

無線ネットワークにおける TCP の改善に関する考察

西 田 佳 史†

無線技術の発達に伴い、無線ネットワークにおける TCP/IP の通信性能の向上は大きな課題となっている。しかし、高信頼性の通信サービスを提供する TCP は、有線ネットワークに適應する様に設計されており、無線ネットワークでは性能を発揮できない場合がある。本稿では、無線ネットワーク上での TCP の転送性能を向上させる様々な手法について解説する。

Thought of TCP enhancement in wireless networks

YOSHIFUMI NISHIDA †

As a result of the rapid advancement of wireless technology, the importance of the technologies that improve the performance of TCP/IP in wireless networks is increased. However, TCP which provide reliable transport service is tuned to perform well in wired networks. Thus using TCP in wireless networks would suffer significant performance degradation. In this paper, we describe and clarify some approaches that improve the performance of TCP in wireless networks.

1. はじめに

無線技術の発達から、今日無線ネットワークは非常に身近なものとなり、TCP/IP を無線ネットワーク上で利用する状況が増えつつある。このような状況の中、無線ネットワーク上における TCP/IP の通信性能の向上は大きな課題となっている。TCP は、TCP/IP プロトコルスタックの中で、高信頼性の通信サービスを提供するトランスポート層プロトコルとして長い歴史を持っている。TCP は、フロー制御や輻輳制御の機能により、様々な通信状況に対応できるように設計されている。しかし、TCP は有線ネットワークや固定されたノードによる通信に適應しやすい様に設計されているため、無線ネットワーク上では逆に TCP の通信性能が低下する場合がある。本稿では、無線ネットワーク上で TCP を用いて通信する場合の問題点と、現在までに提案されている解決策について解説する。

2. 無線ネットワークにおける TCP の問題

無線ネットワークは、有線ネットワークと異なる特性があり、このことが無線ネットワークにおける TCP の通信性能を劣化させる要因となっている。

無線ネットワークの転送性能の最も大きな障害は高いビット誤り率である。通常、無線ネットワークにおけるビット誤り率は、有線ネットワークのそれと比較して非常に高い。有線ネットワークでは、ビット誤りはほとんど起こらないことから、TCP では、パケットの喪失を通信回線上的エラーとは見做さず、ネットワークの輻輳と判断する様に設計されている。TCP の輻輳制御機構は、輻輳崩壊を避けるため輻輳を検出すると通信速度を大きく低下させる。このため、高いビット誤り率を持つ無線ネットワーク上で TCP を利用する場合、輻輳制御機構の誤作動から、通信性能の劣化を招いてしまう。

無線ネットワークを利用する通信機器には、PDA やノート型パソコンなど可搬性の高いものが多い。これらの通信機器を用いる場合、移動をしながら通信を行うことがある。Mobile IP⁷⁾ の様な技術を利用することにより、IP を用いた通信は移動透過性を持ち、ノードが移動しても通信を継続することができる。しかし、ノード間の移動は TCP の性能に大きな影響を与える可能性がある。通信機器が移動すると、ハンドオフや障害物などの存在により、一時的に通信不能な状態が生じ、連続的にパケットが喪失することがある。TCP は、連続的なパケットの喪失を検出すると、指数的に再送タイムをバックオフさせる。このため、通信障害が解決された後に直ちに通信を再開することが

† ソニーコンピュータサイエンス研究所
Sony Computer Science Laboratories, Inc

できず、通信性能が大きく低下することがある。

3. 無線ネットワークにおける TCP の改善策

無線ネットワークにおける TCP の改善策はこれまでに数多く提案されている。本節では、これらの改善策をアプローチの方向性によって分類して述べる。

3.1 End-to-End モデルによるアプローチ

TCP は、中継ノード上に TCP コネクションの情報やコネクション毎の特別な処理を要求しない。TCP の状態は、通信を行う端末ノードの間でのみ把握される。End-to-End モデルによる解決策は、このような TCP が従来持っている通信モデルを保ちつつ、無線ネットワークにおける通信性能を改善する。End-to-End モデルのアプローチは、アーキテクチャが簡素で中継システムに要求される機能が少ないことが長所であるが、通信するノードの双方の TCP スタックに修正が必要なことや、ハンドオフに対応できないといった欠点がある。このようなアプローチには、次の様なものがある。

