．CR もしくは ER において，ある DSCP の利用帯域が限界に近づいてきたときは，それを TM に通知する．図 6 は，ある DSCP のフロー (a) とフロー (b) が発生し，CR2 と CR3 との間に輻輳が発生した例である．このとき，CR2 はある DSCP に対して割り当てたキューが閾値を超えたことを検知し，TM に対してそれを報告する (図 6 (1))．すると，TM はその DSCP を利用して CR2 を通る経路で通信を行っている ER に対して輻輳が発生していることを通知する (図 6 (2))．通知を受けた ER は接続する End Node に対して，要求された通信品質の劣化の可能性を通知する (図 6 (3))．これにより，End Node のアプリケーションは通信レートを変更する，通信を中断するなどの対応ができる．

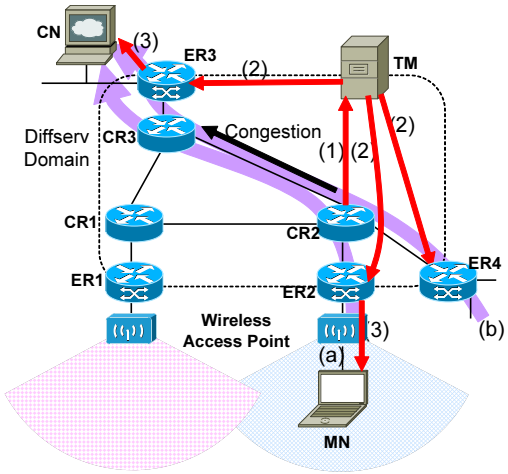

図 6: 輻輳発生時の動作概要

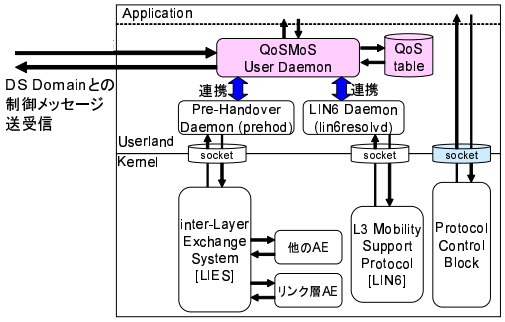

図 7: End Node のモジュール構成

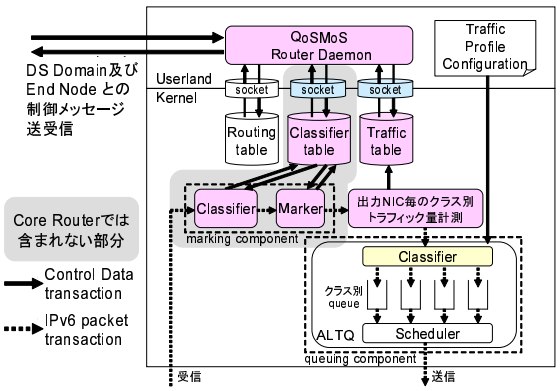

図 8: Edge Router と Core Router のモジュール構成

5 実装および評価

5.1 実装

本研究では，FreeBSD6.2-release のカーネル空間とユーザ空間に実装を行った．マシンは IPv6 通信のみを許可する．

図 1 で示したように，本研究が想定する QoS 保証サービスモデルは 1 つの Domain を構築する Edge Router，Core Router，および Traffic Manager と，そのサービスを利用する End Node からなるので，本実装は上記の 4 種類のノードに分けて行った．ただし，Edge Router と Core Router は共通する部分が多いので，Core Router は Edge Router の機能を一部省略する形で実装した．

また，IPv6 Header に対して EF クラスの DSCP を書き込む処理は，Socket API にその機能を追加する形で実装した．これにより，ユーザ空間から Open した Socket に Socket Option を指定し，送出パケットへ EF クラスの DSCP を書き込むことが可能となった．

5.1.1 End Node の実装

End Node のモジュール構成を図 7 に示す．End Node では LIN6 のマッピングを解決する lin6resolvd と呼ばれる Daemon を利用し，VID と LIN6 アドレスのマッピングを解決するために LIN6 のライブラリである liblin6 を利用した．また，L3 主導のハンドオーバをユーザ空間で制御するためのモジュールとして prehod を利用し，この Daemon とプロセス間通信を行えるようにすることで Pre-Admission Control を実現した．また，現状の IPv6 実装では UDP Connection における Flow Label をユーザ空間で取得できる Socket API が無かったため，Socket API に Flow Label を操作する機能を追加した．

