

柔軟性：
ハイブリッドクラウド戦略が実現？

@IT > クラウド > Windows Server Insider > 第16回 信頼性のある通信を実現するTCPプロトコル...

基礎から学ぶWindowsネットワーク

第16回 信頼性のある通信を実現するTCPプロトコル（3）

（2/4 ページ）

2004年02月13日 00時00分 公開

[デジタルアドバンテージ, 著]

印刷

通知

見る

Share

8

前のページへ

1

2

3

4

次のページへ

TCPオプションとは？

TCP接続のオープン処理では、その接続（コネクション）に関する特別な設定を指定することができる。これを「オプション」といい、通常はTCP接続の送信側が受信側に対して提示するパラメータである。TCP接続は双方向なので、オプション指定は、双方から送信されることになる（それぞれから送信されるオプション設定が異なっているてもよい）。

TCP接続のオプション指定は、「オプション」という名前のとおり、場合によっては存在しない場合もあるし、通信相手とのネゴシエーション（折衝）の結果によっては受け付けられないこともある。また、オプション指定はTCPの接続ごとに個別に設定されるものなので、同じマシン同士でTCP接続を確立しても、常に以前と同じオプション設定が使われるというものでもない（だがたいていの場合は、あるマシンから送信されるTCPオプションはほとんど常に同じである）。

TCPオプションの種類と構造

TCPのオプションは、TCPのヘッダ中にある「オプション」フィールドを使って送信される（フィールドの構造については前回の「連載第15回—1. TCPパケットの構造」の冒頭の図を参照）。

オプションは、常に32bit（4bytes）幅になるようにしてTCPヘッダ中に埋め込まれる（TCPヘッダは32bit単位で可変長になっているため）。具体的なオプションとしては、以下のようなものがあるが、ここでは、代表的なオプションについて、その構造と意味を解説しておこう。

種別	意味
0	オプションの終了。TCPオプション・リストの終了を表す
1	NO-OP（NO-OPERation）。特別な意味を持たず、32bit単位にそろえるために1〜3個利用される
2	MSS（最大セグメント・サイズ）。受信可能な最大セグメント・サイズを表す
3	ウィンドウ・スケール。64Kbytesよりも大きなウィンドウ・サイズを表すために利用される。RFC1323で定義されている
4	選択的受信確認（Selective Acknowledgment＝SACKという）。受信確認の応答アルゴリズムを最適化するために利用される。連続した領域でなく、部分的にACKを返すことにより、無駄な再送を防ぎ、転送効率を向上させることができる。RFC2018で定義されている
5	TCPタイムスタンプ。パケット中にタイムスタンプ（送信時間）情報を埋め込むために利用される。タイムスタンプ情報からパケットの往復時間（Round Trip Time）を計測し、TCPの再送タイマ・アルゴリズムの最適化を行う

オプションの種類とその意味

オプション種別は8bitの数値で表す。現在のIPv4ではこれらのオプションが定義されているが、3以降は初期のTCP規格には含まれていない拡張仕様である。そのため、これらのオプションを使用しても（送信しても）、受信側がそれを了承するかどうかは分からない。

「オプション終了」と「NO-OP」オプション

検索

ホワイトペーパー

- 

産業用イーサネットと直面する10の問題:工場の稼働停止を防ぐためのヒント
- 

「スイッチ」と「ブリッジ」の違いとは? LANを理解するための基礎知識
- 


検知してからどうするか!? 標的型サイバー攻撃における内部対策の提案
- 


もう「Wi-Fi 7」時代? 無線LANの気になる進化

スポンサーからのお知らせ

- 重要なのは発展性 なぜ今、“ストレージ”に注目が集まっているのか
- 「ネットワークが分からない」状態からでも丸ごとサポート

Special

- 

複数ベンダーの「継ぎはぎSASE」で生じる課題、どうすれば解決できるのか？
- 

「守る」だけでは不十分 今どきのストレージには何が必要？
- 「ほとんど誰も見ていない」社内ポータル、どう変えるべき？ **New!**

オプション終了は、オプション・リストが終了することを表す。TCPオプションは、1つのTCPヘッダ中に連続して複数並べることであり、同時に何個も利用することができる。そのオプション・リストが終了して、もう後続のオプションが存在しないことを表すのが、このオプション終了である。ただし、わざわざこのオプションを使わなくても、オプション部分の長さはTCPのヘッダ長フィールドから計算することができるので、使われることはない。