3.1.1 Explicit Loss Notification

TCP の大きな課題は、輻輳によるパケット喪失とビット誤りから生じるパケット喪失を区別できないことである。パケットの喪失がネットワークの輻輳から生じた場合は、輻輳崩壊を避けるため転送速度を落とし、ネットワークの状態が回復するまで待機する必要がある。しかし、ビット誤りからパケット喪失が生じた場合は、このような操作は必要ではない。

Explicit Loss Notification は、無線リンクで生じたビット誤りの情報をトランスポート層で把握することにより、パケット喪失の原因を判別する。もし、ビット誤りからパケット喪失が生じた場合、受信側の TCP は確認応答を返送する際に、オプションやヘッダ中のビットを利用し、パケットの喪失が輻輳によるものではないことを送信側に通知する¹⁾²⁾。Explicit Loss Notification の情報を受信した TCP は、輻輳制御を行わず、指定されたパケットの再送のみを行う。

ELN を用いた方式の問題点は、TCP を用いて通信する両端のノードに修正が必要なことである。またこの技術の標準化は現時点では進んでいない。

3.1.2 Explicit Congestion Notification

Explicit Congestion Notification では、中継ルータがルータ中の待ち行列の平均長を監視し、待ち行列の長さが一定値を超えると、Congestion Experience Bit をパケット中にセットする³⁾。ECN を用いることにより、TCP は、ルータ中の待ち行列が溢れる前に輻輳の存在を把握することができる。ECN によって

輻輳の存在を知った TCP は、転送速度を調節し輻輳によるパケット喪失の可能性を下げるができる。

このような ECN の機能は、無線ネットワークにも応用できる。ECN を受けた後にパケットの送出手を抽出した場合、TCP はパケットの喪失を輻輳によるものと判断し、輻輳制御を行う。しかし、ECN を受けずにパケット喪失を検出した場合、TCP はパケットが輻輳以外の原因によって喪失したものと見做し、輻輳制御を行わずに単純にパケットの再送だけを行う。

ECN を用いた方式の問題点は、ELN と同様に両端のノードに修正が必要となることである。ECN の標準化は比較的進んでおり、実装も広まりつつあるもののまだ十分とはいえない。また、この方式は、中継ノードが ECN をサポートしていない、何らかの不具合により、Congestion Experience Bit をセットできない場合や、急速な輻輳の発生により、待ち行列が溢れ Congestion Experience Bit をセットしたパケットを廃棄せざるを得ない状況において、TCP が輻輳に反応できないという大きな問題を含んでいる。

3.1.3 選択確認応答方式

TCP のオリジナルの設計は、累積型の確認応答方式を利用している。累積型確認応答では、TCP のウィンドウ中に複数のパケット喪失が存在した場合、喪失したパケットを正確に把握することができず、再送タイムアウトを招いてしまう場合がある⁴⁾。このような問題に対処するため、M. Mathis らは、選択確認応答オプション方式による解決方法を提案した⁵⁾。選択確認応答オプションでは、TCP オプションを用いて、確認応答パケットに TCP が正しく受信できたパケットの正確な情報を通知する。

3.1.4 最大セグメントサイズの選択

TCP が転送するパケット長は、コネクションセットアップ時に交換される最大セグメント長によって決定される。

高信頼性の通信サービスを提供する TCP は、パケット全体のチェックサムを計算し、チェックサムの合わないパケットは廃棄される。ビット誤り率が低い有線ネットワークでは、パケットヘッダのオーバーヘッドを抑えるため、できるだけ大きなパケットサイズを選択する。しかし、ビット誤り率が高い場合は、大きなパケット長を選択するとパケットの廃棄率が高くなり、転送性能が劣化してしまう。この様なことから、データリンク層のビット誤り率に合わせて、パケット長を適切に選択することにより、TCP の通信性能を向上できる。データリンク層の制限からビット誤り率を正確に把握することができない場合、ビット誤りによる

パケット廃棄率から、ビット誤り率を求めることも可能である⁶⁾。

最大セグメントサイズを調節する方式は、無線ネットワークに接続されたノードの TCP に変更を加えるだけで実現可能である。しかし、最大セグメント長はコネクションの途中では変更できないため、急速なエラーレートの変化に対応できないなどの問題がある。