5.1.2 Edge Router と Core Router の実装

Edge Router と Core Router のモジュール構成を図 8 に示す．図 8 に示したように，ER では IPv6 Header を利用した独自の Classifier と Marker を備えた marking component をカーネル空間に実装した．ただし，Meter および Dropper は実装していない．この Classifier と Marker を実装した関数は qosmos_input() と呼び，受信した IPv6 パケットに対し L3 入力処理を行う ip6_input() によって呼ばれる．ER の ip6_input() は qosmos_input() を呼び，その処理が終了するとルータとしての転送処理を行う ip6_forward() を呼ぶ．この Classifier や Marker に利用する情報を保持する Classifier table はカーネル空間に生成するようにし，カーネル空間とユーザ空間の QoSMoS Router Daemon との通信のインタフェースとして独自のソケットを実装した．これにより，ユーザ空間からの Classifier table に対するエントリの新規作成，編集，削除，閲覧等を実現した．

また，出力 NIC 毎のクラス別トラフィック量を計測する機構を実装した関数は qosmos_get_traffic() と呼び，IPv6 パケットに対し L3 出力処理を行う nd6_output() によって呼ばれる．nd6_output() では近隣ノード探索や出力 NIC を決定した後 qosmos_get_traffic() を呼び，その処理が終了すると L2 出力処理を行う if_output() を呼ぶ．計測した DSCP 値，出力 NIC 名，トラフィック量などを保持する Traffic table

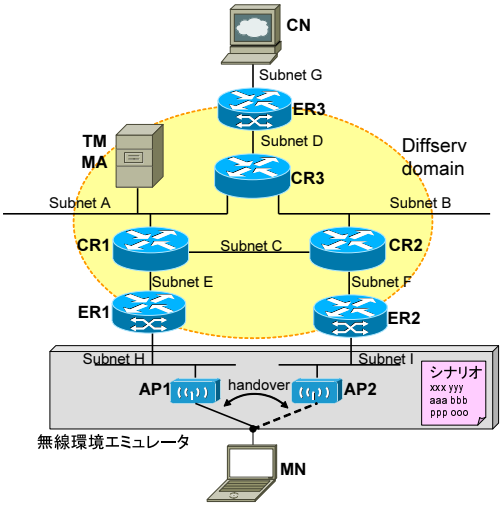

図 9: 実験ネットワークトポロジ

はカーネル空間に生成するようにし，カーネル空間とユーザ空間の QoSMoS Router Daemon との通信のインタフェースとして独自のソケットを実装した．これにより，ユーザ空間からの Traffic table に対するエントリの閲覧を実現した．

なお，queuing component には既存の ALTQ モジュールを利用した．

5.2 評価環境

QoSMoS の動作確認や性能評価は図 9 に示すネットワークトポロジにて行った．ネットワーク上には ER3 台，CR3 台，TM1 台から構成される 1 つの DS Domain を構築した．

無線 LAN 環境は，雑音や電波の干渉などの様々な要因でネットワークの状態が変化してしまうため，正確な動作確認や評価が困難である．よって，本研究では実際の無線 LAN を利用することなくリアルタイムにハンドオーバ実験を行うことのできる無線環境エミュレータを利用した．無線環境エミュレータは，電波強度，リンク切断時間，接続先などの様々なパラメータを自由に設定することができ，シナリオ通りのハンドオーバを実現できる．フローの発生には送信帯域が自由に設定できる Iperf [12] を利用した．

ALTQ の制御方法に CBQ を用いる．FreeBSD6.2-release に実装されている ALTQ_CBQ には各々のクラスに対して 1 (最低) から 7 (最高) までの優先度を割り当てることができ，より高い優先度を持つキューは，輻輳の際により低い優先度よりも優先的に出力処理される．また PHB については，EF，AF，DF の 3 つに限定して実験を行う．実験に用いる PHB の設定を表 1 に示す．各 NIC に割り当てる許容帯域の合計は 11Mbps とする．この設定を実験ネットワークの各 PC に設定した．実験ネットワークの各 PC は表 2 に示す性能を持つ．

表 1: 実験に用いた PHB の設定

PHB	QoS レベル	DSCP	CBQ	許容帯域
EF	高	0xfc	7	3 Mbps
AF	中	0xf0	4	3 Mbps
DF	要求無し (低)	0 (default)	1	5 Mbps

