■3 群（コンピュータネットワーク）－4 編（トランスポートサービス）

2 章 TCP (Transmission Control Protocol) の改善

■概要■

■3群 － 4編 － 2章

2-1 広帯域高遅延環境における TCP の課題と解決策

（執筆者：甲藤二郎）［2011 年 1 月 受領］

2-1-1 広帯域高遅延環境における TCP の問題点

従来のインターネットでは TCP-Reno（NewReno と SACK を含む）[1] が標準的なトランスポートプロトコルとして広く用いられてきたが，ネットワークのブロードバンド化の進展とともに，TCP-Reno では広帯域高遅延ネットワークの利用可能帯域を十分に利用することができない問題が顕在化してきた．これは，TCP-Reno の保守的な輻輳制御方式に起因しており，輻輳ウィンドウの増加幅と減少幅がネットワーク環境とは独立に固定されているため，広帯域高遅延環境に適応することができないためである．具体的には，まず，輻輳ウィンドウの増加幅がリンク速度に対してゆっくり過ぎるため，広帯域リンクの容量を満たすまでの時間が長くなる.また,パケット廃棄が発生すると輻輳ウィンドウを単純に半減させるため，広帯域リンクに空き帯域が発生したり，望ましい輻輳ウィンドウ値に回復させるまでの時間が遅くなったりする．結果として，TCP-Reno では広帯域リンクを有効活用することができず，スループットが低下することになる．

以上の問題を受けて，1990 年代後半より現在に至るまで，多種多様な TCP の改善方式が提案されてきた．これらは TCP Variants とも呼ばれ，大きく

・Loss-based TCP 制御方式
・Delay-based TCP 制御方式
・Loss-based と Delay-based を組み合わせた TCP 制御方式（Hybrid TCP 制御方式）

の三通りに分類することができる．以下，それぞれについて説明を加えていく [*1].

2-1-2 Loss-based TCP 制御方式

(1) AIMD 制御

Loss-based TCP 制御方式は，TCP-Reno と同様にパケット廃棄を輻輳指標とする方式である．TCP-Reno の輻輳制御方式（厳密には輻輳回避フェーズにおける輻輳ウィンドウ制御方式）は AIMD（Additive-Increase / Multiplicative-Decrease）と呼ばれ，パケット廃棄が発生するまでは RTT（ラウンドトリップ遅延）ごとに 1 パケット（加算的に）輻輳ウィンドウを増加させ，パケット廃棄が発生すると半分の値に（乗算的に）輻輳ウィンドウを減少させる．

AIMD 制御を一般化すると，ACK パケットを受信するたびに更新される輻輳ウィンドウの計算式は次式で表される．

$$\begin{cases} cwnd = cwnd + a / cwnd \\ cwnd = cwnd * (1 - b) \end{cases} \qquad (2・1)$$

ここで，変数 a は RTT 当たりの輻輳ウィンドウの増加量，変数 b はパケット廃棄検出時の輻輳ウィンドウの減少量を定めるもので，TCP-Reno の場合は $a = 1$，$b = 0.5$ となる．

図 2・1 には，TCP-Renoの輻輳ウィンドウの振舞いの例を示す．ここではまた，リンク帯域幅に相当する量をBDP（Bandwidth-Delay Product：厳密には帯域幅にRTTを乗じた値），ル

[*1] 以下では輻輳回避フェーズのみを対象とし，スロースタートについては言及しない．

ータのバッファサイズをbufferと表し，輻輳ウィンドウがbufferを満杯にするたびにパケット廃棄が発生し，輻輳ウィンドウを半減してはゆっくり回復させる動作を繰り返すことを示している．このとき，バッファサイズがBDP以上であればパケット廃棄後もリンクは有効利用される（空き帯域は発生しない）が，BDP未満の場合には図に示すような空き帯域（リンク帯域と輻輳ウィンドウに挟まれた三角形の領域）が発生し，リンクの使用効率（Efficiency）が低下する [*2].

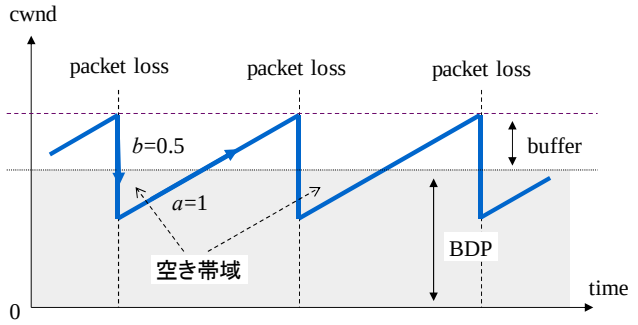


図 2・1　AIMD 制御による輻輳ウィンドウの振舞いの例（TCP-Reno の場合）

(2)　High-Speed TCP と Scalable TCP

広帯域高遅延環境における TCP-Reno の問題を緩和するためには，AIMD 制御の変数 *a*, *b* を広帯域環境に合わせて調整してやればよい. このような背景のもとに提案されているのが，High-Speed TCP（HS-TCP）[2] と Scalable TCP（STCP）[3] である.

HS-TCP では，リンク帯域幅と RTT に従って AIMD 制御の変数 *a*，*b* を適応的に定める変換式が導かれている. これによれば，例えば 10 Gbps のリンク速度で RTT が 100 ms の場合，$a = 70$, $b = 0.1$ が推奨値として与えられている. また STCP では，輻輳ウィンドウの増加幅が，ACK パケットを受信する度に $cwnd = cwnd + 0.01$ で更新され，また，輻輳ウィンドウの減少は $b = 0.125$ で定義されている．いずれの場合も，TCP-Reno に比して急速に輻輳ウィンドウを増加させ，パケット廃棄の検出時には，輻輳ウィンドウを半減させるのではなく，より緩やかに減少させ，所望の輻輳ウィンドウの値に回復する時間を短縮している．こうして，HS-TCP と STCP は，広帯域リンクの使用効率を高めることに成功している.

