Vol.2017-GN-100 No.16
Vol.2017-CDS-18 No.16
Vol.2017-DCC-15 No.16
2017/1/20

リアルタイム協調作業環境のためのHTTP/2を用いた同期機構の構築

山本 修平[1]　西出 亮[2]　高田 秀志[2]

概要：Web ブラウザを用いたリアルタイム協調作業支援アプリケーションでは，リアルタイムに端末間でデータを同期する必要がある．しかし，Web サーバとブラウザの間にあるネットワーク環境によっては，画像などのデータ量の多いものが配信される際に，その他の同期するデータの配信に遅延が発生する可能性がある．そこで本研究では，リアルタイム協調作業環境において，優先度制御が可能な同期機構の構築を行う．この同期機構では，HTTP/2 の特徴の 1 つである優先度制御機能を拡張し，同期するデータごとに優先度を定義することのできる機能を実現する．この機能を用いて，アプリケーション開発者は，同期するデータがブラウザから送信される際に優先度を付加することができる．この優先度が付加されたデータを受信した Web サーバが，優先度を解析することで，優先的に配信するデータを決定する．

A Synchronous Mechanism Using HTTP/2 in Real-time Collaborative Work

Yamamoto Shuhei[1]　Nishide Ryo[2]　Takada Hideyuki[2]

Abstract: Applications for real-time collaborative work using Web browser require synchronizing data among terminals in real time. When a large amount of data distributed from a Web server, delivery delay may occur in other data depending on the network environment. In this research, we construct a synchronous mechanism that can distribute the data to the clients based on defined priority using HTTP/2. We extend the priority control function of HTTP/2 to define priority for each data to be synchronized. The application developers can assign a priority on the request header when the data is distributed from the browser. When receiving this data, the Web server chooses the data to distribute preferentially by parsing priority.

1. はじめに

近年，Web ブラウザのリッチクライアント化やネットワーク環境が整備されてきたことにより，Web ブラウザを用いた同期型の協調作業支援アプリケーションを実現する環境が整ってきている [1][2]．このような協調作業支援アプリケーションを複数人で使用できるようにするためには，Web ページを取得するのに加えて，Web ページ内に記述された JavaScript プログラムによって Web ブラウザ間でアプリケーションデータの同期処理を行う必要がある．このデータの同期を Web サーバを介して行う場合，Web サーバとブラウザの間にあるネットワーク環境によっては，画像などのデータ量の多いものが配信される際に，その他の同期するデータの配信に遅延が発生する可能性がある．この遅延が発生することによって，リアルタイムな協調作業を支援することが難しくなると考えられる．例えば，複数人で行う同期型の協調作業において，他のクライアントがどの部分を編集しているかをリアルタイムに同期する際に，画像などのデータ量の多いものが先に同期されると，リアルタイムに同期すべきデータが遅延する可能性がある．このことから，リアルタイム性の高いデータをより優先的に同期することで，円滑な協調作業を支援することができると考えられる．

[1] 立命館大学大学院情報理工学研究科
Graduate School of Information Science and Engineering, Ritsumeikan University
[2] 立命館大学情報理工学部
College of School Information Science and Engineering, Ritsumeikan University, Kusatsu, Shiga 525-8577, Japan

Vol.2017-GN-100 No.16
Vol.2017-CDS-18 No.16
Vol.2017-DCC-15 No.16
2017/1/20

Hypertext Transfer Protocol 2(HTTP/2)[3] では，優先的にレスポンスを返して欲しいデータを，Web ブラウザが Web サーバに伝えることで，リソースの割り当てをコントロールすることが可能な優先度制御機能を搭載している．この機能を利用することにより，例えば，クライアントは Web ページの枠組みを表示するための HTML ファイルや CSS ファイルを他のデータより優先的に受け取ることが可能になる．これによって，Web ページの初期描画を早く実行することができる．しかし，この優先度制御機能は，Web ページを取得する際に利用することを想定しているため，JavaScript プログラムによって Web ブラウザ間でデータの同期を行うような協調作業支援アプリケーションに対して有効に働くかは明らかにされていない．また，これらの優先度情報はコンテンツの種類に応じて固定されているため，同期するデータごとに異なる優先度を指定することができない．

