

# 無線 TCP における MAC 情報を用いた明示的無線リンクロス通知方式

張 兵<sup>†</sup> M. N. シラジ<sup>††</sup> 田 中 信 介<sup>†</sup>

携帯端末の普及や無線通信技術の発達にともない、無線環境に対応する TCP に関する研究が注目されている。従来の TCP は無線の高い誤り率によるバケットロスを輻輳によるロスと区別できないため、輻輳状態にないにもかかわらず不必要にデータの送信速度を抑制するという問題があった。この問題を解決するため、無線リンクロスを明示的に通知する EWLN (Explicit Wireless Loss Notification) が提案されている。しかし、この手法では、有線から無線へならびに無線から有線への両方の通信に対して同様な仕組みで実現できないうえに、物理層ならびにデータリンク層にエラー訂正技術や再送技術を適用した場合において基地局における実装方法が明確ではなかった。そこで本論文では、MAC 層の情報を利用して、リンク層の再送を考慮した homogeneous な EWLN 方式を提案する。特に、従来の TCP と互換性を持つ TCP-Reno と TCP-SACK ベースの EWLN の実現方法について具体的な検討を行う。提案方式は、移動端末、基地局のいずれにおいても用いることができる。有線から無線へならびに無線から有線への両方の TCP 通信をサポートすることができる。

## Improving Wireless TCP Performance with Explicit Wireless Loss Notification Using MAC-layer Information

BING ZHANG,<sup>†</sup> M. N. SHIRAZI<sup>††</sup> and SHINSUKE TANAKA<sup>†</sup>

This paper discusses the problem of TCP Performance degradation over wireless links. This is an important issue to be addressed due to the rapid spread of wireless Internet access through mobile terminals. The Explicit Wireless Loss Notification (EWLN) scheme has been proposed as a viable solution to this problem. Previous works on deploying EWLN often lacked explicit specification of the interplay between EWLN and link layer protocols, including error recovery mechanisms, and sacrificed TCP's end-to-end property because of the need to monitor and process all TCP traffics at a base station. Moreover, they did not address compatibility issue. In this paper, we present a homogeneous EWLN scheme that deploys the error detection mechanism of the MAC protocol and takes into account the interplay with the error recovery mechanism at the link layer. The scheme does not require different implementations for mobile terminals and base stations and hence can support efficient TCP-based data transmission over wired-to-wireless and wireless-to-wired connections. We also present backward-compatible and optimal EWLN-enabled implementation for the standard TCP-Reno and TCP-SACK protocols, and show significant improvements of TCP performance by simulation studies over ns-2 network simulator.

### 1. はじめに

グローバルなインターネット・インフラにおける無線技術の重要性が高まるにつれて、様々な伝送方式で使用される無線、有線インターネットを統合する効果的な通信アーキテクチャの必要性が日増しに高まってきた。現在ではインターネット全トラフィックの90%以上を占めるWWW、ファイル転送、電子メー

ルなどのサービスは、TCP プロトコルによって支えられている。TCP は、有線リンクや固定ホストで構成される従来型のネットワークにおいて、輻輳によるバケット損失やエラー訂正などといった、主たる問題を克服することを念頭にデザインされているため、無線ネットワークの特徴はこれまで考慮されてこなかった<sup>1)</sup>。

TCP は、バケット損失の対策として失われたバケットを再送信すると同時に、輻輳ウィンドウサイズを減少させる輻輳制御動作を開始し、再送信タイムをバックオフすることによってバケット損失に対応している。そのバケット損失や特異な遅延の原因が主にネッ

<sup>†</sup> ATR 適応コミュニケーション研究所

ATR Adaptive Communications Research Laboratories

<sup>††</sup> 通信総合研究所

Communications Research Laboratory

トワーク上の輻輳によるものと考えられるため、TCP は無線リンクの特徴であるランダムエラーやバーストエラーには対応していない。そのため、無線ネットワークにおいては TCP の通信効率が著しく劣化することが知られている。

近年、無線リンクを持つネットワーク上で TCP 転送を行うための改善方式がいくつか提案されている。これらの方式を方法論的にまとめると、2つのカテゴリに分類することができる。1つは、ネットワークの輻輳と無関係の packets 損失を TCP 送信側から隠蔽し、従来の TCP をそのまま使用する方式である。こうしたアプローチの例としては、ネットワークを有線部と無線部に分離する Indirect TCP (I-TCP<sup>2)</sup>、ならびに Snoop プロトコル<sup>3)</sup> のような TCP をアウェアするリンク層プロトコル<sup>3)</sup> などがある。リンク層プロトコルがネットワークプロトコルの階層構造や TCP のエンドツーエンド通信に適合し、迅速かつ効率的なエラー回復を行うことができるため、有効な手法だと考えられる。しかし、これまで提案されてきた I-TCP や Snoop のようなリンク層プロトコルは基地局に大きな負荷をかけるとともに、TCP がエンドツーエンドの通信サービスをサポートする本来の役割を果たさなくなるというデメリットがある。2つ目の分類では、種類の異なる packets 損失を識別することによって、TCP の性能改善を行う方式である。この例としては、TCP 送信側でエラーのタイプを認識するもの<sup>4)</sup> や、EWLN (Explicit Wireless Loss Notification)<sup>5)</sup> などがある。EWLN 方式はネットワークプロトコルの階層構造に完全に適合しているだけでなく、TCP が packets 損失の原因を完全に識別することを可能にしているため、特に有望だと考えられる。

EWLN 方式は、固定の有線端末から無線端末への通信、または移動端末から有線端末への通信という、2つのシナリオに対して提案が行われてきた。しかし、EWLN の実装にあたって、1) 有線から無線へと無線から有線への両方の通信に対して同様な仕組みで実現できない、2) 基地局での計算オーバーヘッド、3) データリンク層のエラー訂正技術や再送技術を考慮していない、などの問題がある。

本論文では、ネットワークプロトコルの階層構造に適合する一方で、リンク層の再送を考慮した homogeneous な EWLN 方式を提案する<sup>6)</sup>。特に、従来の TCP との互換性 (compatibility) を持ちつつ、TCP-Reno と TCP-SACK ベースの EWLN 方式の実現について具体的な検討を行う。提案方式は、移動端末、基地局のいずれにおいても適用することが可能であり、