一方で，HS-TCP と STCP は共に TCP-Reno と共存させた場合には，TCP-Reno を追い出してしまう欠点が指摘されている．これは両者の輻輳制御がアグレッシブ過ぎるためであり，TCP-Reno との親和性（Inter-protocol Friendliness）が低いと言われる.

(3)　TCP Westwood

TCP-Westwood（TCPW）[4] は，もともとは無線環境のようなランダムエラーの多い通信環境を想定して提案されたものであるが，そこで検討された数多くのアイデアは，その後の多

[*2] リンクの使用効率を高く保つためにルータのバッファサイズを BDP に設定するように示唆する報告もあるが，バッファサイズが大きいほど遅延は増大し，メモリのコストも増大するために，単純に大きくすればよいと結論づけることはできない.

数の TCP Variants に影響を与えている．

TCPW では，まずパケット廃棄検出時の輻輳ウィンドウの減少を次式によって行う（厳密には *cwnd* を 1/2 倍したものと比較して，大きい方を選択する）．

$$cwnd = cwnd \cdot \frac{RTT_{\min}}{RTT} \qquad (2・2)$$

ここで，$RTT_{\min}$ は観測された *RTT* の最小値であり，パケットがバッファリングされていないときの *RTT* に対応する．この式が意味することは，バッファサイズが BDP より小さい場合でも，リンクに空き帯域を作ることなく，丁度バッファ内のパケットをクリアするように輻輳ウィンドウを減少させることである．

また, TCPW では, FSE（Fair Share Estimates）と呼ばれる変数を推定し, これに従って ssthresh（スロースタートと輻輳回避のモードを切り替える閾値）を設定している．FSE は，競合フローがない場合はリンク帯域そのものであり，競合フローがある場合は公平に使用可能な帯域（利用可能帯域）を表す．FSE の求め方として複数個の手法が提案されており，まず TCPW-BSE（Bandwidth Share Estimation）では，パケットペア（Packet Pair）の原理に従い，ボトルネックでパケット到着間隔が拡がることを利用し, ACK パケットの到着間隔から FSE の推定を行っている．また，TCPW-RE（Rate Estimation）では，複数個のパケットをまとめて送った際の送信レートの推定値から FSE の見積もりを行っている．BSE と RE には一長一短があり，BSE は推定値が大きめになりやすく，RE はパケット廃棄があると推定値が小さめになるという問題がある．そこで更に輻輳状態か否かを検出して適応的に BSE と RE を切り替える TCPW-ABSE（Adaptive Bandwidth Share Estimation）の提案も行われている．

TCPW 自体の輻輳ウィンドウの増加方法は TCP-Reno のそれと同じ（$a = 1$）であり，広帯域高遅延環境における TCP の特性改善を目的にしたものではない．しかしながら，輻輳ウィンドウの減少方法や FSE の推定は，広帯域高遅延環境における帯域適応にそのまま活用可能なものであり，後述する各種の Hybrid TCP 制御方式に引き継がれている．

(4) BIC-TCP と CUBIC-TCP

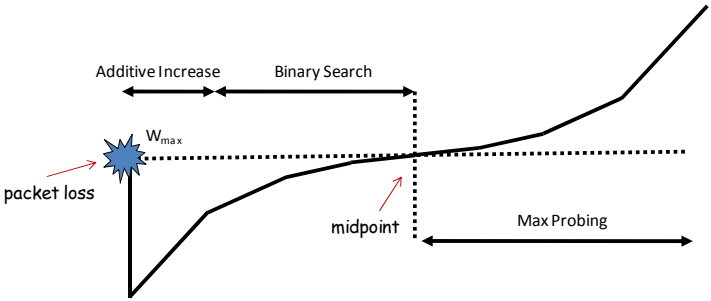
BIC-TCP（Binary Increase Congestion Control）[5] は，直前にパケット廃棄が発生した輻輳ウィンドウの値を平衡点とみなし，Additive Increase，Binary Search，Max Probing の三つのモードで次の平衡点（パケット廃棄が発生する点）を探索する動作を繰り返す輻輳制御方式である．

図 2･2(a)は，BIC-TCP の輻輳ウィンドウの挙動を示している．直前の動作でパケット廃棄が起きた輻輳ウィンドウの値を $W_{\max}$ とし，最初は Additive Increase で急速に輻輳ウィンドウを増加させ，平衡点（Midpoint）に近づくにつれて輻輳ウィンドウの増加幅を指数的に減少させる．そして，平衡点付近でパケット廃棄が起きない場合は，それまでとは逆の曲線で輻輳ウィンドウを増加させ，次のパケット廃棄を発生させる．この動作を繰り返し，輻輳制御を実現する．

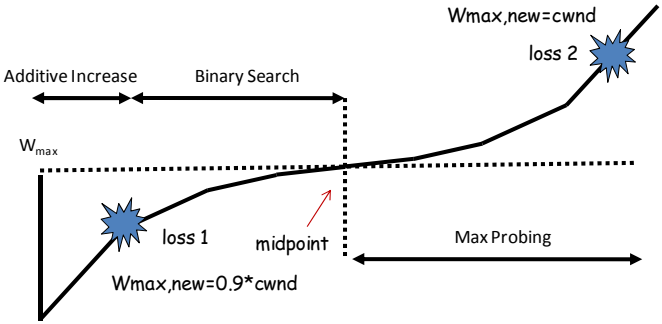
BIC-TCP の，平衡点に達するまでの輻輳ウィンドウの増加メカニズムを式で表せば，次式のようになる．

$$\begin{aligned} &\text{if } (cwnd > W_{\max})\ W_{\text{inc}} = (W_{\max} - cwnd)/2 \\ &\text{else } W_{\text{inc}} = (cwnd - W_{\text{inc}})/2 \\ &cwnd = cwnd + W_{\text{inc}}/cwnd \end{aligned} \qquad (2・3)$$