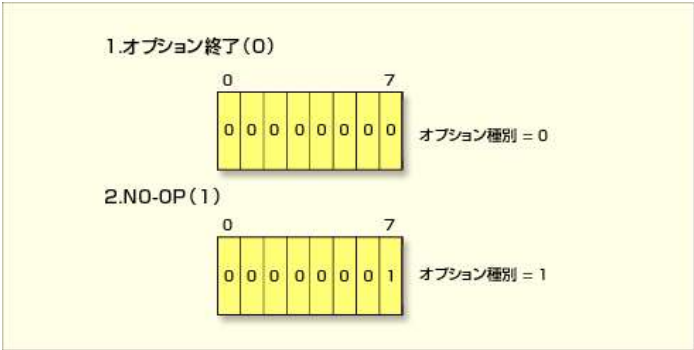


Special

- PR -

オンプレのITインフラを「サブスク」で利用できるサービスは
何がスゴイのか？

具体的なオプションのビット・パターンを以下に示す。

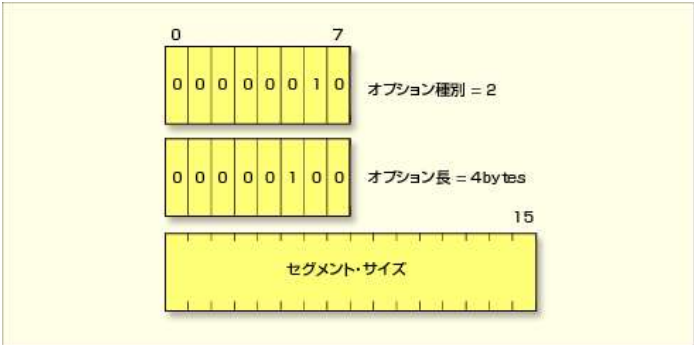


「オプション終了」と「NO-OP」オプションの構造
いずれも1byte長のオプションであり、それぞれ整数で0と1というオプション・コードを持っている。実際にはNO-OPオプションの方だけが利用される。

NO-OP（No Operation）は、何もしないという意味のオプションである。実際には、オプション・フィールドの総サイズがちょうど32bit（4bytes）の倍数になるように、必要なだけ（1〜3個）、このNO-OPオプションを詰め込む。

「MSS（最大セグメント・サイズ）」オプション

MSS（Maximum Segment Size. 最大セグメント・サイズ）は、受信可能なセグメントの最大サイズを通信相手に通知するためのオプションである。具体的には、次のような4bytes長のオプションであり、現実のTCPのオープン時（SYNパケットの送信時）において、ほとんど必ず使用されているTCPオプションである。



「MSS」オプションの構造
MSSは最大セグメント・サイズを相手に通知するためのオプションである。オプション長は、このMSSオプション全体の長さ（4bytes）を表している。セグメント・サイズには16bitの数値がセットされる。



自分が作ったアプリがスマホで動くさまを見ると、学生の目が輝きます **New!**



オンプレのITインフラを「サブスク」で利用できるサービスは何がスゴイのか？



「ネットワークが分からない」状態からでも丸ごとサポート **New!**



データは「守りながら活用する時代」に



社内ルールだけでは限界 有名無実化した「ローカル保存禁止」にどう対応？



NTTデータと日本IBMがタッグ！AIは仕事をどう変える？

[@IT Specialへ](#)