有線から無線へならびに無線から有線への両方の TCP 通信を同様な仕組みで実現することが可能である。

## 2. 関連研究

無線リンクは高い誤り率と干渉、フェージング、シャドウイングおよび端末の移動に起因するバーストエラーによって特徴付けられている。TCP は無線の高い誤り率による packets ロスを輻輳によるロスと判断し、輻輳制御メカニズムが起動されるため、転送特性が大幅に劣化する。この問題を回避するため、最後のホップが無線リンクであるケースを主に最近様々な TCP の性能改善方式が提案されてきた<sup>7)</sup>。

### 2.1 分離 TCP

分離 TCP は基地局においてネットワークを有線部と無線部に分離して、無線部の高いビット誤りを有線部から隠蔽する方式である。その中で代表的な手法は I-TCP である。I-TCP は分割された有線部において従来の TCP、無線部において無線環境に合わせて改良を施された W-TCP を使用する。この手法は固定端末において従来の TCP をそのまま使用することによって TCP の性能向上を試みている。しかし、欠点は、1) 基地局で TCP の重複処理によってオーバーヘッドが発生している、2) end-to-end semantics が保持されない、3) 基地局で各々の TCP 接続に対して大量の状態保存によってハンドオフの手順が煩雑である、などがあげられる。

### 2.2 データリンク層プロトコル

データリンク層プロトコルとしては選択的に再送する技術 ARQ (Automatic Repeat reQuest) や誤り訂正技術 FEC (Forward Error Correction) などがよく知られている。リンク層の再送によるエラー回復はフレームが順序良くかつ信頼性のある転送を行うのに使われている。FEC 方式は無線リンク上の損失率を事前にあるいは実時間的に推定することによってエラー訂正を行っている。

単純なリンク層方式は端末における TCP の変更を要求しないうえに、ネットワークの階層構造に適合し、なおかつ前述した分離 TCP のような欠点を持っていない。しかし、リンク層とトランスポート層においてタイマの互換性がないため、再送における競合が生じて、通信効率が劣化する恐れがある<sup>8)</sup>。

TCP アウェアなリンク層プロトコルの代表的な手法はリンク層と TCP 層の再送の連結によって TCP の性能劣化を防ぐ Snoop プロトコル<sup>3)</sup> である。本方式は固定端末から移動端末への通信に対して、基地局が Snoop Agent を配することにより、コネクション

を監視する。Snoop Agent は移動端末によって受信確認がとれていないパケットをいったん基地局にキャッシュする。DupACK あるいはタイムアウトによるパケットロスが発生する場合、Snoop Agent はロスしたパケットがキャッシュにあれば、固定送信端末に代わってそのパケットの再送処理を行う。このような局所的な再送処理によって、Snoop プロトコルは高速なエラー回復を実現できるとともに、TCP 送信側から無線リンクロスを隠蔽することによって不必要な輻輳制御を回避することができる。一方、移動端末から固定端末への通信に対しては、基地局の Snoop Agent に加えて、Snoop プロトコルは移動端末においても TCP の変更を行った。Snoop Agent は送信ウィンドウにおけるパケットロスを記録し、それらに対して負の ACK を生成し、移動端末に送信する。SACK オプション<sup>10)</sup>を使用するという前提のもとで、移動端末は負の ACK を用いて選択的な再送を行う。

Snoop プロトコルは基地局においてオーバーヘッドが生じるとともに、固定端末から移動端末へと移動端末から固定端末への両方の通信に対して homogeneous な処理ができない。さらに、他のデータリンク層プロトコルと同様に無線リンクロスを TCP 送信側から完全に隠蔽することができないという欠点がある。

### 2.3 エンドツーエンドプロトコル

TCP は無線リンクにおける性能劣化の主な原因が、ビット誤りによるパケットロスを輻輳によるものと判断してしまうことにある。この問題を解決するために、最も有効な対策方法は輻輳によるロス、あるいは無線リンク上のビット誤りによるロスを認識し、ロスの種類によって適切な処理を施すメカニズムを組み込むことである。パケット損失のタイプを識別するにあたって、以下の 2 つの手法がある：1) TCP 送信側においてエラーのタイプを識別する；2) 明示的に無線リンクロス (EWLN) を通知する、手法である。1) の手法は実測値と一連の仮定に基づくため、その正確さが欠けている。一方、2) の EWLN 方式は無線リンクによるパケットロスを基地局あるいは受信端末によって明示的に TCP 送信側に通知するため、かなり精度の良い手法となることが期待できる。

Balakrishnan ら<sup>9)</sup> は受信端末が無線リンクロスを判断する十分な知識を持つことを前提に、固定端末から移動端末への通信を行う場合において、EWLN の評価を行った。また、別の文献<sup>5)</sup>においては、彼らは移動端末から固定端末への通信についても考察を行った。その場合、彼らの EWLN 方式は Snoop 方式と同様に、基地局にエージェントを設けることによって無

線リンク上で到着するすべてのセグメントを監視する。ACKs が受信端末から届くと、基地局のエージェントは保持している無線リンク上で失ったセグメントに対応した順番空間のホールリストと照合して、EWLN ビットをセットする。そして、現時点の ACK ナンバより小さい番号のセグメントホールをすべて削除する。Snoop 方式と違って、基地局による再送を行わないため、基地局におけるパケットのキャッシュを行わない。

上記の EWLN の実現方法にいくつかの短所がある。特に、1) 固定端末から移動端末へと移動端末から固定端末へのそれぞれの通信に対して、受信端末と基地局において異なったリンク層の実装方式が必要なため、homogeneous なアプローチではない、2) エージェント処理をするため、基地局への計算負荷が大きい、3) 無線リンクの高い誤り率に対応するために不可欠なリンク層におけるエラー回復機構についてまったく考慮していない、などが問題となる。

### 3. Homogeneous なエンドツーエンドベースの EWLN 方式

本論文では、我々はエンドツーエンドで下記の特徴を持つハイブリッドな手法を提案する。1) 有効なリンク層におけるエラー訂正メカニズムを考慮し、なおかつ受信端末と基地局において homogeneous な EWLN 方式が実現できる；2) 提案手法は従来の TCP と互換性がある（この問題は今まで無視されてきた）。

#### 3.1 MAC 情報を利用した EWLN の実現方法

EWLN 方式は輻輳と無関係なパケットロスを TCP 送信側に通知する方式である。TCP 受信側はパケットロスの原因が輻輳によるものでないと分かれば、対応する ACK に EWLN ビットをセットし、それを TCP 送信側に送る。しかし、どういうメカニズムによってパケットロスの原因を識別するかについていままでの研究では明らかにできなかった。本章では、MAC プロトコルによって無線リンクロスを検出して、その発生を TCP 受信側に転送するためのメカニズムについて議論する。