表 2: 実験に用いた各 PC の性能

PC	CPU	Memory (MByte)
MN	Intel Pentium M 1.2 GHz	512
CN	VIA C3 997.46 MHz	503
ER1	VIA C3 Ezra 797.97 MHz	503
ER2	VIA C3 Ezra-T 797.97 MHz	503
ER3	VIA C3 997.46 MHz	503
CR1	VIA C3 Ezra-T 797.97 MHz	503
CR2	VIA C3 Ezra 797.97 MHz	503
CR3	Intel Xeon 2.4 GHz	992
TM	VIA C3 Ezra-T 797.97 MHz	503

5.3 処理時間

5.3.1 パケット転送処理の処理時間

QoSMoS による Edge Router のパケット転送処理のオーバヘッドは，変更を加えた ip6_input() と nd6_output()，および ALTQ モジュール部分に発生する．一方，Core Router のパケット転送処理のオーバヘッドは，nd6_output() および ALTQ モジュール部分のみに発生する．

まず，MN が AP1 に接続している状況で CN から MN に対して 1 Mbps のフローを 60 秒間送信し，ER3 における ip6_input() と nd6_output() の処理時間を計測した．ここで，処理時間とは各パケットに対して Pentium Clock Counter の値から算出し，それらから求めた平均値のことである．ルータにおける ip6_input() の処理とは ip6_input() が呼びだされてから ip6_forward() が呼ばれるまでを指し，nd6_output() の処理とは if_output() が呼ばれるまでを指す．

既存の IPv6 の通信における測定結果，および QoSMoS を利用した通信における測定結果を図 10 に示す．なお，QoSMoS は Classifier Table のエントリ数を 0，1，10 の場合において測定した．1 エントリの場合は，フローが常にそのエントリにヒットし，Marker によって DSCP を書き込むように設定した．10 エントリの場合は，フローが 10 エントリのうちの 1 つにヒットし，Marker によって DSCP を書き込むように設定した．フローには EF クラスを割り当てた．

図 10 に示すように，QoSMoS における ip6_input() と nd6_output() の処理時間のオーバヘッドは既存の IPv6 の処理時間に比べ，合計で約 30%から 45%増加したことが分かった．QoSMoS における処理時間は，Classifier Table のエントリ数の増加に伴いエントリの検索時間が増加するた

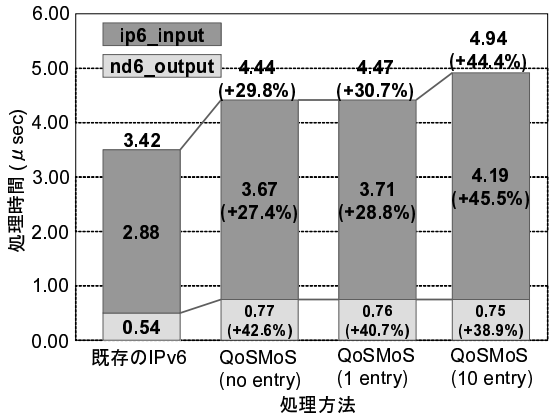

図 10: ip6_input() と nd6_output() における処理時間

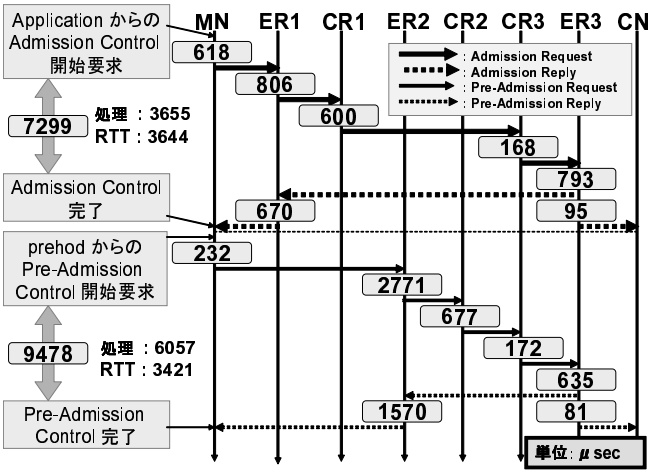

図 11: Admission Control と Pre-Admission Control の処理時間

め，処理時間も増加したと考えられる．

次に，ALTQ モジュールによる処理のオーバヘッドを測定するために CN から MN に対して 1 Mbps のフローを 60 秒間送信し，CR3 におけるパケット転送の処理時間を計測した．測定には tcpdump[13] を用い，ALTQ を利用しない場合と ALTQ で常に EF クラスを受信し処理を行う場合の処理時間の平均を算出し，その差を取ることで ALTQ モジュールの処理時間を求めた．その結果，ALTQ モジュールの処理時間は 26.26 μsec であった．