3.2 データリンク層の利用

無線ネットワークのデータリンク層で FEC や再送の機能を提供することによって、TCP 自体に修正を加えずに、無線ネットワークにおける TCP の転送性能を向上できる可能性がある。しかし下位層による再送機能の提供には、上位層の TCP との関係が重要な課題となる。TCP との関係を考えていない場合は、以下にあげる問題によって逆に転送性能を劣化させてしまう場合がある。

● ラウンドトリップタイムの増加

TCP は、通信中にラウンドトリップタイムの計測を行い、その結果から再送タイムアウト時間を決定する。下位層による再送機能の提供は、高い信頼性を提供する反面、TCP の計測するラウンドトリップタイムを増加させてしまう。TCP は、1 ウィンドウ中に複数のパケット喪失があった場合など、再送タイムアウトによってパケット喪失を検出せざるを得ない場合がある。ラウンドトリップタイムの増加は、再送タイムアウトを検出するまでの時間を増加させるため、大きな転送性能の劣化を招いてしまうことがある。

● タイマアウトによる再送

下位のデータリンク層と上位層の TCP で個別の再送機構を持つことは、冗長な再送を生じさせる場合がある。例えば、データリンク層でパケット喪失を検出し、確認応答を待つ間に、TCP の再送タイムアウトが起こり同じパケットの再送を行ってしまう場合がある。

● 高速再転送アルゴリズムの誤作動

下位層でのパケットの再送機構が、単純にタイムアウトしたパケットの再送を行うものであった場合、この機構は上位層に対して、パケットの順序通りの受信を保証できない。このような場合、TCP の転送性能は大きく劣化する可能性がある。

TCP は、多くの場合パケットが順序通りに配達されることを想定している。パケットの転送順序が大きく異なると、TCP はパケットが輻輳によって失われたものと判断し、高速再転送アルゴリズムによって受信の遅れたパケットの再送を行って

しまう。そして、パケットの再送後は、存在しない輻輳に備えるため、転送速度を半分に減少させてしまう。

3.3 コネクション分割型アプローチ

コネクション分割型のアプローチは、固定ノードと移動ノードの間の TCP コネクションを2つの異なる TCP コネクションに分割する方式である。この手法では、ベースステーション (もしくはその近傍のノード) が、データの中継ノードとして特別な役割を果たす。TCP のコネクションは、有線ノードと中継ノードの間のコネクションと中継ノードと無線ノードのコネクションに分割される。中継ノードは、2つのコネクションから送られるデータを中継することにより、終端ノードに対して仮想的に2つのコネクションを1つのコネクションに見せる。

このようなアプローチの具体的な実装には、I-TCP⁸⁾ や M-TCP⁹⁾ がある。両者はほぼ同様のアーキテクチャを持つが、細部の実装に違いがある。まず、I-TCP では、2つのコネクションは、非同期的に通信を行う。I-TCP の場合、中継ノードの TCP はデータパケットを受信すると、送信ノードに対し直ちに確認応答を送信する。これに対し、M-TCP は2つのコネクション間で簡単な同期がとられる。M-TCP の場合、中継ノードはデータパケットのシーケンス番号を監視し、これまでに一方のコネクションから受信した最大のシーケンス番号に対する確認応答は、他方のコネクションからの確認応答を受信するまでは送信しない。それ以外のデータパケットに対する確認応答は、他方のコネクションとの同期をとることなく、ただちに転送される。このような M-TCP の手法は、End-to-End 型の通信モデルのセマンティクスを破壊するコネクション分割型アプローチの欠点を緩和する目的を持っている。コネクション分割型アプローチの欠点については、3.3.3 節で詳しく述べる

3.3.1 コネクション分割型アプローチによるハンドオフ

コネクション分割型のアプローチによるハンドオフの手順を図 3.3.1 に示す。図 3.3.1 では、固定ノード F と移動ノード M が TCP による通信を行っている。ノード F とノード M の間の通信は、中継ノード I1 によって、ノード F-ノード I1 間とノード M-ノード I1 間の2つの TCP コネクションに分割される。移動ノード M が新しいネットワークに移動すると、固定ノード F との通信は、新しい中継ノード I2 によって分割される。移動ノード M が新しいネットワークに移動する間、一時的に通信障害が生じ、移動ノードへの