Windows Server Insider 記事ランキング

本日	月間
----	----

Excel（エクセル）で日付から自動的に曜日を入力する

【Excel】重複データを色付けして瞬時にダブりをチェックする

【Excel】パスワードロックを強制的に解除する方法

TCP/IP通信の状態を調べる「netstat」コマンドを使いこなす【Windows OS】

Windows OSのdirコマンドでファイル名の一覧を取得する

システム要件を満たさないPCをWindows 11 2023 Update（23H2）にアップデートする方法

【Windows 10／11】えっ、UTF-8じゃなくてShift-JISで？ お手軽文字コード変換方法まとめ

PDFファイルにキーボードから直接文字入力する方法【本家Acrobat Reader編】

Excelの落とし穴「先頭のゼロ（0）」問題の対処法

【Windows 10／11】PCが数分で勝手にスリープするのを防ぐ

[ランキングをもっと見る](#)

あなたにおすすめの記事

- PR -



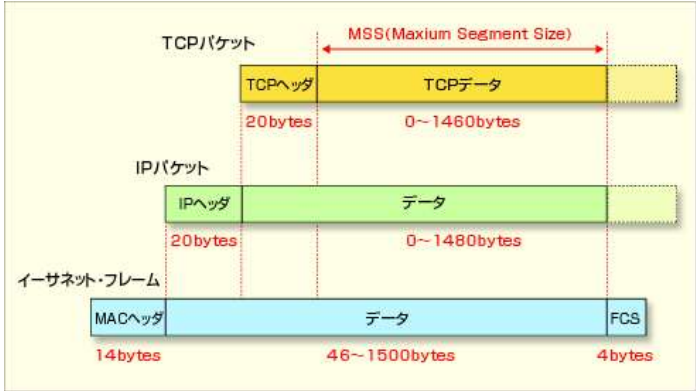
社内ルールだけでは限界 有名無実化した「ローカル保存禁止」にどう対応？



支笏湖の鏡面現象を予測せよ！ ーコード×機械学習で地域課題を解決する挑戦 **New!**

「セグメント」とは、TCPにおけるデータの送信単位である。TCPは、IPパケット上に構築されているプロトコルであるため、1つのTCPパケットで最大64Kbytesまでのデータ（ただしウィンドウ・サイズよりも小さいサイズ）を一度に送信することが可能である。だが実際には、TCPではセグメントというサイズごとに区切ってデータを送信している。そして、その許容される最大のセグメント・サイズを「MSS（最大セグメント・サイズ）」という。

TCPとIP、そしてイーサネットのパケット長の関係を図にすると、次のようになる。



TCPパケットと最大セグメント・サイズ
1つのTCPパケットで、IPフラグメントを発生させずに送信可能なデータ量の最大サイズをMSS（Maximum Segment Size. 最大セグメント・サイズ）という。例えばイーサネットでは、最大1500bytesのデータを送信できるので（ローカルのLAN上のネットワークを想定。PPPoEなどの場合はもっと小さくなる）、ここからIPヘッダの最小サイズ20bytesとTCPヘッダの最小サイズ20bytesを減算すると、MSSは1460bytesとなる。もしIPルーティング経路の途中に、さらに小さなMTU（最大転送サイズ）のネットワークが存在すると、MSSはもっと小さくしなければならない。

すでに述べてきたように、IPパケットでは最大64Kbytes（ただしIPヘッダの部分は除く）までのパケットを送信することができる。だが、その下にある物理的なネットワークの制約によって、実際にはもっと小さなサイズのパケットしか送信することができない。

例えばイーサネットでは、一度に送信できるデータは最大1500bytes（最小は46bytes）までとなっている。IPヘッダの標準的なサイズは20bytesなので、IPパケットで送信するデータのサイズが1480bytesを超えると、イーサネット上ではIPフラグメンテーションが必要になる（「連載第10回—IPパケットの構造とIPフラグメンテーション」参照）。