IEEE802.11 などの無線 LAN MAC プロトコルでは、Cyclic Redundancy Check (CRC) によってデータパケットが正しく受け取られたかどうかをチェックする。もしデータパケットが正しく転送されていなければ、リンク層における局所的な再転送は制限回数まで送り続ける。最大再送回数まで送ってなお CRC によりビット誤りが検出される場合、IEEE802.11 ではそれらのパケットを破棄してしまう。ここで、ビット誤りが含まれるパケットを捨てる代わりに、そのパ

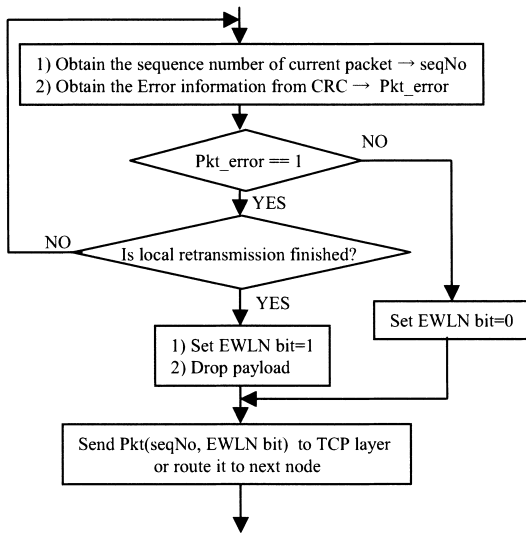


図1 MACプロトコルにおいて EWLN ビットを設定するフローチャート

Fig.1 Flow chart of setting EWLN bit within MAC protocol.

ケットを EWLN メッセージとして、移動端末においてはトランスポート層に送るが、基地局においては次のノードに転送する。ビット誤りがたいていペイロードに発生するため、我々はビット誤りにより破壊されたパケットの IP および TCP のヘッダ情報が利用できると仮定する。図 1 は MAC プロトコルにおいて EWLN ビットを設定するフローチャートを示す。

EWLN ビットを設定する流れとしては、まずビット誤りがあるかどうかをチェックする。もし誤りがあれば、先に到着しているパケットの番号といま到着しているものとの比較によって局所再送が終わったかどうか確認される。ここでは、我々はリンク層プロトコルにおいて IEEE802.11 のように局所再送が実行されると仮定する。他のエラー回復スキームとそれに対応する検証メカニズムについては、ここでは特に言及しない。上記の検証を経てから、EWLN ビットは CRC の誤り情報に基づいて設定される。もしパケットに誤りが含まれれば、パケットのペイロードが落とされて、TCP ヘッダにある EWLN ビットを 1 に設定する。ここで、TCP ヘッダにおける EWLN ビットの場所については、文献 5) に特に記述がなかったが、TCP ヘッダにおいて空いているビット、たとえば reserved field (6 bits) の 1 ビットを使うことが考えられる。そして、ペイロードのない、かつ EWLN 情報を持つパケットが移動端末においてはトランスポート層に送られるが、基地局においては次のノードに転送される。このような EWLN 実現方法は移動端末と

基地局の両方で実行できるとともに、TCP が輻輳によるロスと無線リンクロスの両方を適切に扱うことが可能になる。以下の章において EWLN 機能を持つコンパチブルな TCP-Reno と TCP-SACK の実現方法、さらにコンパチブルではないが、最も効果的な EWLN 機能を持つ TCP-Reno の実現方法について述べる。

### 3.2 EWLN 機能を持つコンパチブルな TCP の実現方法

TCP にはいくつかのバージョンがあるが、ここでは一番よく使われている TCP-Reno<sup>8)</sup> と選択的な ACK に基づく効率的なエラー回復を行う TCP-SACK<sup>10)</sup> をベースに EWLN 方式の実現について具体的な検討を行う。以降、我々は EWLN 機能を持つ TCP-Reno と TCP-SACK 方式をそれぞれ EWLN-Reno と EWLN-SACK と呼ぶ。累積 ACK を使用するため、TCP-Reno は 1 RTT (Round-Trip Time) あたり 1 個のパケット損失しか検出できない。そのため、1 つの送信ウィンドウに複数のパケット損失が発生する場合、TCP-Reno は再送タイムアウトにならないと、2 個目以降のパケット損失を発見できない。しかし、無線リンクにおける高い誤り率とバーストエラーのため、1 つの送信ウィンドウに複数のパケット損失がしばしば発生する。一方、TCP-SACK は選択的な ACK 機能を TCP-Reno に付け加えることによって、TCP 送信側が 1 つの送信ウィンドウに複数のパケット損失が発生しても対応処理ができるようになった。そのため、EWLN 機能を持つ TCP-SACK の実現も重要となる。

EWLN 機能を生かすため、従来の TCP に以下の特徴を持たせる必要がある。

- TCP 受信側はパケット損失の原因を TCP 送信側に知らせることができる。
- TCP 送信側は、輻輳と無関係のパケット損失を検出する場合、輻輳制御メカニズムを起動しない。
- TCP 送信側は輻輳と無関係のパケット損失に対して、3 つの Dupacks 或いはタイムアウトを待たずに、再送を行うことが望ましい。
- TCP 送信側と受信側は今後普及していくため、標準 TCP とコンパチブルであることが望ましい。

以下のセクションでは、上記の特徴を満たすような EWLN 実現方法を TCP-Reno と TCP-Sack の両方について述べる。

#### 3.2.1 TCP 受信側

TCP 受信側が EWLN 情報を持つパケットを受信すると、無線リンクロスによるパケット損失に対応するため、EWLN ビットをセットした ACK パケットを生

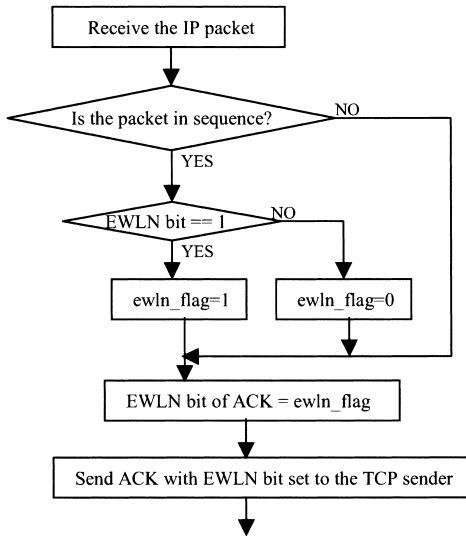


図 2 TCP 受信側における ACK の EWLN ビットを設定するフローチャート

Fig.2 Flow chart of setting EWLN bit within ACK at TCP receiver.

