

HTTP/2

梅田 玲旺, 松井 健人, 内村 祐之, 市川 耀

Reo UMEDA, Kento MATSUI, Yushi UCHIMURA, Hikaru ICHIKAWA

1 はじめに

最近の Web サイトでは Web サーバとブラウザ間でやり取りをしなければならないデータの量が増えてきている。そのためページ全体が表示されるのに遅くなる要因になってしまう。HTTP/1.1 では複数のコネクションを同時に使いデータを早く送る方法が存在するがサーバやブラウザの負荷を増やしてしまい余計に遅くさせてしまう可能性がある。また、ダウンロードするのに時間のかかる HTTP のリクエストなどに対する応答が遅くなってしまい、先頭の応答が処理中であるため後続の応答まで全てブロックされるという HTTP/1.1 の仕様もあり、コネクション数を増やしても効果が得られないことがある。

2 HTTP の歴史

HTTP は 1990 年ごろ素粒子物理学の研究所である CERN のティム・バーナーズ・リー氏が HTML と共に CERN 内の情報にアクセスするために設計をした。HTTP/0.9 はテキストがメインの簡単なやり取りのみだったがその後 1996 年に HTTP/1.0 の仕様が公開され Content-Type のような各種ヘッダが追加され音楽や画像、動画などの様々なデータのやり取りに対応した。1999 年に公開された HTTP/1.1 は複数のデータを効率よく転送するための持続的接続やプロキシの利用等を想定した仕様になっていた。持続的接続によって TCP 接続を毎回行わずに繋いだままにして他のデータも通信していくことができる。また 2000 年からは日本では回線が ADSL になりさらに今では光回線となり通信速度は速くなっていった。しかし回線が速くなったためにより多くのデータのやり取りが出来るようになり一つのページを構成する要素が多くなり、またデータ量が大きくなるなどしたためページサイズは増え続けていっている。よって HTTP/1.1 で無駄になってしまっている部分を改良し効率的にする必要がある。

3 HTTP/2 の仕様

3.1 HTTP/2 の特徴

HTTP/2 では実質的な転送速度を向上させることを目的としている。そのために通信路を仮想的に多重化して素早いレスポンスを実現しつつ、ヘッダのデータ圧縮で送信するデータを少なくしている。いままでの HTTP と混在していくために HTTP/1.1 から HTTP/2 の通信に移行するように際に、HTTP/2 に対応しているかどうかを確認する。HTTP/2 に未対応であれば HTTP/1.1 の通信を行うことで互換性を保っている。

3.2 バイナリの使用

HTTP/1.1 のときはテキスト形式を使っていたが HTTP/2 は高速化のためコマンドやパラメータの送信データを「フレーム」という形の決まったバイナリ形式のデータにまとめて送信するようになりフレームで HTTP/2 はやり取りをするようになっている。そうすると小さな容量で転送ができテキストプロトコルとは違い空白の処理や大文字小文字など対応しなくてもよくなり間違いを少なくなり決まりきった形になっているため解析しやすくすることが出来る。フレームは以下 Fig.1 のようになっている。

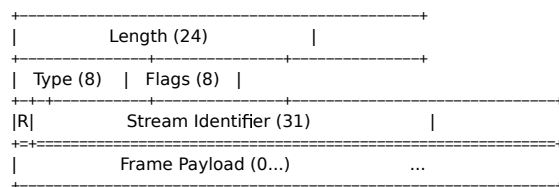


Fig.1 HTTP/2 のフレーム

Length が中身についての大きさで Type はフレームのタイプを表し Flags はフレームタイプのフラグのためにとられている。また R は定義されていない意味のない部分です。Stream Identifier はストリーム ID を表しています。その後は中身です。

3.3 多重化

コネクションを確立してまた「ストリーム」という仮想的な通信路を作りそれを使い双方向に「フレーム」を送受信する。1 つのコネクションで複数のストリームを作ると HTTP/1.1 での複数のコネクションを使ったのと同様以上やり取りができるが HTTP/1.1 では複数のコネクションを使用しやり取りをするが HTTP/2 はコネクションが 1 つで済むためルーターやファイアウォール、プロキシサーバなどのネットワーク全体に対する負担が少なくなる。コネクションのイメージを Fig.2 に示す

図のように HTTP/1.1 とは違いブラウザとサーバ間のコネクションは 1 つだけにしたほうが良いようになっている。HTTP/1.1 でも「パイプライン」機能を使えば、1 つのコネクションで複数のリクエストを連続して送信することが可能だったが、その場合は、ブラウザに帰るレスポンスは送信したリクエスト順になる。だが HTTP/2 のストリームではレスポンスの順番を変更できるので、例えばサーバ側の処理が重い場合は、それ以外の軽い処理のレスポンスを先に返す、といったことが可能になる。それを Fig.3 に示す。