データ転送が行われなくなる。この間に固定ノード F から転送されたデータは、中継ノード I1 によって保存される。ノード I1 に保存されていたデータは、ノード M の移動終了後に、新しい中継ノード I2 に転送される。このような手法は、固定ノードからハンドオフの発生を完全に隠蔽することができる。以下に移動ノード M が新しいネットワークに移動した直後の時点からのハンドオフの詳細な手順を以下に説明する。

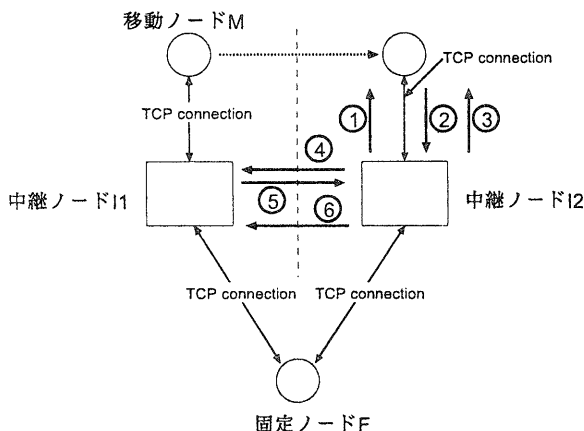


図 1 コネクション分割型アプローチによるハンドオフ手順

- (1) 中継ノード I2 から送られるビーコンを移動ノード M が受信する。(図中 1)
- (2) 移動ノード M は、中継ノード I2 に対して、通信中の TCP コネクションの情報と、移動前の中継ノード (ノード I1) のアドレスを通知する。(図中 2)
- (3) 中継ノード I2 は、ノード M に対しメッセージの受信通知を送信し、受信したコネクション情報に基づき、ノード M とノード F 間を結ぶ TCP コネクションのエンドポイントを設定する。(図中 3)
- (4) 中継ノード I2 は、ノード I1 に対し、forwarding pointer の登録要求を送信する。(図中 4)
- (5) ノード I1 は、forwarding pointer を登録する。そして、ノード M に関する全ての TCP コネクションを停止し、ノード M のコネクション情報をノード I2 に転送する。(図中 5)
- (6) ノード M のコネクション情報を全て受信したノード I2 は、ハンドオフ手順終了の確認応答をノード I1 に転送する。

3.3.2 コネクション分割型アプローチの長所

コネクション分割型のアプローチは次の様な長所を持つ。

- TCP のコネクションを 2 つに分割することにより、無線ネットワークで生じるパケット喪失に素早く対応でき、転送性能を向上させることができる。また、無線ネットワーク間の通信に、無線ネットワークに特化した TCP や、特別な通信プロトコルを利用することにより、さらに転送性能を向上させることができる。この手法は、移動ノードの処理能力が低い場合に、無線ネットワーク上で簡素な通信プロトコルを利用することにより、通信性能を高めることができる。
- コネクションを分割する中継ノードで、一時的にパケットを保存することにより、無線ネットワーク上で生じる通信障害にも対応することができる。また、この中継ノード上で TCP コネクションの情報を管理し、中継ノード間で管理する TCP コネクションの状態を交換することにより、スムーズなハンドオフを実現することができる。
- このアプローチは、中継ノードと移動ノード上のプロトコルスタックに変更を加えるだけで実現でき、固定ノードには一切変更を加える必要がない。有線ネットワーク上に存在する無数の固定ノードに変更を要求しないアーキテクチャは、普及が容易である。

3.3.3 コネクション分割型アプローチの短所

上述した様に、コネクション分割型のアプローチには様々なメリットがあるものの、同時に以下の様な問題点も持っている。

- コネクション分割型のアプローチは、TCP が本来持っている End-to-End の通信モデルを破壊してしまう。End-to-End 型の通信モデルが破壊されると、IPsec¹⁰⁾ などのセキュリティ技術を利用することが難しくなる。固定ノードと中継ノードの間と、移動ノードと中継ノードの間にそれぞれ Security Association を設定することにより、IPsec を利用することも不可能ではない。しかし、中継ノードに高い信頼性が要求されるため、状況によっては、この方法は利用することができない。
- コネクション分割型のアプローチは、End-to-End 型の通信モデルではないため、TCP の確認応答の意味が従来の TCP と異なってしまふ。このアプローチでは、固定ノードや移動ノードが確認応答を受信しても、実際に通信相手にパケットが送達したことは保証されない。このため、確認応答