成する。EWLN ビットを設定する以外に、累積 ACK と SACK オプションが従来どおりの動きをする。したがって、TCP 送信側が EWLN 機能を持たなくても、TCP コネクション上でのデータ通信を従来どおり行うことができる。ここで注意すべき点は TCP におけるチェックサム機能である。EWLN ビット付加に際し、中継ルータにおいてパケットのペイロードが落とされるので、EWLN 機能が付いていない受信端末であれば、チェックサムエラーにより EWLN パケットが落とされることになる。EWLN パケットは本来ならばビットエラーにより破棄すべきものなので、EWLN 機能を持たない受信端末においては特に問題が生じない。一方、EWLN 機能を持つ受信端末においては、TCP のチェックサム処理より先に EWLN ビットをチェックする必要がある。具体的には、EWLN ビットが 1 であれば、TCP のチェックサムを通らずに、処理を行うが、EWLN ビットが 0 であれば、通常の処理を施す。図 2 は TCP 受信側で ACK の EWLN ビットを設定するフローチャートを示す。ACK の EWLN ビットがいったん設定されると、ACK 落ちに対して耐性を持たせるために、ロスしたパケットの再送が確認されるまでずっと立て続ける。それを実現するために、*ewln\_flag* というフラグ変数を導入する。TCP 受信側はパケットが到着すると、まずパケットが順番どおり到着しているかどうかをチェックする。もしパケットが順番どおりに来ていれば、*ewln\_flag* が EWLN の値に設定されるが、もしパケットが順番どおり来てい

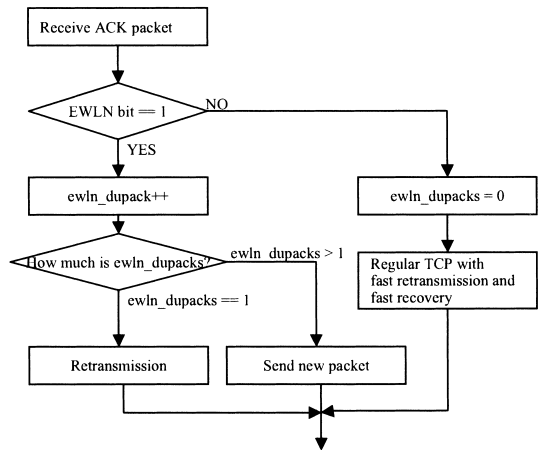


図 3 EWLN 情報に基づくパケット再送を行うフローチャート  
Fig.3 Flow chart of retransmission strategy based on EWLN message.

なければ、*ewln\_flag* の値が前の値のままにキープされる。次に、ACK の EWLN ビットは *ewln\_flag* に基づいて設定されて、最後に EWLN 情報を持つ ACK パケットが TCP 送信側に送られる。EWLN ビット = 1 の ACK が無線リンク上でビット誤りによるパケットロスを発生したことを TCP 送信側に知らせるとともに、そのパケット番号を含んでいるため、TCP 送信側がただちにロスしたパケットの再送を実行できる。

### 3.2.2 TCP 送信側

ACK に含まれる EWLN 情報を受け取ると、TCP 送信側は輻輳制御メカニズムを呼び出さず、再送のみを実行する。EWLN が無線リンク上でビット誤りによるパケットロスを明示的に TCP 送信側に通知しているため、TCP 送信側は通常の TCP のように 3 DUPACKS が届くまで待つ必要がなく、EWLN が設定されている最初の DUPACK を受けると、すぐパケットの再送を実行できる。重複再送を避けるために、EWLN によってパケットロスを示す最初の DUPACK が届くときにのみ、再送が行われる。EWLN ビットがセットされていない ACK パケットを受け取る場合、輻輳関連のパケット損失に対して通常の TCP の早期再送と早期回復が実行される(図 3 を参照)。

図 4 は 1 つの送信ウィンドウに 2 つのパケット損失が無線リンク上に発生する場合、EWLN-Reno によるパケットの流れと確認応答の例を示す。EWLN-Reno は最初の EWLN=1 の DUPACK を受信すると、3 つの DUPACKS を待たずに、ただちに再送を行う。しかし、Reno ベースの TCP が累積 ACK を使用するため、2 つ目以降の無線リンクロスに関する EWLN 情報が TCP 受信側に届いても無視されてしまう。従来

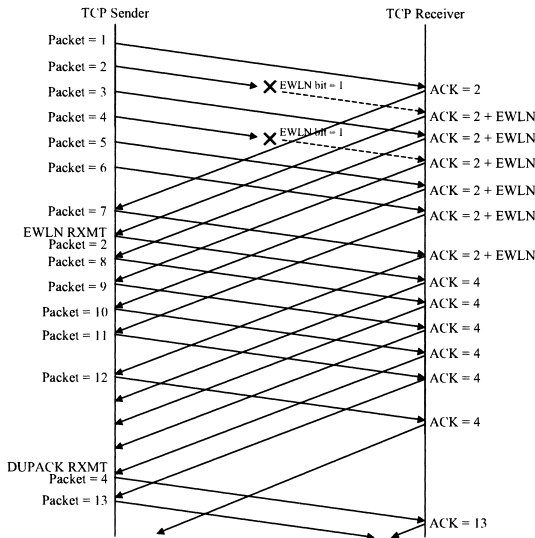


図 4 1つの送信ウィンドウに無線リンクロスが2個発生する場合、EWLN-Renoによるパケットの流れと確認応答

Fig.4 Transmission of data and acknowledgement packets in EWLN-Reno protocol when two packet losses occur over a wireless link in a single transmission window.