実際は輻輳ウィンドウの増加量（W_{inc}）に最大値を設けているため，線形の Additive Increase と指数的な Binary Search の二つのモードが存在することになる．



(a) 輻輳ウィンドウの増加



(b) パケット廃棄の発生と $W_{\max}$ の更新

図 2・2　BIC-TCP の輻輳ウィンドウの振舞い

また，図 2･2(b)は，BIC-TCP におけるパケット廃棄の発生と $W_{\max}$ の更新のメカニズムを示している．平衡点に達する前にパケット廃棄が発生した場合と達した後で発生した場合で $W_{\max}$ の更新方法が異なっており，前者の場合は $W_{\max} = cwnd * 0.9$ とし，後者の場合は $W_{\max} = cwnd$ とする．BIC-TCP の定常的な動作として，前者によって更新した場合，次のラウンドでは平衡点（$W_{\max}$）を超えてからパケット廃棄が発生し，後者によって更新した場合，次のラウンドでは平衡点に達する前にパケット廃棄が発生する．このため，BIC-TCP の輻輳ウィンドウの振舞いを観測すると，S 字型のカーブと凸型のカーブが交互に繰返し表れる様子が観察される．

CUBIC-TCP [6] は，図 2･2 に示した BIC-TCP の輻輳ウィンドウの振舞いを，三次関数で簡略化近似（Cubic Approximation）したものである．安定性に優れると言われ，また Hybrid 方

式と同様の，TCP-Reno との親和性を考慮したモードを備えている．また，パケット廃棄の発生間隔を関数近似に組み入れることで，RTT 公平性（RTT Fairness：RTT の異なるフロー間の公平性）も改善されるとの報告がある．

CUBIC-TCP は，TCP-Reno を置換え，Linux デフォルトのトランスポートプロトコルとなっている．一方で，バッファを積極的に埋め尽くす設計であるため，遅延が増加したり，他プロトコルが追い出される問題は指摘されている [7].

2-1-3 Delay-based TCP 制御方式

Delay-based TCP 制御方式は，パケット廃棄を輻輳の指標とする Loss-based TCP 制御方式とは異なり，RTT の増加を輻輳指標とし，パケット廃棄が起きる前に輻輳ウィンドウを小さくする輻輳制御方式である．**図 2･3** は，Delay-based TCP 制御による輻輳ウィンドウの振舞いを示す．図に示すように，送信パケットがリンク帯域を満たし，RTT の増加が観測されると，Delay-based TCP 制御はバッファ溢れが起きるまで輻輳ウィンドウを増加させるのではなく，バッファ内に一定数（α 個）のパケットを保持するように輻輳ウィンドウの調整を行う．

Delay-based TCP 制御では，理想的に振る舞う限りはパケット廃棄が発生せず，かつ低遅延伝送も可能になるため，Loss-based TCP 制御を凌ぐスループット効率の実現が示唆されている．一方で，Delay-based TCP 制御と Loss-based TCP 制御を共存させた場合には，Loss-based TCP 制御によって追い出される不公平性の問題が指摘されている．

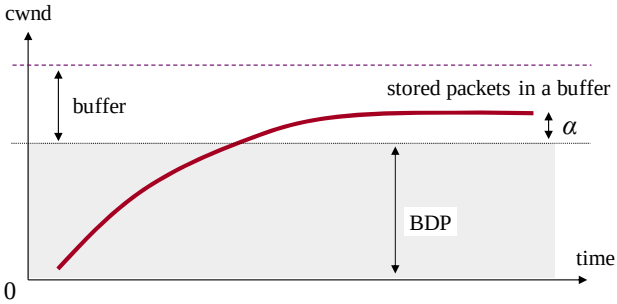


図 2・3　Delay-based TCP 制御における輻輳ウィンドウの振舞い

(1)　TCP-Vegas

TCP-Vegas [8] における輻輳制御は RTT ごとに次式によって実行される．

$$cwnd = \begin{cases} cwnd + 1 & diff < \alpha \\ cwnd & otherwise \\ cwnd - 1 & diff > \beta \end{cases} \qquad (2・4)$$

ただし，

$$diff = \left(\frac{cwnd}{RTT_{\min}} - \frac{cwnd}{RTT} \right) \cdot RTT_{\min} \qquad (2・5)$$

式(2･5)は，TCPW の式(2･2)を包含する形式で，バッファ内に保持されているパケット数を表している．そのうえで，式(2･4)は，バッファ内パケット数がα個とβ個の間に収まるように輻輳ウィンドウを増減するフィードバック制御を掛けている．αとβの典型値は$\alpha=1$，$\beta=3$とされることが多いが，$\alpha=\beta$としてもよい．

(2) FAST-TCP

TCP-Vegas は広帯域高遅延環境を想定して設計されたものではなく，輻輳ウィンドウの増加量は TCP-Reno のそれと同等である．これに対して FAST-TCP [9] は，Delay-based TCP 制御を広帯域高遅延環境に拡張したものであり，その輻輳制御は次式によって実行される．

$$cwnd = \min\left\{2\cdot cwnd,\ cwnd\cdot\frac{RTT_{\min}}{RTT}+\alpha\right\} \qquad (2\cdot 6)$$

式(2･6)において，右辺の第 1 項はスロースタートと同様に急速に増加する項であり，広帯域リンクを埋めるのに貢献する．一方，第 2 項は，リンク帯域を埋め切るパケット数とバッファ内に保持したいパケット数（α）の和を明示的に輻輳ウィンドウの設定値としている．