したがって，QoSMoS を用いたパケット転送処理時間のオーバヘッドは Edge Router では最大でも約 28.8 μsec，Core Router では約 26.5 μsec であったことが分かる．しかし，実ネットワーク環境において RTT は通常 $msec$ オーダーであり，QoSMoS における処理時間のオーバヘッドは μsec オーダーである．このように，RTT と比較するとオーダーが異なるため処理時間は無視できる．

以上により，ER および CR における QoSMoS を用いたパケットの転送処理には大きなオーバヘッドが発生しないことが分かった．

5.3.2 Admission Contorl と Pre-Admission Control の処理時間

Admission Contorl と Pre-Admission Control の処理時間を測定した．処理時間の計測のために，図 9 で示した実験ネットワークにて実験を行った．

最初 MN は AP1 に接続している状態で Admission Control を行う．続いて MN は AP1 から AP2 へ移動を開始し，電波強度から事前にハンドオーバを検知し Pre-Admission Control を行う．このようなシナリオで評価を採った．

処理時間の計測には c 言語のライブラリ関数である gettimeofday() を利用した．上記のシナリオで 10 回実験を行い，各ノードの各処理毎に算出した値の平均値を処理時間として求めた．

Admission Control を開始するインターフェースからのシグナルを MN が受信した時刻を，Admission Control の開始時刻とした．また，MN が ER からメッセージを受信し，QoSMoS User Daemon が Admission Control の完了を確認した時刻を Admission Control の終了時刻とした．各ノードにおける処理時間とは，Daemon がメッセージを受信してから処理を行い，次のノードへメッセージを送信するまでの時間のことを指す．

Pre-Admission Control に関しては，prehod からハンドオーバの発生予告のシグナルを受信した時刻を Pre-Admission Control の開始時刻とした．また，MN が ER からメッセージを受信し，それに対する ACK ををを送信した後 QoSMoS User Daemon が Pre-Admission Control の完了を確認した時刻を Pre-Admission Control の終了時刻とした．各ノードにおける処理時間の定義は上記の Admission Control のものと同様である．

実験によって得られた Admission Control と Pre-Admission Control の処理時間を図 11 に示す．図 11 中の Round Trip Time (RTT) は，各ノードで ping6 コマンドを実行した結果得られた値である．

図 11 で示したように，本実験では Admission Control に要する時間は 7.30 $msec$ であった．Admission Control の処理時間は通信経路におけるホップ数の増加と RTT により大きな影響を受けるが，Admission Control の処理は QoS 保証された通信の開始時に行われるのみなので，数十 $msec$ から数百 $msec$ 単位であるこの処理時間は許容できる範囲である．

また，Pre-Admission Control に要する時間は 9.48 $msec$ であった．Pre-Admission Control の処理時間は Admission Control と同様に通信経路におけるホップ数の増加と RTT により大きな影響受けるが，この処理はハンドオーバ前に行う処理であるため，処理のオーバヘッドは無視できる．

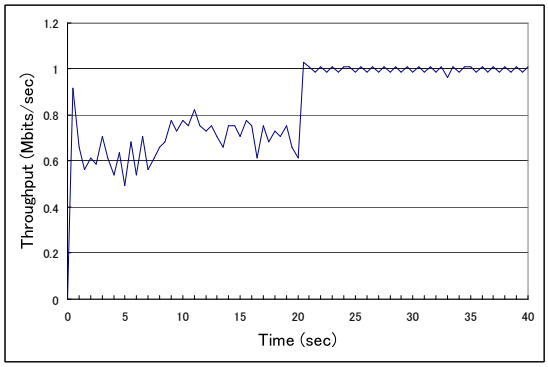

図 12: Admission Control によるスループット変化

5.4 帯域保証

5.4.1 固定環境時のスループット

固定環境時のスループットの評価をとった．最初フロー A を ER3 から ER1 を通るように DF クラスで 4 Mbps で流し，フロー B を CN から ER1 経由で MN まで DF クラスで 1 Mbps で流した．20 秒経過した後，Admission Control を行いフロー B を DF クラスから EF クラスに変えた．