の受信が重要な役割を持つアプリケーションでは、問題が起こる可能性がある。

- コネクション分割型の TCP コネクション 1 つに対して、中継ノードでは 2 コネクション分のトランスポート層の処理が必要となる。このため、中継ノードの処理負荷は、通常の TCP と比較して大きくなる。また中継ノードでコネクション毎の状態を管理する必要があるため、中継ノードの障害によって、TCP コネクションが致命的なダメージを受ける可能性がある。
- このアプローチは、固定ノードから移動ノードと移動ノードから固定ノードへの通信経路が同じ中継ノードを通過することを想定している。ルーティングなどの制限によって、このようなネットワークの構成がとれない場合、このアプローチを利用することができない。
- このアプローチでは、ハンドオフ時に移動前の中継ノードから移動後の中継ノードにコネクションの情報を転送する必要がある。中継ノードで一時的に保存しているデータがある場合など、転送する情報が多いとハンドオフに要する時間が増加する。

3.4 snoop 型アプローチ

Balakrishnan らは、コネクション分割型のアプローチに対し、より簡素なアーキテクチャを持つ snoop¹¹⁾ という手法を開発した。snoop 型のアプローチは、End-to-End 型の通信モデルを保ちながら、コネクション分割型の様に無線ネットワーク上で生じるパケット喪失に素早く対応することができる。snoop 型のアプローチでは、コネクション分割型アプローチと同様、中継ノードで TCP コネクションの監視を行うが、コネクションの分割は行わない。snoop では、固定ノードから移動ノードへのデータ転送と、移動ノードから固定ノードへのデータ転送に対して、それぞれ以下の様な操作を行う。

- 固定ノードから移動ノードへのデータ転送
中継ノードが、固定ノードから移動ノードに向かうパケットを受信した場合、中継ノードは、パケットを移動ノードに向けて転送すると同時にパケットのコピーをメモリ上に保存する。同時に中継ノードは、移動ノードから固定ノードへ送られる確認応答を監視する。中継ノードがタイムアウトや重複する確認応答パケットにより、無線ノード上でのパケット喪失を検出すると、保存しておいたデータパケットを再送する。移動ノードから送信された重複確認応答パケットは、中継ノード

によって廃棄され、固定ノードへは転送されない。この様に、中継ノードによる素早いパケットの再送と、無線ネットワーク上で生じたパケット喪失を固定ノードから隠蔽する操作により、TCP の転送性能の向上させることができる。

- 移動ノードから固定ノードへのデータ転送
無線ネットワーク上で生じるパケット喪失に対して、効率的なデータ転送を行うことができるように、中継ノードと移動ノード間の通信は、否定応答方式を利用してデータ転送を行う。中継ノードは、移動ノードから一定量のデータパケットを受信した場合や、一定時間経過しても期待するデータが受信できない場合に、否定応答を転送し再送を要求する。このため移動ノード側の TCP には、若干の修正が必要となる。

3.4.1 Snoop の通信モデル

図 3.4.1 に Snoop の通信モデルを示す。Snoop における移動ノードは、ホームアドレスと呼ばれる恒久的な IP アドレスを持つ。ホームアドレスは、ホームネットワーク上のアドレスであり、ホームネットワーク上には、ホームアドレスへのパケットの転送を管理するホームエージェントが存在する。

移動ノードと通信するノードは、ホームアドレスに対してデータを転送する。ホームエージェントは、Proxy ARP を用いてホームアドレス宛のパケットを受信し、移動ノードに関連する IP マルチキャストグループのアドレスでパケットをカプセル化して転送する。移動ノードに関連するマルチキャストグループには、移動ノードの周辺のベースステーションが参加している。

移動ノードは、ベースステーションから定期的に転送されるビーコンを計測することにより、最も近いベースステーションと近い将来移動する可能性のあるベースステーションを決定する。そして移動ノードは、これらのベースステーションに対して、指定したマルチキャストグループに参加するように要請する。最も移動ノードに近いベースステーションは、プライマリベースステーションとして選択される。プライマリベースステーションは、移動ノードへのパケットの転送を行う。他のマルチキャストグループに加入しているベースステーションは、受信したパケットを一定量保存するが、移動ノードへの転送は行わない。