2-1-4 Loss-based と Delay-based を組み合わせた TCP 制御方式

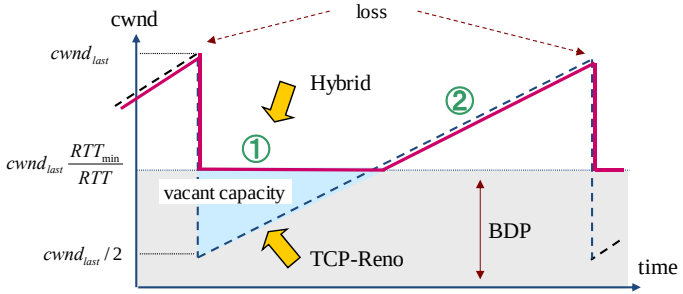
(1) Hybrid TCP 制御の動作原理

Loss-based と Delay-based を組み合わせた TCP 制御方式は Hybrid TCP 制御方式と呼ばれ，双方の課題を同時に解決する手法として注目されている．Loss-Based 方式では，リンクに空き帯域が生じる問題（効率性の問題）と，改善方式が既存の TCP-Reno を追い出してしまう問題（親和性の問題 I）がある．また，Delay-based 方式では，Loss-based 方式と共存させたときに追い出されてしまう問題（親和性の問題 II）がある．Hybrid TCP 制御方式では，効率性の問題に対しては Delay-based モードで動作させ，親和性の問題に対しては Loss-based（TCP-Reno）モードで動作させることで問題の解決を図る．二つのモードの切り替えには，主に RTT を指標とする．

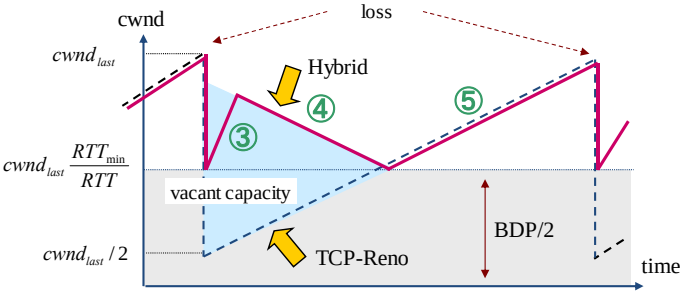
図 2･4(a)は，バッファサイズが BDP よりも小さい場合の，Hybrid TCP 単独，ならびに TCP-Reno 単独の場合の輻輳ウィンドウの理想的な挙動の違いを表している．まず, TCP-Reno の場合，パケット廃棄発生後に輻輳ウィンドウを半減させるため，リンクの空き帯域が発生する．これに対して，Hybrid TCP の場合，TCPW の式(2･2)に従って輻輳ウィンドウを減少させることで，リンクの空き帯域の発生を回避できる．また，Hybrid TCP は，内部に TCP-Reno の輻輳ウィンドウを保持し，それが BDP に到達するまでは Delay-based モードで動作し（図中の①），到達後に Loss-based（TCP-Reno）モードに移行する（図中の②）．ただし，②の動作は必須ではなく，RTT の増加が検出されない場合は Loss-based モードに移行しない実装も考えられる．

図 2･4(b)は，バッファサイズが BDP よりも小さい場合の，Hybrid TCP と TCP-Reno が互いに競合している場合の輻輳ウィンドウの理想的な挙動の違いを表している（2 フローが競合しているため, リンク帯域を BDP/2 に変更している）. パケット廃棄発生後, TCP-Reno, Hybrid TCP それぞれ輻輳ウィンドウを減少させるため，図中に三角形領域として示す空き帯域が発生する．次に，TCP-Reno は徐々に輻輳ウィンドウの回復を図るのに対して，Hybrid TCP は急速に輻輳ウィンドウを増加させ，空き帯域を埋め切ろうとする（図中の③）．その後，

TCP-Reno の輻輳ウィンドウの増加に適応して，Hybrid TCP は RTT が増大しないように徐々に輻輳ウィンドウを減少する（図中の④）．最終的に，TCP-Reno がリンク帯域を埋め切った後は，Hybrid TCP は TCP-Reno として動作し，RTT の増加が観測される区間では，TCP-Reno との親和性を確保する（図中の⑤）．以上をまとめると，③は広帯域高遅延環境に対応するための TCP Variant に共通に求められる動作，④は Delay-based モードの動作，⑤は Loss-based モードの動作，となる．



(a) Hybrid TCP 単独（及び TCP-Reno 単独）の場合



(b) Hybrid TCP と TCP-Reno が競合している場合

図 2・4　Hybrid TCP 制御における輻輳ウィンドウの理想的な振舞い

以上のように，Hybrid TCP は，リンクに空き帯域が存在する区間（RTT が増加しない区間）に Delay-based の利点を活かして効率性の問題を解決し，Loss-based とバッファを共有する区間（RTT が増加する区間）には Loss-based として動作することで親和性の問題を解決する．ただし，バッファサイズが BDP に比して十分に大きい場合は，Hybrid TCP は常に Loss-based で動作するため，効率性の利点は得られない．また，RTT が不規則に変動するようなネットワークでは適切なモードの移行が難しいのではないか，との指摘がある．

(2)　Hybrid TCP の実際

Hybrid TCP 制御の動作は以下のようにまとめられる．

A) 空き帯域が存在する区間は輻輳ウィンドウを急速に増加させる．

B) RTT が安定している区間（一定個数のパケットがバッファリングされている区間）は，その状態を保つように輻輳ウィンドウを微調整する．

C) RTT の増加が観測される区間は TCP-Reno として動作させる．

D) パケット廃棄が検出されると輻輳ウィンドウを減少させる．

これまで種々の Hybrid TCP 制御が提案されているが，動作 B)と動作 C)はほぼ同様であり，個別の手法の違いは主に動作 A)と動作 D)に表れる．

Hybrid TCP 制御の具体例として，Compound TCP（CTCP）[10]，Adaptive Reno（ARENO）[11]，YeAH-TCP [12]，TCP-Fusion [13] をとり上げ，それぞれの手法の違いを**表 2･1** にまとめる．ここで，B は推定帯域幅，$D_{\min}$ はタイマの解像度，N は推定フロー数，PS はパケットサイズである．輻輳ウィンドウの増加手段は，手法はすべて異なるものの，リンク帯域幅を埋め切るまで急激に増加させる意図では同様である．ARENO や TCP-Fusion のように帯域推定を併用すれば，理屈上は数回のラウンドで帯域幅を埋め切ることもできる．このため，長時間フロー（Long-lived Flow）の評価実験では，輻輳ウィンドウの増加方法の違いによる特性の差異は小さい．