実験から得られた Admission Control によるフロー B のスループット変化を図 12 に示す．

図 12 に示すように，通信開始時は 5 Mbps が割り当てられた DF クラスに対し，フロー A とフロー B で合計 5 Mbps のフローを発生させたため輻輳が発生した．フロー B のスループットはパケットロスや遅延によって 1 Mbps に満たず約 0.7 Mbps となりジッタも大きい．しかし，20 秒後に Admission Control が行われると，フロー B には EF クラスの DSCP が ER3 によって書き込まれるようになる．EF クラスには 3 Mbps を割り当てられているので，フロー B は 1 Mbps を確実に利用することができ，スループットが向上した．

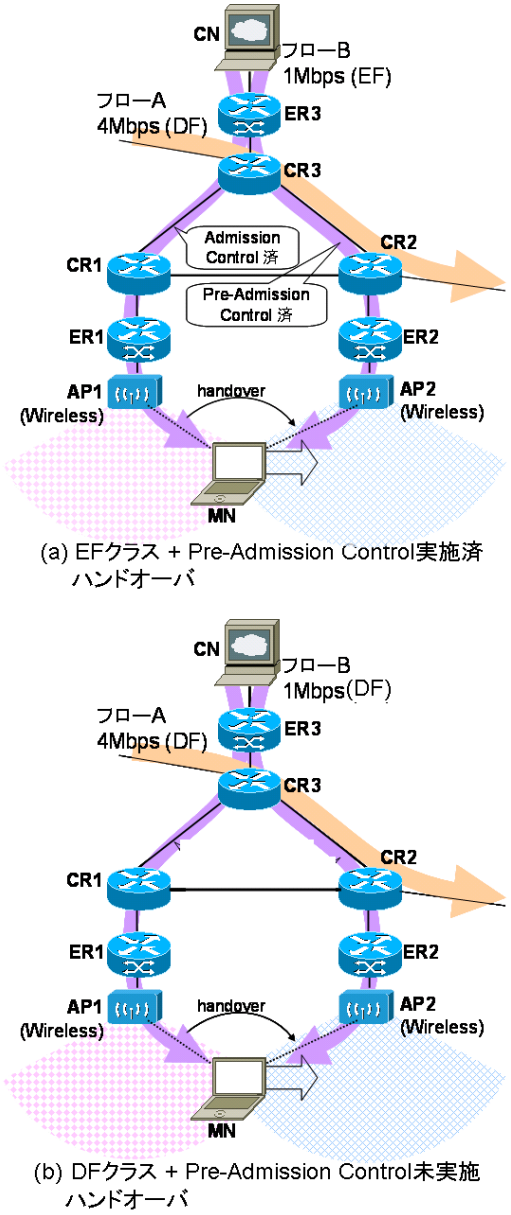

図 13: 移動環境における実験で想定するネットワーク

5.4.2 ハンドオーバ時のスループット

ハンドオーバ時の QoSMoS の動作確認と，Pre-Admission Control による QoS 保証の有効性に確認するために，図 9 で示した実験ネットワークにて次のようなシナリオで実験を行った．

MN は AP1 に接続していて，CR3 から CR2 に向かう経路に DF クラスのフロー A を 4 Mbps で発生させ，CN から CR1，ER1 を通り MN に向かう経路に 1 Mbps のフロー B を発生させる．MN はフロー B に対して優先度を最高にしてもらうよう ER1 に対して Admission Control 要請を行う．Admission Control が完了し，フロー B は EF クラスになる．MN が AP1 から AP2 の方向に移動を開始する．AP1 よりも AP2 から受信する電波強度の方が大きくなったとき，移動後の経路でも EF クラスの通信が継続できるように ER2 に対して Pre-Admission Control 要請を行う．その後，MN が AP1 から AP2 へハンドオーバを行う．

このシナリオにおいて，MN が移動し AP1 より AP2 から受信する電波強度の方が大きくなるタイミングは 15 秒後とした．このシナリオで想定するネットワークを図 13 (a) に示す．