ベースステーションから移動ノードへのデータ転送と、移動ノードから固定ノードへのデータ転送は、ユニキャストで行われる。

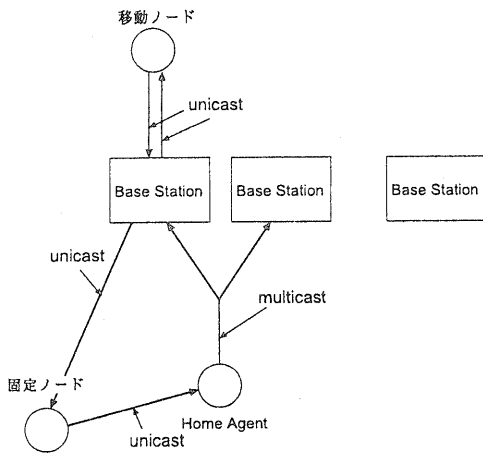


図 2 Snoop の通信モデル

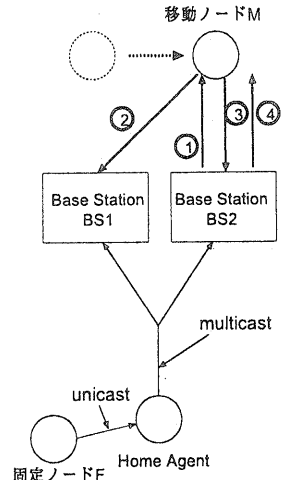


図 3 Snoop によるハンドオフ手順

3.4.2 Snoop によるハンドオフ

コネクション分割型のアプローチによるハンドオフは、中継ノード間で TCP コネクションの状態に関する情報と、キャッシュした TCP パケットを転送する必要があるため、ハンドオフに要する時間が大きくなる。これに対し、Snoop は、マルチキャストを応用した手法を用いることにより、短時間でのハンドオフを実現する。

図 3 に Snoop によるハンドオフの手順を示す。図 3 では、固定ノード F と移動ノード M が通信を行っている。ノード M は、ベースステーション BS1 のセル内におり、その後、BS2 のセル内に移動する。ノード M の移動前のプライマリベースステーションは BS1 である。BS2 は、BS1 と共通のマルチキャストグループに加入し、移動ノード宛のパケットを受信するが、転送は行わない。この場合のハンドオフの手順は次のようになる。

- (1) 移動ノード M は、ベースステーションからのビーコンを計測し、BS1 内のセルから BS2 内のセルに移動したことを検知する。(図中 1)
- (2) ノード M は、以前のプライマリベースステーション BS1 に移動を通知し、パケットの転送の中止と転送データの保存を依頼する。ノード M からのメッセージを受信した BS1 は、ノード M 宛のパケット転送を中止し、パケットの保存を開始する。(図中 2)
- (3) ノード M は、新しいプライマリベースステーションとなる BS2 に対し、パケットの転送依頼と、これまでにノード M が受信したデータの情報を転送する。BS2 は、これまでに保存し

たパケットのうち、ノード M が受信していないものを転送し、ノード M 宛のパケットの転送を開始する。

このように Snoop によるハンドオフの手法は、移動先のベースステーションが移動前からパケットを受信しているため、ノードの移動直後からハンドオフ中のデータを転送することができる。

3.4.3 Snoop 型アプローチの長所

Snoop 型のアプローチは、コネクション型アプローチと同様なメリットを持つ。Snoop 型アプローチでは、固定ノード側のプロトコルスタックに修正を加えずに、無線ネットワークにおける通信の転送性能を向上させることができる。中継ノードにおいてパケットの再送を行うことにより、無線ネットワークで生じるパケット喪失に素早く対応することができる。さらに中継ノードのサポートにより、ハンドオフ時に転送性能を劣化させることなく通信を継続することができる。

また、Snoop 型のアプローチは、コネクション分割型アプローチと異なり、中継ノードは、トランスポート層の機能を持たない。このため、Snoop の通信モデルは、End-to-End 型の通信モデルを破壊しない。このような設計は、中継ノードの負担を小さくすることができる。また中継ノードは、TCP コネクションの状態を持たないため、中継ノードの障害は TCP コネクションに大きな影響を与えない。

さらに、Snoop のマルチキャストを利用したハンドオフの手法は、コネクション分割型のハンドオフの手法と比較して手順が簡潔であり、ハンドオフに要するデータ転送量を減らし、ハンドオフの時間を短縮することができる。

