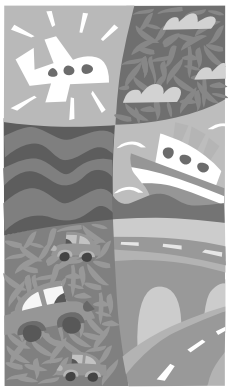

モバイルは今

無線データ通信環境とTCP/IPの進歩

楯岡 孝道
電気通信大学 tate@cs.uec.ac.jp

近年の，高速な無線LANや携帯電話による高速データ通信を例に挙げるまでもなく，無線データ通信環境の進歩は目覚しい．PHSのインフラを利用する定額制の広域無線データ通信回線が近年普及し始めたが，それによる移動中の安価な常時接続も現実味を帯びてきている．

このような無線データ通信回線経由のインターネット接続の通信速度が存外遅くて失望したこともあろう．このような回線の多くは最善努力（ベストエフォート）方式であり，パケット交換や通信路の空間的／時間的分割を行うことで安価で高性能なサービスを実現している．そのためカタログ値ほどの速度が出ないのにもやむを得ない面はある．一方で，従来のTCP/IPでは回線性能が十分引き出せないことがある．もうTCPは枯れた技術で，改良することなどない技術だと思われるかもしれないが，計算機環境や通信技術の変化にともないTCPも改良が続いている．今回は，無線データ通信環境の問題点とTCPの改善技術を紹介する．

●

TCPはインターネットの基礎となる通信プロトコルであり，さまざまな特性を持つ通信路上に，信頼性のある仮想回線を構築する．Webにアクセスするための HTTPや，メール転送のためのSMTPも，すべてこのTCPを利用している．途中の通信路でデータ消失や到着順序の変化があっても，TCPがこれを検出し再送することで，アプリケーションは意識することなく正しいデータを受信できる．

TCPでは送信側でデータを適当な大きさのパケットに分割し，シーケンス番号をつけて送信する．受信側ではどのシーケンス番号まで正常に受信したかを確認応答（ACK）として返信し，送信側でこれを確認することで信頼性のある通信を実現する．送信時にウィンドウサイズという量まではACKを待たない送信を許すことで，応答遅延が大きい場合も効率よく送信可能となる．通信エラー等が生じると受信側のシーケンス番号に抜けが生じ，その後はパケットを受信するたびに正常受信した最後のシーケンス番号を重複ACKとして返信する．送信側は再送タイマがタイムアウトするか，重複ACKが一定数以上到着すると，パケットが失なわれたと判断して再送する．再送タイマの値はデータ送出からACK受信までの時間，すなわちパケットの往復時間から動的に算出される．この機構により，TCPはさまざまな遅延を持つ通信回線への動的適応が可能となる．

ところが無線データ通信の場合，ノイズや障害物の影響を受けやすく，一般的な回線よりもパケット消失率が高い．また，中継等により伝送遅延も極端に大きくなる傾向がある．このような回線でTCPを用いると，再送効率の悪さから転送速度が急激に低下することがある．これは，TCPのACKが「どのシーケンス番号まで届いたか」だけを通知するため，複数パケット消失時には不要なパケットまで再送してしまったり，逆に必要なパケットの再送が遅れたりするためである．

このような問題に対するTCPの改良として，到着済みシーケンス番号を詳細に通知するSACK（Selective Acknowledgement）オプションがRFC2018として標準化されている．**図-1**の（a）に通常のACKの，（b）にSACKを用いたデータ転送の様子を示す．どちらの場合もシーケンス2000-2999と3000-3999の2パケットが消失し，受信側に4000-4999のパケットが到着した際に，2回目の1999まで到着済みとのACKを返信している．通常のACKでは送信側はシーケンス2000以降が再送対象としか判断できない．しかし，SACKでは送信側でも2000から3999までが再送対象だと明確に判断でき，再送は必要最低限で済む．TCPの通信開始時に両端でSACKの利用を交渉し，利用不能ならば従来通りの単