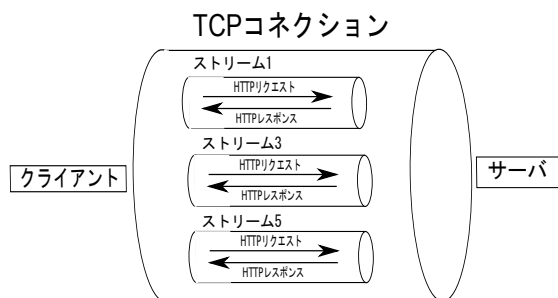


Fig.2 HTTP/2 のコネクションとストリーム

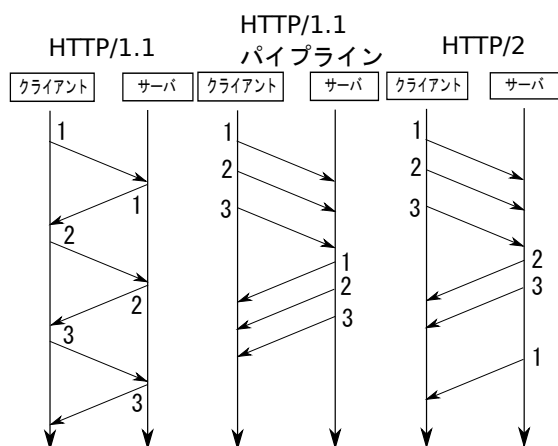


Fig.3 HTTP のリクエストとレスポンス

ストリームには「優先度」を指定することもでき、ブラウザは後から優先度の高いリクエストを送って、その結果を先に返すようにサーバに対して要求できる優先度制御がある。例えば文字よりも表示するために時間のかかる画像を優先度を上げ先に送って貰うことができるようになる。HTTP/2では通信の状態を制御する機能として、「フロー制御」機能も用意されている。一度に送信可能なデータ量を調整して受信側のバッファがあふれて容量不足になってしまうのを防いだり、逆に一度に送受信できるデータを大きく指定して、巨大なデータを効率よく送信させたりできる。HTTP/1.1ではTCP/IPのネットワーク機能を実現するために必要なTCPの機能に含まれるウィンドウ制御に依存していたが、HTTP/2ではWebサーバやブラウザがある程度制御できるようになる。HTTP/2では、よりよくエラー処理がしやすいようになっている。HTTP/1.1ではエラーが発生した場合、サーバがそのリクエストを受け付けて処理したのか、それとも単にリクエストが届いていないのか、などを区別する方法がなかった。処理内容によっては、リクエストを再送して再実行すると不具合を起こすものもあり、注意してシステムを設計する必要があるがHTTP/2では、ストリームを使うことにより、どのストリーム番号の処理まで完了したかをブラウザ側から確実に確認できるようになっており、より信頼性の高いシステムを構築できる。

3.4 ヘッダ圧縮

HTTPのリクエストとレスポンスには、「HTTPヘッダ」と呼ばれるテキストデータが含まれている。この部分は長さがある割には決まった文字列も多く、また連続する複数のHTTPヘッダを観測して比較すると、その差が少ないことが多い。そこで、このHTTPヘッダ部分を圧縮できれば通信データの削減につながる。もし1ページが約80個の要素で構成されており、各リクエストが1400byteのヘッダを持つと仮定するとヘッダの送出だけで7-8回データのやり取りが必要になってしまうため圧縮することによりパケットの数を減らすことになりデータのやり取りが減るためパフォーマンスの向上が期待できる。そのためHTTP/2では「HPACK」という方法が採用されている。HPACKはハフマン符号により送信した平均のデータ量が少なくすることとヘッダの差分情報のみを送るReference Set、またよく使うヘッダと値をペアにしてIDを登録し設定されたIDのやりとりによってヘッダを送信するStatic Tableによってできている。

3.5 サーバプッシュによるデータ送信

サーバプッシュによるデータ送信とはWebサーバ側からWebブラウザに対して、あらかじめ必要となるデータを送信しておく機能である。そのようにするとブラウザがHTMLを解析するまで他の必要な要素のリクエストを出すことが出来ないブラウザにサーバはページを表示するために必要な画像などを送ることによりレスポンスがすべて帰って来るまでの時間を減らすことができる。

4 HTTP/2 今後

HTTP/2を使うためにはサーバ側とブラウザ側の両方が対応していく必要があるが主要ブラウザの最新バージョンはHTTP/2に対応しておりその中には「Windows10」の「Internet Explorer 11」、「Firefox 36」、「Chrome 40」などがある。HTTP/2では暗号化をすることによりセキュリティを向上させるTLS利用が必須ではなくなったがChromeやFirefoxではTLS利用をしたHTTP/2のみをサポートするなどセキュリティ面でも良くなっていくと考えられる。HTTP/2はまだIETFに承認されてRFCとして文章化されている最中でありHTTP/1.1に最適化されていると単にHTTP/2に変えるだけではページの表示は遅くなる可能性もありまだHTTP/2の恩恵を受けるためには時間がかかる。

参考文献

- 1) 高速・大規模ネットワーク時代に向けて改良されたHTTP/2 プロトコル。
<http://www.atmarkit.co.jp/ait/articles/1409/18/news135.html>
- 2) ウェブを高速化する「HTTP/2」を知る
<http://japan.zdnet.com/article/35061196/>
- 3) Stephe Thomas, 『HTTP プロトコル セキュア&スケーラブルなWeb開発』(ソフトバンクパブリッシング)

グ, 2002)