表 2･1　Hybrid TCP 制御方式の具体例と分類

Hybrid TCPs	Window Increase	Window Decrease
CTCP	$0.125*cwnd^{0.75}$	1/2
ARENO	B/10 Mbps	1/2〜1
YeAH-TCP	STCP	1/2, RTT_{min}/RTT, 7/8
TCP-Fusion	$B*D_{min}/(N*PS)$	RTT_{min}/RTT

一方，輻輳ウィンドウの減少手段は，TCP-Reno と同様に半減する手法，TCPW と同様の式(2･2)で行う手法，独自の連続的な関数を定義して行う手法など多種多様である．高品質な伝送路で実験する限りは，手法の違いはさほど顕著ではない．しかし，意地悪な実験として，伝送路にランダムエラーを加えると，手法の違いが顕著になる．簡潔に言えば，固定的に半減させる手法は，誤り率の増大とともにスループットが急速に劣化する．これは，輻輳ウィンドウの半減を連続的に実行してしまうためである．一方，式(2･2)に従う手法は，ランダムエラーに耐性を有し，ある程度の誤り率まで高スループットを保つことができる．これは，TCPW としての利点である．ただし，複数フローの競合時に一部のフローがリンクを占有してしまう問題はあり，YeAH-TCP のように，時折空き帯域を設けてやる工夫は有効である．

Hybrid TCP 制御の代表格として，CTCP は既に Microsoft Windows に組み込まれており，コマンド設定でいつでも利用可能である．Linux に組込み済の CUBIC-TCP と同様に，長年に渡って使われてきた TCP-Reno を置き換えるプロトコルとして注目されている．一方，CTCP が CUBIC-TCP に追い出されてしまう問題が知られており [7]，TCP-Reno の置き換えを想定し，

CUBIC-TCP との親和性の確保が課題になると思われる.

■参考文献

1) M. Allman et al., “TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms,” RFC 2581, Apr. 1999.
2) S. Floyd, “Highspeed TCP for Large Congestion Window,” IETF RFC3649, Dec. 2003.
3) T. Kelly, “Scalable TCP: Improving Performance in High-speed Wide Area Networks,” PFLDnet 2003, Feb. 2003.
4) C. Casetti et al., “TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links,” ACM Mobicom 2001, Jul. 2001.
5) L. Xu et al., “Binary Increase Congestion Control for Fast, Long Distance Networks,” IEEE INFOCOM 2004, Mar. 2004.
6) I. Rhee and L. Xu, “CUBIC: A New TCP-Friendly High-speed TCP Variant,” PFLDnet 2005, Feb. 2005.
7) K. Munir et al., “Linux beats Windows! or the Worrying Evolution of TCP,” PFLDNet 2007, Feb. 2007.
8) L. S. Brakmo and L. L. Peterson, “TCP Vegas: End-to-End Congestion Avoidance on a Global Internet,” IEEE Journal on Selected Areas in Communications, vol.13, no.8, 1995.
9) C. Jin et al., “FAST TCP: Motivation, Architecture, Algorithms, Performance,” IEEE INFOCOM 2004, Mar. 2004.
10) K. Tan et al., “Compound TCP: A Scalable and TCP-Friendly Congestion Control for High-speed Networks,” PFLDnet 2006, Feb. 2006.
11) H. Shimonishi et al., “TCP-Adaptive Reno for Improving Efficiency-Friendliness Tradeoffs of TCP Congestion Control Algorithm,” PFLDnet 2006, Feb. 2006.
12) A. Baiocchi et al., “YeAH-TCP: Yet Another Highspeed TCP,” PFLDnet 2007, Feb. 2007.
13) K. Kaneko et al., “TCP-Fusion: A Hybrid Congestion Control Algorithm for High-speed Networks,” PFLDnet 2007, Feb. 2007.

■3群 － 4編 － 2章

2-2 無線環境における TCP の課題

（執筆者：村瀬　勉）[2013年6月 受領]

有線リンクを前提に発展してきた TCP 性能は，無線リンクでは，最適にはならないと言われている．ここでは，無線アクセス網に着目し，有線と比べたときの無線リンクの特性とその特性からくる TCP スループットの低下及び不公平性といった性能上の問題への対処アプローチについて述べる．特に IEEE 802.11（以下単に 802.11）に関しては詳しく述べる．

2-2-1 無線の特性と TCP への影響

(1) 無線リンクの特徴

無線通信においては，固定の場合と移動の場合がある．移動の場合には，固定の場合の特徴に加えて，基地局（AP）からの距離の増減に応じた電波の減衰と AP を切り替えるハンドオーバの特徴が加わる．まず，固定の場合について述べる．TCP に与える影響要因は，無線の電波特性に起因する要因，無線方式・アルゴリズムに起因する要因，半二重通信や低容量回線に起因する特性，などに分けられる．雑音や他の無線通信からの干渉に起因する受信電波の時間的な変動がリンク容量の時間的変化につながる．このリンク容量の変化は，ビットエラーレートの変動と言い換えることもできる．一般に AP と端末は，1 対多で接続され，また通信リンクは排他的に共用される．そのため，同時送信して通信が衝突すること（コリジョン）が起こらないようなアクセス制御が導入されており，携帯電話や IEEE 802.16（WiMAX）では，基地局から端末への通信機会割り当てによる衝突回避方式，802.11 では AP も含めて全端末がランダムアクセスによる衝突回避方式が行われる．802.11 の場合には，CSMA/CA と呼ばれるランダムアクセスを用い，衝突に対して決められた回数の再送を行う．この再送は，衝突の他にビットエラーなどで無線データフレームが正常に受信されない場合にも，起動される．このほか，半二重通信，比較的小さなバッファ容量，AP から端末への（ダウンリンク）通信と端末から AP への（アップリンク）通信との非対称性，などが，TCP 性能に影響を与える要因である．以下では，802.11 について，具体的に TCP への影響を述べる．