IPフラグメンテーションが発生すると、IPパケットのフラグメンテーション（分割）と再構成が必要となり、若干のオーバーヘッドが生じる。このIPパケットの分割と再構成に似た操作は、TCP層におけるウィンドウ制御でも行われるため、2つの層で行うと効率がよくない。またIP層における分割と再構成は、すべてのIPフラグメント（分割された各IPパケットの断片）がそろわないと完成しないが、TCP層におけるウィンドウ制御では、部分的なTCPパケットであっても正常な受信と見なすことができ、処理を先へ進めることができる。TCPではウィンドウの先頭の方のパケットから順番にまとめて受信確認を返信するため、送信されたTCPパケットがすべてそろわなくても、先頭からいくつかが続いていればそれでよいからである。

以上のようなIP層とTCP層の特性を考えると、なるべくIP層でのフラグメントを避け、TCP層で処理した方が望ましいことが分かるだろう。例えば2Kbytesのデータを送信したい場合、1つのTCPパケットで送信するか（この場合はIPフラグメンテーションが発生する）、それともあらかじめ小さく分割した2つのTCPパケット（IPフラグメンテーションなしに、2つの独立したIPパケットが生成される）を送信するかという、2つの方法が考えられる。後者では、部分的に先に受信確認（ACK）を返信できるので、1パケット受け取るごとに処理を進めることができる。また、いずれかのパケットが喪失したような場合でも、前者では全IPパケットの再送が必要になるが、後者では欠落した部分だけ再送すればよい。



中堅中小企業の“ネットワーク課題”はこれで解決！ **New!**

@IT Special ^

ミドルの転職・AMBIの人気コンテンツ - PR -



若手7割がスタートアップ転職に意欲 | AMBI（アンビ）



あなたの職務適性が15分でわかる | AMBI（アンビ）



官公庁関連の厳選求人、多数掲載中！「ミドルの転職」

@IT eBook



解決！Python CSVファイル編



「断えてやる！」の前に読むIT新説／徹底解説 vol.5



Windowsファイル共有を“極める”ためのPowerShellコマンドレット基本集



IT人材ゼロから始める中小企業のDXマニュアル

解決！Python CSVファイル編

誰か、要件追加を止めてくれ！——「旭川医大の惨劇」徹底解説

目指せ、共有フォルダ管理の達人！ Windowsファイル共有を“極める”ためのPowerShellコマンドレット基本集

IT人材ゼロでDX!? お悩み中小企業のためのDX推進が分かる無料の電子書籍とは

一覧ページへ

注目のテーマ



AI for エンジニアリング



「サプライチェーン攻撃」対策



1P情シスのための脆弱性管理／対策の現実解




OSSのサプライチェーン管理、取るべきアクションとは



Microsoft & Windows 最前線 2023

システム開発ノウハウ【発注ナビ】 - PR -



「脱リファラル営業」がエンジニアの実力を高める



「AI開発」でおすすめの25社【2023年版】



Excelではもう限界！2万点以上の在庫管理をシステムで解決

このような事情があるため、TCPの送信処理においては、なるべくIPフラグメンテーションが発生しないような工夫が行われている。それが「MSS（最大セグメント・サイズ）」オプションの使用である。MSSとは、上の図「TCPパケットと最大セグメント・サイズ」から分かるように、IPフラグメンテーションなしに送受信可能な最大のTCPデータ・サイズのことを指す。ローカルのイーサネット・ネットワーク上のケースで考えると、MSSは1460bytesとなる。この値は、イーサネットの1パケットで送信可能な最大データ・サイズ1500bytesから、IPヘッダの最小サイズ20bytesとTCPヘッダの最小サイズ20bytesを引いたものである。ただしPPPoEのようなプロトコルの場合は、PPPoEヘッダのために何bytesか必要になるのでさらにMSSは小さくなるが、それについてはここでは触れない。

MSSのサイズが分かると、TCPデータを送信する側では、常にこのサイズ以下になるようにデータを分割して送信することができる（UDPの場合はIPフラグメンテーションは気にせず、より大きなサイズで送信するのが一般的である。UDPの場合は上位層での再送処理などがないからだ）。こうすれば、無用なIPフラグメンテーションを起こすことなく、常に最大の効率でパケットを送信することができる。このMSSサイズを通知するのが、TCPのMSSオプションである。ただしMSSオプションが使われない場合は、送信側は任意のセグメント・サイズで送信することになるので、IPフラグメンテーションが発生する可能性がある。