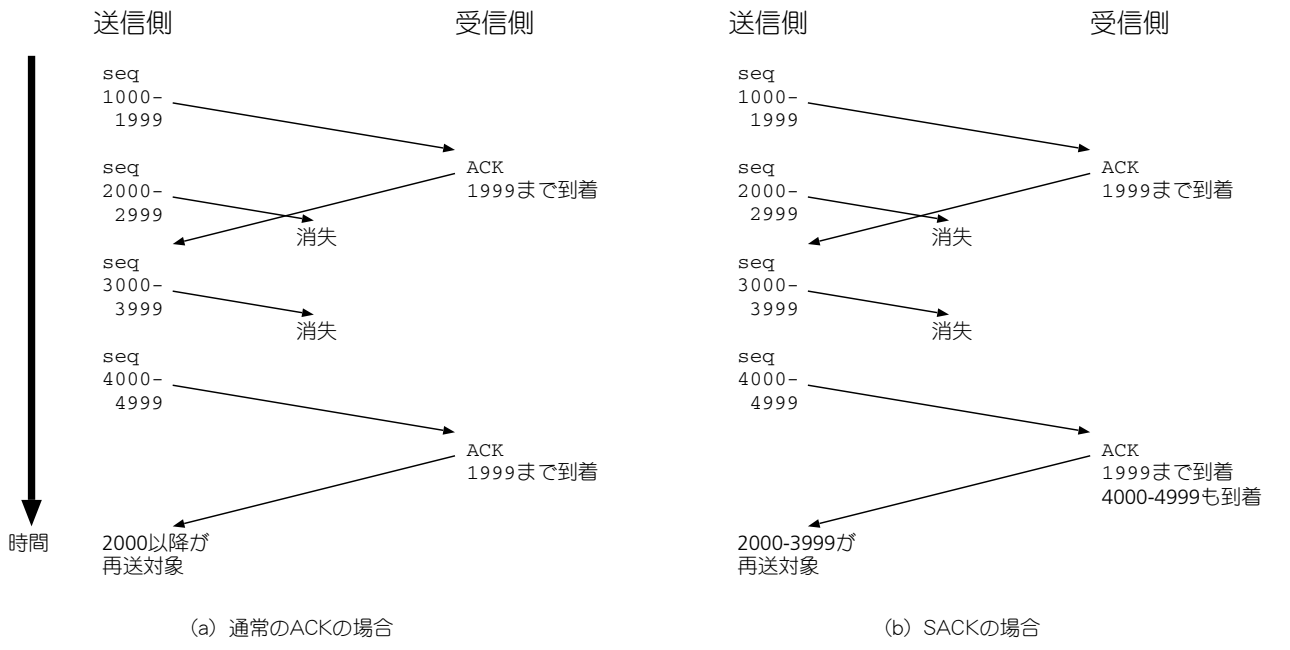

図-1 TCPのデータ転送例

純なACKを用いることで，SACKは従来の規格との相互運用性も保っている．

SACKのほかにも，再送アルゴリム，ウィンドウサイズの拡大，タイムスタンプオプションなど，TCPにはさまざまな改良が行われている．PMTUD（Path MTU Discovery）もこのような改良の1つである．

IPパケットが回線を通過する際，パケット長がその回線で許されている長さ（MTU, Maximum Transmission Unit）を超えていると，それに納まるように断片化（fragmentation）される．たとえば一般的なイーサネットではMTUは1500byteだが，その上にVPN（Virtual Private Network）を構築した場合，VPNのヘッダが加わるため1480byte以下になる．断片化されたパケットは，終点に届くと元のパケットに再構成される．しかし，1つでも通信路上で消失すると，再構成できずにすべてを再送する必要が生じる．そのため，特にパケット消失率の高い回線ではパケット長を小さくして断片化を避ける必要がある．一方で，パケット長が大きいほどパケット化のオーバーヘッドは小さい．したがって，通信の両端間で許される最大パケット長で送信することが望ましい．このために提案されたのがPMTUDである．PMTUDでは送信パケットに断片化を禁止するDF（Don't Fragment）ビットを設定する．もし途中の回線で断片化が必要になると，送信元にはICMP（Internet Control Message Protocol）によって通過可能なパケット長が通知される．通過する回線すべてで断片化が不要になったとき，送信元は最適なパケット長による送信が可能になる．

PMTUDは1990年に提案されており，SACKよりも長い歴史を持っている．また，その有用性から次世代IPであるIP version 6でも標準規格として採用されている．しかし実際にはPMTUDが有効に動作しないケースがある．RFC2923では，その原因の1つとして，不用意にICMPパケットの通過を禁止しているファイヤウォールを挙げている．ファイヤウォールにおいて，セキュリティ向上のために不要なパケットの通過を禁止するのは一般的だが，必要なパケットまで禁止してしまうことで，通信障害を引き起こしているわけである．近年のTCP実装では，このような場合でもタイムアウトによってPMTUDを中止して通信を継続するため，一見正常に通信できているように見えてしまう．しかし実はファイヤウォールの設定を修正すれば，より効率のよい通信が可能なのである．これは管理者がTCPの進歩に追従できていない状況といえるだろう．

●

インターネットは社会の重要なインフラとなったが，だからといって技術的発展が止まったわけではなく，相互接続性を保ちながらも発展し続けている．

あなたの使っているTCP/IPは，回線を無駄使いしていないだろうか？ また，あなたの組織のファイヤウォールは正しく設定されているだろうか？ SACKは現在ほとんどのOSで有効になっているが，RFC3481等に示された他の項目と合わせて確認してみるとよいだろう．せっかくのインターネットである，進化に追従して効率よく使いたいものである．

（平成15年3月20日受付）