(2) 無線での TCP の性能

上述のように無線リンクでは，輻輳とフレーム（無線リンク，MAC レイヤでのデータ通信単位）欠落は独立に発生する場合もある．そのため，フレーム欠落に対して通信レートを低下させ，輻輳崩壊を回避するのは，無線の場合には不適当であり，スループットが低下しすぎることが課題とされている．ただし，TCP においてパケットロスが発生するような状況では，通信容量が低下しているという場合もあるため，通信レートを低下させることに全く意味がないわけではない．一般に，MAC フレーム欠落に対しては，リンクバイリンクで再送を行い，これの回復を試みる．802.11 では，デフォルトで 7 回の再送を行い，7 回目の再送が失敗すると MAC フレームを廃棄するため，遅延及びパケットロスが生じることになる．すなわち，パケットロスは，MAC フレームを通信できない状況が続いているために発生する．

802.11 の無線 LAN のインフラストラクチャモードでは，公平性の問題が生じる [1), 2)]．無線 LAN は，AP も端末も同じ送信機会をもっているため，一つの AP に対して，多数の端末があると，端末ごとの TCP スループットに不公平が生じる．これは上り（端末→AP）下り（AP→端末）の通信容量の非対称さに起因する問題であり，非対称な有線網でも起こり得るが，無線の場合には，半二重通信であることが，この現象を顕著にする．

移動無線においては，AP からの距離が大きくなるとともに，電波の減衰に伴う通信容量の減少が現れる．電波の減衰は，距離の 2 乗に比例するが，誤り訂正符号などを用いているため，MAC フレームロスレートは，ある距離のところから急激に大きくなる特性をもつ．このため，TCP のスループット特性も同様になる．ただし，レートアダプテーションと呼ばれる状況に応じて適当な伝送符号化（＝伝送レート）を使用する方式を用いる場合には，スループット特性も距離に応じてなだらかに変化することになる．802.11 では，送信機会が平等であるため，端末自身としては，高レートでフレームロスを生じるよりも，低レートを用いてフレームロスを軽減する方が，結果として高いスループットを得られる．一方で，AP が提供できる総スループットは，低レートのトラヒックが多いほど低下してしまう [3)]．

ハンドオーバでは，AP の切り替えにともなって通信経路が大きく変わる可能性をもつため，TCP にとっては，パケットの順序逆転のみならず使用可能帯域の変化や競合トラヒックの状況といった通信環境が不連続的に大きく変化する [4)] ことを想定する必要がある．また，このような問題は，アドホックネットワークやメッシュネットワークのようなマルチホップネットワークにおいては，ルートの頻繁な変更や上り下りの干渉などの要因により，更に深刻になる．

2-2-2 課題へのアプローチ

上記課題に対する解決策に，いくつかのアプローチがある [5), 6)]．まずは，パケットロスによるスループット低下に対処するためのアプローチである．これをカテゴリに分けて考えると，まず(1)FEC などでエラーを隠蔽するアプローチ，(2)無線特有の問題に対処する特殊 TCP を無線区間のみ使用するアプローチ，(3)エンドエンドの TCP として，無線特有の問題にも有線の問題にも対処可能な特殊 TCP を使用するアプローチ，の三つに分けられる．このうち，(3)は，更に，細分化でき [5)]，(a)輻輳と無線エラーを区別する TCP，(b)無線を意識して TCP ステートを制御する TCP，(c)制御のタイミングをずらす TCP，などである．

(1)に関しては，既に 802.11 で行われている前述の MAC フレームの再送，トランスポートレイヤにて FEC を使用する方法，無線リンクに対する TCP レイヤでの再送に相当する TCP Snooping（**図 2･5**）[6)]，などがある．MAC フレームの再送では，ロスを軽減できる代わりに，遅延が大きくなってしまうという課題がある．FEC を用いる場合には，冗長による帯域のムダが課題となる．これに対して，状況に応じて FEC の強弱を制御する方法 [7)] などの対策がある．TCP Snooping については，RTT の大きさに応じた分量のバッファをもち，RTT とデータの対応付けを行うといった特殊機能を AP や AP 近辺のノードに配備する必要があるため，コスト増が課題である．いずれにしても，想定外の長いバーストエラー発生のような場合には，対処が難しく，ある程度安定した無線リンクへの適応となるであろう．(2)に関しては，Indirect TCP, SelectiveRepeatProtocol, Mobile TCP など多くのプロトコルが研究されている [5)]．これらは，TCP プロキシ機能（**図 2･6**）[6)] を AP や AP 近辺のノードに配備する必要があるた

め，ハンドオーバへの対処が難しく，コスト増も課題である．(3)に関しても，TCP Westwood をはじめとする多くの TCP[8), 9)] が研究されており，性能を競い合っている．それらには，ルータからの明示的な輻輳状況通知（Explicit Congestion Notification）を要求するものや送信側のみならず受信側 TCP の改造も要するものやデータ送出レートを調整するものなどがあり，追加機能の実現容易性や実現コストなどが大きく違っている．また，前述のように，それぞれの TCP は得意とする無線課題をもつため，どれが一概に良いか判断するのは，非常に困難である．また，無線とは無縁の端末の TCP をも変更しなければならないため，広く浸透するにはある程度の時間を要するであろう．