のTCP-Renoと互換性を持たせるため、2つ目以降の無線リンクロスに関するEWLN情報は利用できない。無線リンク上でロスしたすべてのパケットのEWLN情報を有効に使用するため、我々は次節で、高い性能を有するがコンパチブルでないRenoベースのEWLN実現方法について述べる。

図5は1つの送信ウィンドウに2つのパケット損失が無線リンク上に発生する場合に、EWLN-SACKによるパケットの流れと確認応答の例を示す。SACKオプションを使用するため、TCP受信側に順番どおりに到着したすべてのパケットがTCP送信側に知られる。したがって、SACKベースのTCPは実際に失われたすべてのパケット(正確には1つの送信ウィンドウに3つまで)を検知し、再送することができる。ここで注意すべき点は、SACKオプションが有効なパケットの到着によってトリガーされるため<sup>10)</sup>、TCP受信側がEWLN=1のパケットを受け取っても、次に正常なパケットを受信するまでACKパケットが生成できないことである。ACKに含まれるEWLN情報が届くと、TCP送信側は輻輳ウィンドウサイズを減少させず、再転送のみを実行する。EWLN-Renoと比較すると、EWLN-SACKスキームは1つの送信ウィンドウに複数のパケットロスによる不必要なタイムアウトを避けられるだけでなく、パケットのEWLN情報も有効に使用できる。

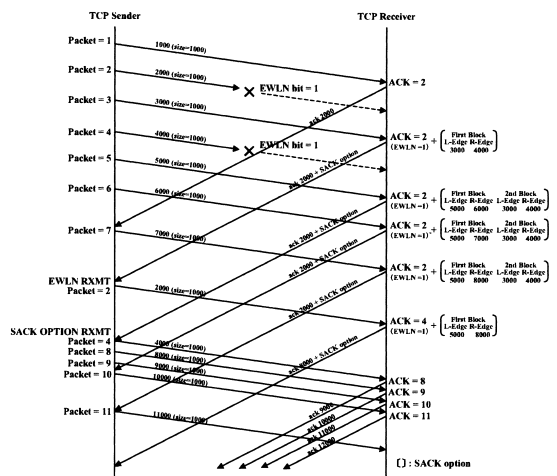


図 5 1つの送信ウィンドウに無線リンクロスが2個発生する場合、EWLN-SACKによるパケットの流れと確認応答

Fig.5 Transmission of data and acknowledgement packets in EWLN-SACK protocol when two packet losses occur over a wireless link in a single transmission window.

### 3.3 EWLN機能を持つ高性能なTCP-Renoの実現方法

前節では、我々はMAC層の情報に基づくコンパチブルなEWLNの実現方法について議論した。提案したEWLN-Renoが従来のTCPと互換性を持つため、EWLN情報を持つACKパケットを完全に利用することができなかった。それを改善するために、我々はペイロードなしのEWLNパケットの受信に対応して、TCP受信側で新しいACKを発生させるEWLN-Reno-Optimalを提案する。EWLN-Reno-Optimalはたとえ1つの送信ウィンドウに複数のパケット損失が無線リンク上で発生しても、コンパチブルなEWLN-Renoのように最初の無線パケット損失の再送が完了するまで待つのではなく、あらゆる新しいEWLNパケットに対応したACKパケットがTCP受信側で生成される。図6はEWLN情報に基づきEWLN-Reno-OptimalによるACKパケットを生成するフローチャートを示す。ここでは、パケットが順番に来ているどうかをチェックする代わりに、TCP受信側が新しいパケットを受け取ると、EWLNビットのチェックを最初に行う。EWLNが1に設定されれば、どのパケットが無線リンク上で失われているのかをTCP送信側に示すために、TCP受信側がEWLN情報を持つ新しいACKを生成するとともに、*ewln\_seqNo*を最新のものに更新する。ACKのEWLNビットがいったん設定されると、ACK落ちに耐えられるため、送信ウィンドウにおける最後の無線リンク上でロスしたパ

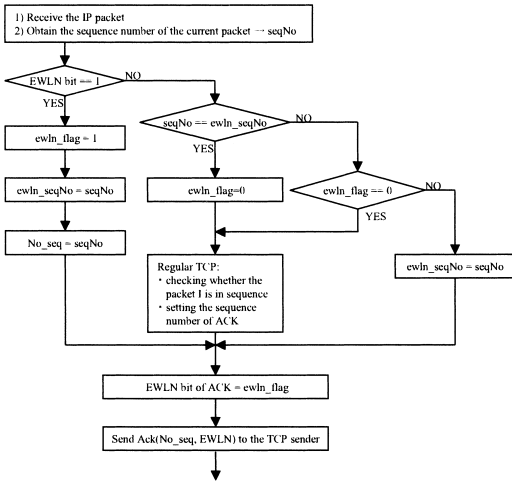


図 6 EWLN 情報に基づき EWLN-Reno-Optimal による ACK パケットを生成するフローチャート

Fig.6 Flow chart of generating ACK packet based on EWLN information for EWLN-Reno-Optimal scheme.

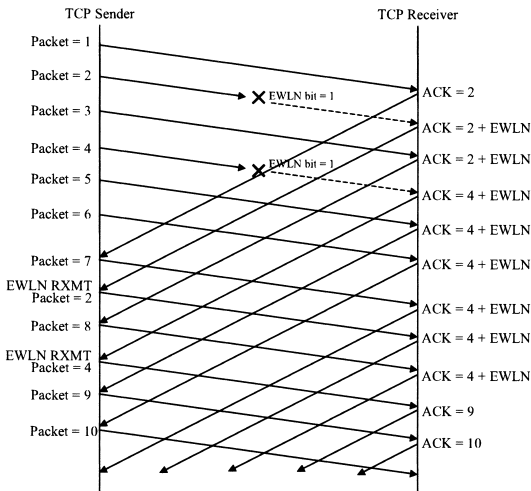


図 7 EWLN-Reno-Optimal によるパケットの流れと確認応答の生成

Fig.7 Transmission of data and acknowledgement packets in EWLN-Reno-Optimal protocol when two packet losses occur over a wireless link in a single transmission window.

ケットの再送が確認できるまで立て続ける．ここで、EWLN ビットがセットされている ACK パケットが TCP-Reno のように、すべての有効なパケットを確認する累積 ACK でなくなる点に留意すべきである．

図 7 は 1 つの送信ウィンドウに 2 つのパケット損失が無線リンク上で発生する場合、EWLN-Reno-Optimal によるパケットの流れと確認応答の例を示す．EWLN が 1 つの送信ウィンドウにおけるすべての無線リンクロ

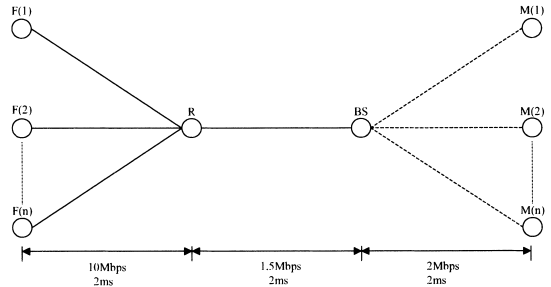


図 8 シミュレーションモデル  
Fig.8 Simulation topology.