Pre-Admission Control が行われなかった場合と比較するために，上記のシナリオでフロー B に対して Admission Control および Pre-Admission Control を行わず，DF クラスのままハンドオーバを行う実験も行った (図 13 (b))．実験から得られた Pre-Admission Control によるフロー B のスループット変化を図 14 に示す．

図 14 に示すように，図 13 (a) の場合と図 13 (b) の場合

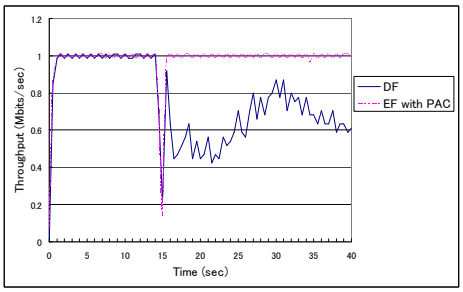

図 14: Pre-Admission Control によるスループット変化

は共に計測開始から約 15 秒後にハンドオーバが発生し，スループットが 1 Mbps から 0.2 Mbps 付近まで減少した．その後，図 13 (a) の場合はハンドオーバ後も EF クラスを使えるよう Pre-Admission Control によって設定されており，約 1.5 秒後には要求帯域が利用できるようになった．図 14 においてスループットが減少した時間が約 1.5 秒間生じた原因の 1 つとして，L3 ハンドオーバにおける Duplicate Address Detection (DAD) が挙げられる．DAD では標準で 1 秒間の IP アドレス重複検知を行い，重複アドレスがないと判断された場合インタフェースに IP アドレスを付与する．この検知中は通信が中断するので，スループットが大きく減少したと考えられる．事前に Pre-Admission Control と同時に DAD 処理も行うことができれば，この大きなスループットの減少は軽減できると考えられる．

一方，図 13 (b) の場合ハンドオーバ前は DF クラスに空き帯域があったので問題なかったが，ハンドオーバ後の経路では 5 Mbps 割り当てられた DF クラスに対しフロー A とフロー B で合計 5 Mbps のフローを発生させたため輻輳が発生し，フロー B のスループットはパケットロスや遅延によって 1 Mbps に満たず変動し続けた．

5.5 考察

5.5.1 同時に複数の Admission Control や Pre-Admission Control への対応

本研究では，Admission Control や Pre-Admission Control を要求された時，Edge Router や Core Router においてその時点の使用帯域から利用可能帯域を計算し，許可するかを判断している．よって，同時に複数の Admission Control や Pre-Admission Control を受けた時，個々の要求に対して許可をしてしまい，総和の要求帯域を考慮しない．つまり，2 Mbps の利用可能帯域に対して同時に 1 Mbps の Admission Control を 10 個受けると，それぞれに対して許可してしまい，結果的に輻輳を招いてしまう．このような場合も考慮して，帯域要求に対する判断の仕組みが必要である．

5.5.2 通信品質保証のドメイン間への拡張

本研究では同一ドメイン内における通信やハンドオーバのみを扱っており，異なるドメイン間における通信やハンドオーバは考慮していない．QoSMoS のドメイン間への拡張性はあると考えられるが，実際に必要な状況の分析や仕様検討が不十分である．あらゆる状況における Admission Control の評価をふまえた上で，ドメイン間においてもモビリティがサポートされた通信品質保証を提供できる QoSMoS の設計と実装が必要である．

5.5.3 輻輳発生時の対応

本研究では，輻輳に対応するためのエージェントとして Traffic Manager を導入したが，輻輳発生時の対応はその輻輳の要因となるフローを発生させているノードに対してメッセージを送信する処理のみである．実際，EF クラスに対しては輻輳は発生させてはならないのだが，本研究の実装では各クラスに割り当てられた帯域は限界近くまで利用することができ，どのクラスにおいても遅延やパケットロスが発生する可能性がある．よって，どの程度の利用率以上を輻輳と判断するのか，Admission Control の時点で NG を出す判断基準，輻輳と判断されたときの DS Domain や End Node における対応方法についてさらなる詳細な設計と実装が必要である．

5.5.4 End Node における処理の軽減

本研究の実装では，既存モジュールとの連携によりモビリティと Pre-Admission Control を実現したが，複数プロセスの起動やプロセス間の通信が必要になるため End Node への処理のオーバヘッドが大きくなっている．また複数のプロセス起動に対する設定が必要であり，ユーザの操作が複雑であるため，モジュール間の連携でなく統一化して処理することで，End Node における処理のオーバヘッドを小さくすることが可能であると考えられる．よって，今後複数のモジュールを統一した実装を行う必要がある．