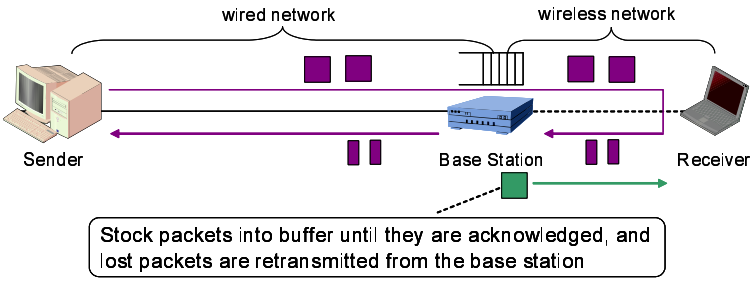


図 2・5　TCP コネクションの Snooping[6)]

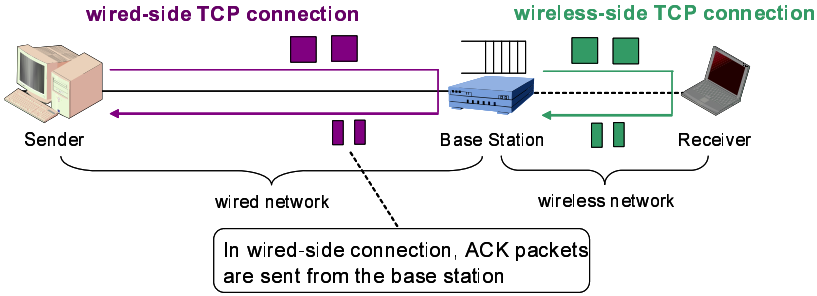


図 2・6　TCP コネクションのプロキシでの分割[6)]

次に，公平性の課題についてのアプローチについて述べる．一番目の公平性の問題は，TCP のデータの廃棄率がほとんどなく，なおかつ ACK の廃棄率が非常に大きい場合に発生する問題である．上りの TCP フローのスループットにおいて，いくつかのフロー（不幸なフロー）のスループットがほぼゼロになり，それ以外のフローは高スループットを得る．これは，TCP が累積 ACK を用いているために生じる現象である．この不幸なフローは，初期状態においてランダムに決定される CSMA/CA の待ちウィンドウサイズや電波状況といった偶然の要素で決まることが多い．この不公平は，有線側にも無線側にも TCP のデータパケットのロスが無い場合には顕著に現れる．ただし，TCP データパケットにロスがあることを前提とすると，ロスイベントが起こるたびに，不幸なフローになるフローが次々に切り替わるため，長期にわたって不幸なフローが生じることはない（**図 2・7**）[10)]．したがって，実際に通信のように

パケットロスがある程度生じることが自然である場合，あまり深刻な問題ではなく，文献 11) で示されているような簡単な制御を行うだけでほぼ解決できる．

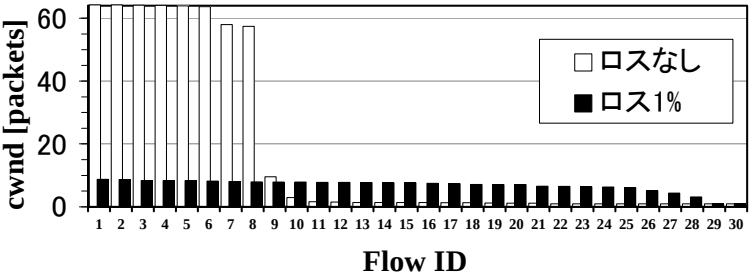


図 2・7　フレームロスレートと TCP スループットの不公平性 [10)]

公平性に関する二番目の問題は，上り TCP が下り TCP よりも大きいスループットを得る，という問題である．AP のバッファが溢れるとき，上り TCP フローは ACK を下り TCP フローはデータを失う．この場合，データを失う方が TCP の輻輳制御に与えるインパクトが大きいので，下り TCP フローが不利になる．この問題には，AP において，データを優遇することで解決できる [12)]．ただし，データの優遇度合いに関する検討が課題として残り，また優遇するための優先制御のコストも課題となるため，端末側（TCP 受信側）にて，ACK を制御する方法 [13)] などが提案されている．

最後に，これまで述べてきた対策の実装方法に関する技術について述べる．無線環境における TCP の性能向上には，クロスレイヤ制御が有効であるとされている．クロスレイヤ制御は，従来それぞれのレイヤで独立に行われてきた制御をレイヤ同士が協調することで，効率的に性能向上を行う制御である．例えば，MAC レイヤでのコリジョン確率や伝送レートといった情報をトランスポートレイヤが考慮して，TCP の輻輳制御を行うクロスレイヤ制御においては，動的に変動するチャンネルの状況に合わせた最適な TCP レート制御が可能となる．クロスレイヤ制御は，コストパフォーマンスを劇的に向上させる可能性をもち合わせる一方で，レイヤごとの独立性を壊してしまうため，開発コストや実装コストの増大に繋がるという一面ももっている．

2-2-3 TCP 特性と諸技術の関連

ここでは，上記で述べなかった技術と TCP 性能の関係について述べる．無線リンクレイヤでの QoS 制御と TCP，AP のバッファ量と TCP 性能，衛星や携帯電話で用いられる TCP，消費電力と TCP 制御などを紹介する．

802.11 においては，802.11e という CSMA/CA レベルの優先制御（EDCA）が提案され，市販機器にも搭載されつつある．また，EDCA 以外の CSMA/CA レベルの優先制御も多数提案されている．各優先制御は，VoIP（Voice over IP）などのリアルタイムアプリケーションや QoS 要求の厳しい通信を優先するために使用されるほかに，TCP に対しては，優先したい TCP フローへのスループット増加や，逆に TCP フローの公平性向上制御といった用途 [14), 15)] に使用されている．