スを TCP 送信側に明示的に知らせるため、TCP 送信側はただちに無線リンク上で落ちているすべてのパケットを再転送することができる．したがって、EWLN-Reno-Optimal の性能はコンパチブルな EWLN-Reno より大幅に向上させることが可能となる．

#### 4. 計算機シミュレーション

提案手法の有効性を検証するため、ネットワークシミュレータ ns-2<sup>11)</sup> を用いて、計算機シミュレーションによる評価実験を行った．実環境に近い状態で評価実験を行うため、DPSK ( a Differential Phase Shift Keying ) エラーモデル<sup>12)</sup> を適用し、それに対応して 802.11 MAC プロトコルと無線伝播モデルに追加・変更を加えた．DPSK エラーモデルに基づき、ビットエラーの平均確率は次式で表される．

$$P_{e,DPSK} = \frac{1}{2} \exp(-10 \frac{SNR}{10}). \quad (1)$$

図 8 は評価実験で使用されたネットワークモデル ( n=1, 2, 3 ) を示す．TCP 送信側 ( F(n), n=1, 2, 3 ) と基地局との間には有線リンクであるが、基地局と TCP 受信側 ( M(n), n=1, 2, 3 ) の間は無線リンクである．無線リンクにおいて、式 (1) より表されたビットエラーモデルが適用された．無線リンクの最大帯域幅は 2Mbits/s であり、TCP データパケットのサイズは 1Kbytes である．さらに、ACK パケットのサイズは 40 bytes で、TCP 送信側における最大輻輳ウィンドウサイズは 20Kbytes にセットされた．

評価実験は 2 つのシナリオを設定して行われた．1) パケットロスには主にビット誤りによる無線リンクロス ( TCP コネクションが 1 つある場合 ); 2) 輻輳と無線リンク上のビット誤りの両方によるパケット損失 ( TCP コネクションが 2 つ以上ある場合 ) がある．以下、それぞれについて詳しく述べる．

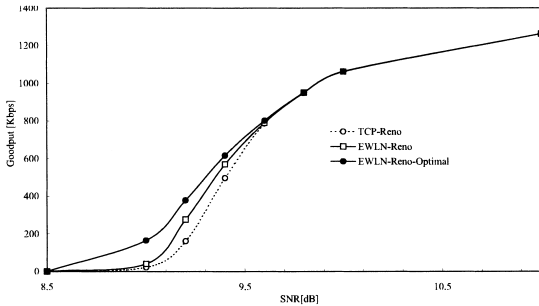


図 9 1 本の FTP 通信を行う無線ネットワークにおける EWLN 機能を持つ TCP-Reno による性能改善

Fig. 9 Performance improvement of EWLN-capable TCP-Reno protocols over the simulation topology having one FTP connection.

4.1 有線と無線が混在するネットワークにおける評価実験

提案した EWLN 手法の効果を評価するために、図 8 に示すネットワークポロジで M(2) から F(2) に、1 本の FTP よりデータ転送を行い、TCP スループットを性能指標として測定した。ここで TCP スループットとは、送受信ノード間の送信成功パケット数をバイト数に換算し、シミュレーション時間で割った値を用いる。シミュレーション時間が 80 秒で、各プロトコルにおいてシミュレーションを 100 回を行い、その平均値をグラフにプロットした。図 9 に SNR 値を変化させた場合の、TCP-Reno ベースのプロトコルによるスループットの変化を示す。この結果より、SNR が 8.5 dB から 9.6 dB までの範囲において、EWLN-Reno と EWLN-Reno-Optimal プロトコルの両方が TCP スループットを大きく改善させることが分かる。最も良いケース (SNR=9 dB) では、EWLN-Reno-Optimal プロトコルによるスループットが TCP-Reno より 7.4 倍、EWLN-Reno によるスループットが 1.8 倍以上、向上することが明らかとなった。

図 10 に SNR 値を変化させた場合の、SACK ベースのプロトコルによる TCP スループットの変化を示す。この結果より、SNR が 8.5 dB から 9.6 dB までの範囲において、EWLN-SACK プロトコルがスループットを大きく改善させることが分かる。最も良いケース (SNR=9 dB) では、EWLN-SACK によるスループットが TCP-SACK より 3 倍以上向上することが明らかとなった。

図 9 と図 10 の両方ともに、SNR 値が大きくなるにつれて、提案手法による TCP スループットの改善効果が減少し、SNR 値が 11 dB に達すると、すべてのプロトコルによるスループットの結果がほぼ同じにな

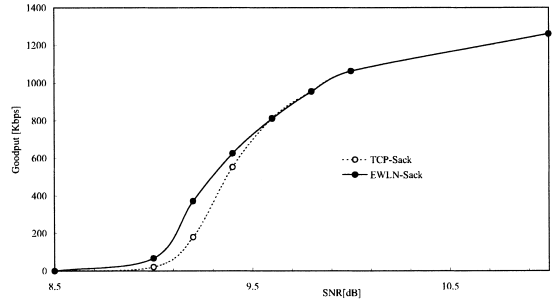


図 10 1 本の FTP 通信を行う無線ネットワークにおける EWLN 機能を持つ TCP-SACK による性能改善

Fig. 10 Performance improvement of EWLN-SACK protocol for the simulation topology having one FTP connection.

表 1 1 本の FTP 通信を行う無線ネットワークにおける EWLN 機能を持つ TCP-Reno と TCP-SACK によるスループットの比較結果

Table 1 Performance improvement of EWLN-capable protocols for the simulation topology having one FTP connection.