3.4.4 Snoop 型のアプローチの短所

Snoop 型アプローチの欠点には、次のようなものがあげられる。

- Snoop 型アプローチは、中継ノードでパケットを監視する必要があるため、IPsec との併用が難しい。
- Snoop 型アプローチは、固定ノードと移動ノードの間の通信経路が対象でなければ利用できない。
- 中継ノードでパケットの再送を行うことにより、固定ノードからみたラウンドトリップタイムを増加させてしまう。W-TCP¹²⁾ は、Snoop 型のアプローチで、固定ノードが計測するラウンドトリップタイムの急速な変動を緩和することができる。しかし、W-TCP の利用には、TCP のタイムスタンプオプション¹³⁾ が必要となる。

4. ま と め

本稿では、無線ネットワークにおける TCP の様々な改善策について解説した。本稿で説明した改善策には、それぞれ長所と短所があり、状況に応じた使い分けが必要である。データリンク層による再送や FEC のサポートを実装することは、一般的に効果が高い。但し、3.2 節で述べた様に、データリンク層による再送機能の提供は、トランスポート層に対する影響を十分に考慮する必要がある。データリンク層での高信頼性のサービスが提供できない場合、ECN, ELN を用いた End-to-End 型のアプローチや Snoop 型のアプローチを利用することは効果的である。但し、End-to-End 型のアプローチには、固定ノードと移動ノードの双方のプロトコルスタックに修正が必要となる。また、Snoop 型のアプローチでは、固定ノードの修正は必要がないものの、ベースステーションと移動ノードのプロトコルスタックに修正が必要となる。構築するシステムの要件に応じて、実装が容易にできる手法を選択すべきである。

もし、構築するシステムがハンドオフでのパフォーマンスを重要視するならば、Snoop 型のアプローチが適している。また、無線ノードに処理能力の低いデバイスを利用する場合は、コネクション分割型アプローチを使って、TCP よりも軽いトランスポートを実装すると効果的である。

参 考 文 献

- 1) Hari Balakrishnan, Venkat Padmanabhan, Srinivasan Seshan, and Randy H. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links," IEEE/ACM Transactions on Networking, December 1997.
- 2) Hari Balakrishnan and Randy H. Katz, "Explicit Loss Notification and Wireless Web Performance," Proc. IEEE Globecom Internet Mini-Conference, Sydney, Australia, November 1998.
- 3) S. Floyd, "TCP and Explicit Congestion Notification," ACM Computer Communication Review, V. 24 N. 5, October 1994, p. 10-23.
- 4) K. Fall and S. Floyd. "Simulation-based Comparisons of Tahoe, Reno, and Sack TCP," Computer Communications Review, 1996
- 5) M. Mathis, J., Mahdavi, S. Floyd, A. Romanow, "TCP Selective Acknowledgment Options," RFC 2018, October 1996.
- 6) W. Chen, J. Lee, "Some Mechanisms to Improve TCP/IP Performance over Wireless and Mobile Computing Environment," proceedings of ICPADS 2000
- 7) C. Perkins, "IP Mobility Support," RFC2002, October 1996
- 8) A. Bakre, B.R. Badrinath, "Handoff and Systems Support for Indirect TCP/IP," In Proceeding of the Second USENIX Symposium on Mobile and Location Independent Computing, April 1995.
- 9) K. Brown, S. Singh, "M-TCP: TCP for Mobile Cellular Networks," ACM Computer Communication Review Vol 27(5), 1997.
- 10) R. Atkinson, S. Kent, "Security Architecture for the Internet Protocol," RFC 2401, November 1998.
- 11) H. Balakrishnan, S. Seshan, E. Amir, R. Katz, "Improving TCP/IP Performance over Wireless Networks," Proc. 1st ACM Conf. on Mobile Computing and Networking (Mobicom), Berkeley, CA, November 1995.
- 12) K. Ratnam, I. Matta, "WTCP: An Efficient Transmission Control Protocol for Networks with Wireless Links," Technical Report NU-CCS-97-11, Northeastern University, July 1997.
- 13) V. Jacobson, R. Braden, D. Borman, "TCP Extensions for High Performance", RFC 1323, May 1992.

- 1) Hari Balakrishnan, Venkat Padmanabhan, Srinivasan Seshan, and Randy H. Katz, "A Comparison of Mechanisms for Improving TCP