AP のバッファ量は，公平性に大きく影響するパラメータでありながら，実際のバッファ量がどの程度 [16] かはあまり知られていない. TCP 上りデータに対して, AP バッファ量（ACK パケットのためのバッファ量）を増加させると，TCP のスループットは，ACK ロスに起因するレート低下が解消されるという要因で，最初増加していくが，後にはキュー長が長くなることに起因する ACK の RTT が大きくなる要因が強くなることで減少していく．また，バッファ量が大きいほど公平性は高くなる．

無線リンクとして 802.11 とは異なる無線に，衛星通信や携帯電話でのパケット通信などがある．衛星では，エラーが多くなおかつ遅延も大きい通信路を有効活用する TCP [17] などが提案されている．携帯電話では，携帯各社でそれぞれの TCP が使用されており，日本では，Wireless-profile TCP [18] が用いられている．

移動通信などでは，バッテリーの節約も重要である．通信性能とエネルギー消費は正の相関があるため，消費削減には，TCP の性能低下が避けらないという問題がある．この問題に取り組んだ研究として，文献 19) がある．

■参考文献

1) D. J. Leith and P. Clifford, “Using the 802.11e EDCF to Achieve TCP Upload Fairness over WLAN Links,” WiOpt, Apr. 2005.
2) S. Pilosof, R. Ramjee, Y. Shavitt, and P. Sinha, “Understanding TCP fairness sover Wireless LAN,” IEEE INFOCOM 2003, 13 Apr. 2003.
3) M. Abusubaih, J. Gross, S. Wiethoelter, and A. Wolisz, “On Access Point Selection in IEEE 802.11 Wireless Local Area Networks,” 31st IEEE Conference on Local Computer Networks, Nov. 2006.
4) K. Tsukamoto, S. Kashihara, and Y. Oie, “A Unified Handover Management Scheme Based on Frame Retransmissions for TCP over WLANs,” IEICE TRANSACTIONS on Communications vol.E91-B no.4, pp.1034-1046, Apr. 2008.
5) Ka-C. Leung and V. O. K. Li, “Transmission control protocol (TCP) in wireless networks: issues, approaches, and challenges,” IEEE Communications Surveys & Tutorials, vol.8, no.4. (4th Quarter 2006), pp.64-79, 2006.
6) 長谷川剛, 村田正幸, “無線ネットワーク環境に適したトランスポート層プロトコル,” 電子情報通信学会技術研究報告(CQ2009-20), pp.29-34, Jul. 2009.
7) T. Tsugawa, N. Fujita, T. Hama, H. Shimonishi, and T. Murase, “TCP-AFEC: An Adaptive FEC Code Control for End-to-End Bandwidth Guarantee,” 16th International Packet Video Workshop (PV 2007), Nov. 2007.
8) C. Casetti, M. Gerla, S. Mascolo, M. Y. Sansadidi, and R. Wang, “TCP Westwood: End-to-End Congestion Control for Wired/Wireless Networks,” Wireless Networks Journal 8, 467-479, 2002.
9) H. Shimonishi, T. Hama, and T. Murase, “TCP-Adaptive Reno for Improving Efficiency-Friendliness Tradeoffs of TCP Congestion Control Algorithm,” Fourth International Workshop on Protocols for Fast Long-distance Networks (PFLDNeT) Feb. 2006.
10) 内藤成文, 小畑博靖, 村瀬　勉, 石田賢治, “無線 LAN 環境における TCP 制御と MAC 制御を共用したフローQoS 保証技術の性能評価,” 電子情報通信学会第 8 回 QoS ワークショップ, QW8-P-07, Nov. 2010.
11) Y. Hirano and T. Murase, “Uplink TCP traffic control with monitoring downlink buffer for throughput fairness over wireless LANs,” IEEE PIMRC 2009, 737-741, Sep. 2009.
12) S. Shioda, H. Iijima, T. Nakamura, S. Sakata, Y. Hirano, and T. Murase, “ACK pushout to achieve TCP fairness under the existence of bandwidth asymmetry,” ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (ACM MSWiM 2010), ACM PM2HW2N, 2010.
13) M. Hashimoto, G. Hasegawa, and M. Murata, “A Transport-layer Solution for Alleviating TCP Unfairness in a Wireless LAN Environment,” IEICE TRANSACTIONS on Communications, vol.E94-B, no.3, pp.765-776, Mar. 2011.

14) 内藤成文, 小畑博靖, 村瀬　勉, 石田賢治, “無線 LAN 環境における TCP 制御と MAC 制御を共に用いたフローQoS 保証,” 電子情報通信学会和文論文誌 B, vol.J94-B, no.2, Feb. 2011.
15) K. Mimura, H. Obata, T. Murase, and K. Ishida, “Restraining Greedy TCP Behavior by MAC Frame Control on Wireless LAN,” IEEE 1st International Workshop on Emerging Densely Connected Networks, pp.1061-1065, Jan. 2011.
16) R. Ando, T. Murase, and M. Oguchi, “Characteristics of QoS-Guaranteed TCP on Real Mobile Terminal in Wireless LAN,” to appear in IEEE CQR 2011 Workshop, May 2011.
17) H. Obata, K. Ishida, S. Takeuchi, and S. Hanasaki, “TCP-STAR: TCP Congestion Control Method for Satellite Internet,” IEICE Transactions on Communications, vol.E89-B, no.6, pp.1766-1773, Jun. 2006.
18) H. Inamura, G. Montenegro, R. Ludwig, A. Gurtov, and F. Khafizov, “TCP over 2.5G and 3G Wireless Networks,” IETF RFC 3481, Feb. 2003.
19) 橋本匡史, 長谷川剛, 村田正幸, “無線 LAN 環境における TCP の動作を考慮した消費電力モデルの提案,” 電子情報通信学会技術研究報告, vol.110, no.339, NS2010-105, pp.1-6, Dec. 2010.