SNR (dB)	スループット(kbps)				
	TCP-Reno	EWLN-Reno	EWLN-Reno-Optimal	TCP-SACK	EWLN-SACK
8.5	0.0	1.5	2.1	0.0	0.0
9.0	46.0	89.2	230.6	46.7	139.5
9.2	280.7	400.7	447.6	306.2	453.1
9.4	631.1	644.8	651.7	649.9	657.8
9.6	824.1	822.3	821.8	828.2	826.7
9.8	963.2	963.6	962.8	963.8	965.7
10.0	1069.8	1069.8	1068.7	1068.0	1069.3
11.0	1262.1	1262.5	1262.5	1264.1	1264.0

ることを示している。さらに、Reno ベースの手法と SACK ベースの手法を比較するため、図 10, 11 における評価結果を表 1 にまとめた。表 1 より、SNR が小さいとき (SNR ≤ 9.2)、すなわち誤り率が高い場合、EWLN-Reno-Optimal が EWLN-SACK よりも高い性能を示すことが分かる。その理由は EWLN-Reno-Optimal が無線リンクロスをただちに再送できるのに対して、EWLN-SACK が次に届く有効なパケットが到着するまで、無線リンクロスを送再送できないことにある。したがって、SNR が 9.2 dB より小さい場合、EWLN-SACK の性能が EWLN-Reno-Optimal より劣化しており、9.2 dB < SNR ≤ 9.8 dB の場合において、EWLN-SACK の性能が EWLN-Reno-Optimal より優れており、さらに SNR が 9.8 dB より大きくなると、無線ネットワークの性能が有線ネットワークに近づくため、輻輳によるパケットロスがほとんど生じない M(2) から F(2) まで 1 本のみ ftp 通信において、提案手法も従来手法も同じ性能を示す結果となった。



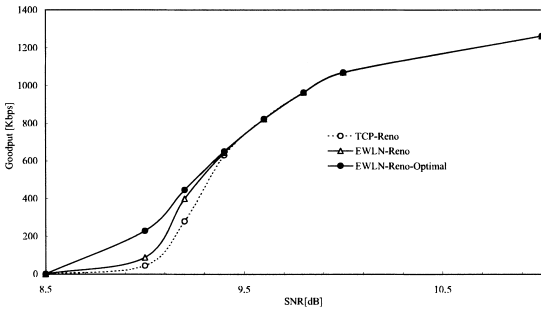


図 11 2本のFTP通信を行う無線ネットワークにおけるEWLN機能を持つTCP-Renoによる性能改善

Fig. 11 Performance improvement of EWLN-capable TCP-Reno protocols for the simulation topology having two FTP connections.

4.2 ボトルネックがあるネットワークにおける評価実験

前節より、TCPコネクションが1つで、輻輳によるパケットロスがほとんど生じない有線と無線が混在しているネットワークにおいて、EWLN機能を持つTCPの有効性が明らかとなった。輻輳によるパケットロスと無線リンクロスの両方が生じる場合について、提案手法の有効性を検証するため、ボトルネックがあるネットワークにおいても同様な評価実験を行った。この場合、図8に示すネットワークモデルでF(1)からM(1)、ならびにF(3)からM(3)までの2本のFTPより同時にデータ転送を行い、トータルなTCPスループットを測定した。

図11はSNRを変化させた場合、TCP-Renoベースのプロトコルを用いることにより、2本のftpコネクションのトータルなスループットの変化を示す。図より、SNR値が8.5dBから9.6dBまでの範囲において、EWLN-RenoプロトコルとEWLN-Reno-Optimalプロトコルの両方がスループットを大きく改善させることが明らかとなった。

図12にSNR値を変化させた場合の、SACKベースのプロトコルによる2本のftpコネクションのトータルなスループットの変化を示す。図より、EWLN-SACKプロトコルがSNR値が8.5dBから9.6dBまでの範囲において、スループットを大きく改善させることが明らかとなった。

さらに、提案手法の性能改善効果が輻輳の状態にどれだけ影響を受けるかについても調べた。そのため、図8に示すネットワークモデルでF(1)からM(1)、F(2)からM(2)、ならびにF(3)からM(3)までの3本のFTPより同時にデータ転送を行い、トータルなTCPスループットを測定した。その結果を表2に示す。こ

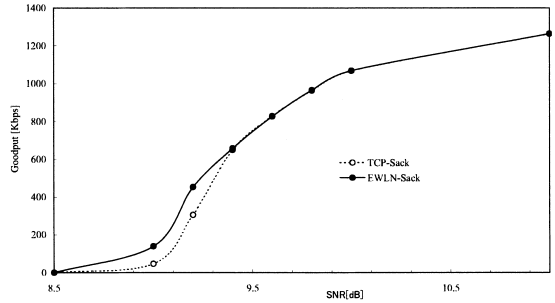


図 12 2本のFTP通信を行う無線ネットワークにおけるEWLN機能を持つTCP-SACKによる性能改善

Fig. 12 Performance improvement of EWLN-SACK protocol for the simulation topology having two FTP connections.

表 2 TCP-Reno ベースの EWLN 方式によるスループット向上率の変化

Table 2 Change in performance improvement rate for EWLN-capable TCP-Reno protocols.

SNR (dB)	スループットの向上率					
	EWLN-Reno-optimal/TCP-Reno			EWLN-Reno/TCP-Reno		
	n=1	n=2	n=3	n=1	n=2	n=3
8.5	8.00	21/0.0	36/0.0	7.00	15/0.0	19/0.0
9.0	7.41	5.01	4.03	1.82	1.94	1.89
9.2	2.34	1.59	1.28	1.71	1.43	1.22
9.4	1.24	1.03	1.00	1.15	1.02	1.00
9.6	1.02	1.00	1.00	1.00	1.00	1.00
9.8	1.00	1.00	1.00	1.00	1.00	1.00
10.0	1.00	1.00	1.00	1.00	1.00	1.00
11.0	1.00	1.00	1.00	1.00	1.00	1.00

こでスループットの向上率とは、提案手法によるスループットを従来のTCP-Renoによるスループットで割ったものを用いる。表2より、TCPコネクションの数が多いほど、EWLNの改善効果が減少していくことが分かる。それはTCPコネクションの数が増えると、輻輳がより深刻になり、無線ビット誤りによるパケットロスの割合が減少していることが考えられる。特に、EWLN-Reno-Optimalの改善効果が輻輳状態による影響が一番大きく、EWLN-Renoがそれほど影響を受けないことが見られた。また、どのような輻輳状態においても、EWLN法が従来の手法より有効であることが確認できた。さらに、SACKベースのプロトコルにおいても、同様な実験を行い、その結果を表3に示す。表より、同様に提案手法の性能改善効果は輻輳状態の変化により影響を受けるが、EWLN-Reno-Optimalほどの変化が見られなかった。また、Renoベースのプロトコルと同様にどのような輻輳状態においても提案手法が従来のTCP-SACKより有効であることが確認できた。

表 3 TCP-SACK ベースの EWLN 方式によるスループット向上率の変化

Table 3 Change in performance improvement rate for EWLN-capable TCP-SACK protocol.