6 結論

本研究では，モビリティ環境におけるスケーラブルな通信品質保証機構である QoSMoS を提案し設計した．そして，QoSMoS を FreeBSD6.2-Release 上に実装し，性能評価と考察を行った．

QoSMoS では IPv6 ネットワークを想定し，Diffserv の概念を導入しつつ IPv6 Header のみによるクラス分けを行った．

シグナリングメッセージによる動的な Admission Control を行うことにより，動的なネットワーク接続環境に対応した通信品質保証を実現した．また，モバイルノードのハンド

オーバ後の通信経路に対してハンドオーバ前に事前の Pre-Admission Control を行うことにより，モバイルノードがサブネットの異なる基地局間を移動する場合においても QoS 保証された通信をスムーズに継続させることを実現した．

そして，QoSMoS は既存の IPv6 と比較して無視できる程度のオーバヘッドでモビリティ環境におけるスケーラブルな通信品質保証が実現できることを示せた．

参考文献

[1] S. Lai, Y. Ma, and H. Deng. HO-RSVP: a Protocol Providing QoS Support for Seamless Handover between Wireless Networks. In *Proceedings of the 2nd ACM International Workshop on Quality of Service and Security for Wireless and Mobile Networks*, pp. 103–110, 2006.

[2] J. Antonio Garcia-Macias, Franck Rousseau, Gilles Berger-Sabbatel, Leyla Toumi, and Andrzej Duda. Quality of Service and Mobility for the Wireless Internet. *Wireless Networks*, Vol. 9, pp. 341–352, 2003.

[3] Yan Cheng and J. W. Atwood. A Hierarchical Agent Assisted MIPv6 Handover Scheme with QoS Support. In *The 3rd International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine'06)*, August 2006.

[4] Özgür B. Akan and Buyurman Baykal. Handoff Performance Improvement with Latency Reduction in Next Generation Wireless Networks. *Wireless Networks*, Vol. 11, pp. 319–332, 2005.

[5] Indu Mahadevan and Krishna M. Sivalingam. Architecture and Experimental Framework for Supporting QoS in Wireless Networks Using Differentiated Services. *Mobile Networks and Applications*, Vol. 6, pp. 385–395, 2001.

[6] Ch. Bouras and K. Stamous. An Adaptive Admission Control Algorithm for Bandwidth Brokers. In *Proceedings of the 3rd IEEE International Symposium on Network Computing and Applications (NCA'04)*, 2004.

[7] Giuseppe Bianchi, Nicola Blefari-Melazzi, Pauline M. L. Chan, Matthias Holzbock, Y. Fun Hu, Axel Jahn, and Ray E. Sheriff. Design and Validation of QoS Aware Mobile Internet Access Procedures for Heterogeneous Networks. *Mobile Networks and Applications*, Vol. 8, pp. 11–25, 2003.

[8] Mitsunobu Kunishi, Masahiro Ishiyama, Keisuke Uehara, Hiroshi Esaki, and Fumio Teraoka. LIN6: A New Approach to Mobility Support in IPv6. In *Proceedings of the Third Inernational Sympsium on Wireless Personal Multimedia Communications*, pp. 1079–1084, November 2000.

[9] 國司光宣, 石山政浩, 植原啓介, 寺岡文男. 移動体通信プロトコル LIN6 の性能評価. 情報処理学会論文誌, Vol. 43, No. 2, pp. 398–407, February 2002. マルチメディアコミュニケーションシステム特集.

[10] 渋井理恵, 神谷弘樹, 寺岡文男. レイヤ間情報伝達機構 LIES. 日本ソフトウェア科学会　インターネット テクノロジーワークショップ (WIT2004) 論文集, 2004.

[11] F. Teraoka, K. Gogo, K.Mitsuya, R.Shibui, and K.Mitani. Unified L2 Abstractions for L3-Driven Fast Handover. Internet Draft, *IETF*, February 2008.

[12] NLANR/DAST : Iperf - The TCP/UDP Bandwidth Measurement Tool. http://dast.nlanr.net/Projects/Iperf/. Accessed 2007.

[13] tcpdump - dump traffic on a network. http://www.tcpdump.org/. Accessed 2007.