ウィンドウ・サイズとMSSの関係

ここで、ウィンドウ・サイズとMSSの関係について少し補足しておこう。

TCPでは、MSSのサイズごとにデータを区切って送信すると述べた。となると、最も効率よくデータを送信するためには、ウィンドウ・サイズをMSSの整数倍にするのが望ましいと考えられる。

次の図を見ていただきたい。



ウィンドウ・サイズとMSS

ウィンドウ・サイズがMSSの整数倍ならば、MSSと同じサイズのパケットを送信することができるので、ネットワークの利用効率が最大になる。

ウィンドウ・サイズとは、受信確認なしに受け取ることができる、最大の受信可能容量である。ウィンドウ・サイズがMSSの整数倍になっていれば、最も大きなセグメント・サイズばかりでウィンドウをいっぱいにすることができる。だが整数倍でなければ、最後のセグメントはMSSのサイズよりも小さなパケットにしなければならない。これではネットワークの効率を最大限に生かすことができない（パケットが小さいとヘッダなどの占めるサイズが相対的に大きくなり、帯域を最大限に利用できなくなる）。

このような事情があるため、TCP/IPにおけるチューニングでは、「ウィンドウ・サイズはMSSの整数倍にする」のが望ましいとされている。例えばMSSが1460bytesなら（これはローカルのイーサネットの場合。PPPoEなどが使われている場合はもっと小さい値になる）、ウィンドウ・サイズはその44倍である64,240bytesといった値にすればよい（TCPの標準仕様では、ウィンドウ・サイズは64Kbytesまでなので、44倍というの

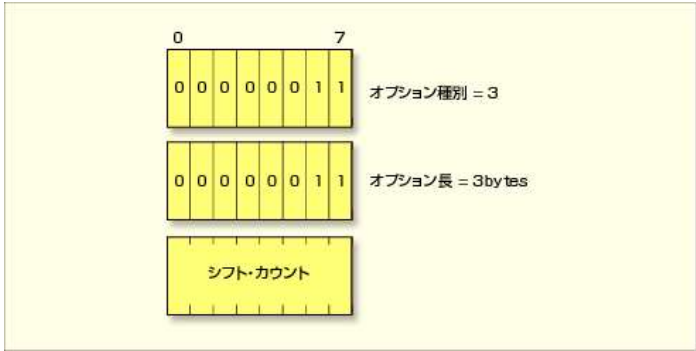


はその最大値に一番近い値である）。ただしむやみに大きくしても必要となるメモリが増えたり、質の悪いネットワーク回線では再送処理に伴うオーバーヘッドなどが増える可能性があるので、常にこの値がよいというわけではなく、ネットワークの環境に応じて適宜変更する必要がある。Windows OSにおけるウィンドウ・サイズは、TCP/IPプロトコル・スタックのレジストリ「TcpWindowSize」や「GlobalMaxTcpWindowSize」の設定に基づいて決定される。デフォルトでは16Kbytes程度になっている（実際の値は、16KbytesをMSSの整数倍になるように切り上げて計算される）。詳細についてはマイクロソフトの技術情報「[Microsoft Windows 2000 TCP/IP 実装詳細](#)」などを参考にしていきたい。

「ウィンドウ・スケール」オプション

ウィンドウ・スケール・オプションは、TCPの標準規格（当初の規格）の制限で、最大64Kbytesとなっているウィンドウ・サイズをより大きくするために利用されるオプションである。TCPヘッダの構造を見れば分かるように（「[連載第15回—1. TCPパケットの構造](#)」の冒頭の図参照）、TCPのウィンドウ・サイズは常に16bitの2進数で表されることになっている。つまり最大値は65,535bytesまでとなっている。