SNR (dB)	スループットの向上率		
	EWLN-SACK/TCP-SACK		
	n=1	n=2	n=3
8.5	0.0	0.0	0.0
9.0	3.13	2.99	2.61
9.2	2.06	1.48	1.21
9.4	1.13	1.01	1.00
9.6	1.00	1.00	1.00
9.8	1.00	1.00	1.00
10.0	1.00	1.00	1.00
11.0	1.00	1.00	1.00

## 5. む す び

本論文では、無線ネットワークにおける TCP の性能劣化問題を解決するため、EWLN ベースのエンドツーエンドなアプローチを提案した。提案手法は MAC プロトコルのエラー検出メカニズムを利用することにより、無線リンクロスを明示的に通知する EWLN 方式の実現を可能にした。特に、TCP-Reno と TCP-SACK プロトコルにおいてコンパチブルな実現方法、さらにコンパチブルではないが、高性能な EWLN の実現方法についての検討も行った。ns-2 ネットワークシミュレータを用いたシミュレーション実験において、我々はエラーモデルとして DPSK (Differential Phase Shift Keying) の実装を行った。パケットの損失が主に伝送エラーであるシミュレーション実験では、SNR の範囲が [8.5 dB, 11 dB] の場合、EWLN-Reno-Optimal と EWLN-SACK の両方が TCP のスループットを大幅に改善させることが明らかとなった。ボトルネックのあるネットワークにおけるシミュレーション実験では、ビット誤りと輻輳の両方によるパケット損失が発生する場合、提案方式の有効性が確認できた。さらに、輻輳の状態が提案手法の性能改善に与える影響も調べた。その結果、輻輳が深刻なほど、改善効果が減少していくことが見られたが、どのような状況においても、提案手法は従来の手法より優位であることが分かった。有線と無線が混在するネットワークにおけるシミュレーション実験の結果が良好なため、今後無線アドホックネットワークにおいても同様な実装と評価、さらにセキュリティ問題も含めて検討していく予定である。

謝辞 本研究を進めるにあたり、日ごろからご鞭撻

いただく弊所小宮山所長、有益な助言をいただいた弊所第一ならびに第三研究室員各位に深謝する。また、本研究は通信・放送機構の研究委託により実施したものである。

## 参 考 文 献

- 1) Jacobson, V.: Congestion avoidance and control, *ACM SIGCOMM'88* (Aug. 1988).
- 2) Bakre, A. and Badrinath, B.R.: I-TCP: Indirect TCP for Mobile Hosts, *Proc. 15th ICDCS* (May 1995).
- 3) Balakrishnan, H., Seshan, S. and Katz, R.H.: Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks, *ACM Wireless Networks*, Vol.1, No.4 (1995).
- 4) Bansal, D., Chandra, A. and Shorey, R.: An extension of the TCP flow control algorithm for wireless networks, *Proc. IEEE Inter. Conf. on Personal Wireless Communications* (1999).
- 5) Balakrishnan, H. and Katz, R.H.: Explicit Loss Notification and Wireless Web Performance, *Proc. IEEE Globecom Internet Mini-Conference* (Nov. 1998).
- 6) Zhang, B. and Shirazi, M.N.: Implementation of Explicit Wireless Loss Notification Using MAC-Layer Information, *IEEE WCNC 2003* (Mar. 2003).
- 7) DeSimone, A., Chuah, M.C. and Yue, O.C.: Throughput performance of transport-layer protocols over wireless LANs, *Proc. IEEE Globecom* (Nov. 1993).
- 8) Stevens, W.R.: TCP slow start congestion avoidance, fast retransmission and fast recovery algorithms, RFC2001 (Jan. 1997).
- 9) Balakrishnan, H., Padmanabhan, V.N., Seshan, S. and Katz, R.H.: A comparison of mechanisms for improving TCP performance over wireless links, *IEEE/ACM Trans. Networking*, Vol.5, No.6 (1997).
- 10) Mathis, M., Mahdavi, J., Floyd, S. and Romanow, A.: Selective acknowledgement options, RFC2018 (1996).
- 11) UCB/LBNL/VINT Network Simulator ns (version2). <http://www.isi.edu/nsnam/ns/>
- 12) Rappaport, T.S.: *Wireless Communications: Principles & Practice*, Prentice Hall (1996).

(平成 15 年 7 月 17 日受付)

(平成 16 年 3 月 5 日採録)

**張 兵**

昭和 37 年生。昭和 58 年中国北京航空航天大学卒業。平成 2 年広島大学大学院博士課程修了。平成 3 年郵政省通信総合研究所（現（独）通信総合研究所）入所。以来、統計的画像処理、ニューラルネットワークの研究に従事。その間、平成 7 年～8 年米国テネシー大学神経・解剖学科博士研究員。平成 12 年より ATR 適応コミュニケーション研究所に出向。現在、適応的 QoS 制御方式、無線 TCP の研究に従事。主任研究員，工学博士。電子情報通信学会会員。

**田中 信介**

昭和 28 年生。昭和 55 年京都大学大学院工学研究科金属加工学専攻修士課程修了。同年国際電信電話株式会社（現 KDDI 株式会社）に入社。光磁気ディスク記録媒体および通信用半導体光デバイスの研究開発に従事。平成 14 年より国際電気通信基礎技術研究所（ATR）の適応コミュニケーション研究所第一研究室室長。無線アドホックネットワークに関する研究開発に従事。平成 6 年社団法人発明協会内閣総理大臣発明賞受賞。電子情報通信学会，応用物理学会各会員。

**マハガドヌリ シラジ（正会員）**

昭和 38 年生。平成 2 年鳥取大学大学院電気工学科修了。平成 5 年神戸大学大学院自然科学研究科博士課程修了。同年科学技術庁特別研究員。平成 7 年郵政省通信総合研究所（現（独）通信総合研究所）入所。以来、統計的画像処理、顔認識、情報秘匿の研究に従事。その間、平成 13 年～14 年米国カリフォルニア大学バークレー校リサーチフェロー。現在、適応的 QoS 制御方式、無線 TCP の研究に従事。