だが高速なネットワーク媒体ではこれでは不十分なので（「[連載第14回—1. 信頼性のある通信を実現するための仕組み](#)」にあるRTTの説明を参照）、より大きなウィンドウ・サイズをサポートするためにこのオプションが新しく定義された。ただしTCP/IPのプロトコル・スタックによってはこのオプションは実装されていない場合もある。その場合はこのオプション要求は無視される。



ウィンドウ・スケール・オプション
シフト・カウントは標準の「ウィンドウ・サイズ」を左へ何bitシフトするかを表す数値。0～14が有効。1ならば、現在のウィンドウ・サイズを左へ1bitシフトし、値を2倍にする。14ならば2の14乗で、16384倍することになる。

このオプションは、TCPヘッダ中にある元の「ウィンドウ・オプション」という16bitの数値に対して、（2進数で数えて）左へ何bitシフトするかということを表している。例えばシフト・カウントの値が2ならば、16bitのウィンドウ・オプションの数値を左へ2bit（×4倍）して、最大値を64Kbytes×4＝256Kbytesにするという意味になる。これにより、非常に大きなウィンドウ・サイズまでサポートすることができる。

	10進数表記	2進数表記
元の数値	64,240	1111_1010_1111_0000
シフト・カウント	2	0000_0010
結果	256,960	11_1110_1011_1100_0000

シフト・カウントの計算例
シフト・カウントは、2進数で表現した場合に、何bit左シフトするかを表す。例えばウィンドウ・サイズ・フィールドの値が64,240bytesで、シフト・カウントが2ならば、最終的なウィンドウ・サイズは×4倍して、256,960bytesとなる。

- PR -



基礎から学ぶWindowsネットワーク 連載一覧

全 23 回

新しい連載記事が 5 件あります

- 第18回 [NetBIOS over TCP/IPプロトコル（その1）](#)
- 第17回 [LLCとNetBEUIプロトコル](#)
- 第16回 **信頼性のある通信を実現するTCPプロトコル（3）**
- 第15回 [信頼性のある通信を実現するTCPプロトコル（2）](#)
- 第14回 [信頼性のある通信を実現するTCPプロトコル（その1）](#)

過去の連載記事が 13 件あります

Special

- PR -



オンプレのハードウェアも「サブスク」の時代へ コストや契約はどう変わる？



「守る」だけでは不十分 今どきのストレージには何が必要？



NTTデータと日本IBMがタッグ！ AIは仕事をどう変える？



自分が作ったアプリがスマホで動くさまを見ると、学生の目が輝きます **New!**



「ほとんど誰も見ていない」社内ポータル、どう変えるべき？ **New!**



「ネットワークが分からない」状態からでも丸ごとサポート **New!**



データは「守りながら活用する時代」に



社内ルールだけでは限界 有名無実化した「ローカル保存禁止」にどう対応？

[@IT Special](#) へ

この記事に関連する製品／サービスを比較（キーマンズネット）

既存のネットワーク構成とマッチする？『WAN高速化』製品の選び方
まずネットワークの性質を十分に見極めよう！『ネットワーク管理』製品比較
信頼性や可用性に対する取り組みは？『ネットワークスイッチ』製品比較
構築したいネットワーク要件で大きく変わる『ルーター』の選び方
L4負荷分散とL7負荷分散どちらを重視？『ADC／ロードバランサ』製品一覧

印刷

通知

見る

Share

8

<div><div>@ITについて</div><div><div>お問い合わせ</div><div>広告について</div><div>採用広告について</div><div>利用規約</div><div>著作権・リンク・免責事項</div><div>サイトマップ</div></div></div>	<div><div>RSSについて</div><div><div>@ITのRSS一覧</div></div></div>	<div><div>アイティメディアIDについて</div><div><div>アイティメディアIDとは</div></div></div>	<div><div>メールマガジン登録</div><div><div>@ITのメールマガジンは、もちろん、すべて無料です。ぜひメールマガジンをご購読ください。</div><div>申し込みページへ</div></div></div>
--	--	--	--