そこで，本研究では，HTTP/2 を用いたリアルタイム協調作業環境における同期機構の構築を行う．この同期機構では，協調作業支援アプリケーションの開発者が自由に優先度を設定することを可能にするため，同期データのリクエストに含まれるヘッダに優先度を含められるようにする．この優先度を Web サーバで解析することで，優先度に基づいたリソースの割り当てを行う．

2. 関連研究

2.1 Web ブラウザを用いた同期的な協調作業

Web ブラウザを用いた同期的な協調作業支援アプリケーションの例として，複数人で Web ページをリアルタイムに編集できる WFE-S(Real-Time Collaborative Editing System)[4] と，協調型資料添削支援システム [5] がある．WFE-S では，テキストの編集や，コメントの挿入，画像ファイルのアップロードなどの機能をシステムのユーザに提供している．この協調作業では，複数のユーザが同一箇所を同時に編集する可能性があるため，編集の競合を防ぐ必要がある．したがって，各ユーザが編集している領域情報などを，画像ファイルなどよりも優先的に同期する必要があると考えられる．同様に，協調型資料添削支援システムでは，システムのユーザが同期的に協調し，資料の添削を行うことで，通常の紙ベースの資料添削の課題を解決することを目的としている．被添削者が Web アプリケーションのサーバへ資料をアップロードし，添削者とリアルタイムに議論を交わすことにより，資料の質の向上を目指している．このシステムにおいて，同期されるデータとしては，添削者が資料のどこに注目しているかを表す資料閲覧情報や，図形やコメントによる添削情報などがある．特に，資料閲覧情報は，各ユーザ間で同期の誤差が生じた場合，議論をうまく進めることができないため，他の情報よりも優先的に同期を行う必要があると考えられる．

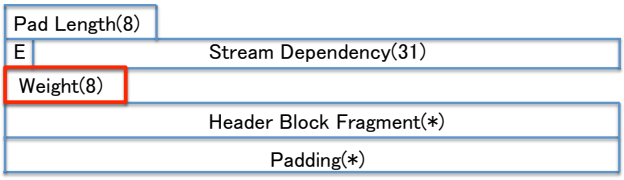

図 1　HEADERS frame

上記のことから，リアルタイム協調作業環境におけるアプリケーションでは，優先的に同期する必要があるデータが存在すると考えられる．また，優先的に同期する必要のあるデータはアプリケーションごとに異なることが予想されるため，アプリケーション開発者が，優先的に同期するデータを指定できる仕組みが必要である．そこで，本研究では，アプリケーション開発者が優先的に同期するデータを指定することが可能な同期機構の構築を行う．この同期機構では，同期データを Web サーバから取得する際のリクエストに，協調作業支援アプリケーション開発者が指定した優先度を付加し，Web サーバ側で優先度に基づいたデータ配信を行うことで，優先度制御を行う．また，本研究で想定するアプリケーションは，データの更新が行われた時にページ全体を再ロードするのではなく，動的にページを変更するシングルページアプリケーションとする．したがって，更新データの送受信は Ajax 等の非同期通信を用いて行うこととする．

2.2 HTTP/2 における優先度制御

HTTP/2 における優先度制御機能について説明する．HTTP/2 では，Web ブラウザで指定される重みと依存関係の 2 つを組み合わせた優先度情報に基づいて，Web サーバが優先的にレスポンスを返すコンテンツを決定している．優先度情報の 1 つである重みには，1〜256 の段階を指定することが可能であり，Web サーバは各リクエストに付加されている重みの割合に応じて割り当てるリソースを変更している．また，この重みは Web ブラウザで生成される図 1 のような HEADERS フレーム内の Weight に格納されている．

この優先度制御がページロードタイムにどの程度影響するかを計測した研究がある [6]．この研究では，HTML>CSS>JS>Images の順番で高い優先度を付加した上で，Web ページを取得する実験を行っている．その結果，優先度制御を行った場合，ページロードタイムが約 6% 改善したという結果を得ている．

HTTP/2 における優先度制御では，主に Web サーバから Web ページを取得することを想定しているため，協調作業などの他クライアントとのデータの同期が必要な場合については考慮していない．しかし，このような場合においても，適切な優先度を設定することにより，協調作業支援アプリケーションに有効に働くことが期待される．また，

Vol.2017-GN-100 No.16
Vol.2017-CDS-18 No.16
Vol.2017-DCC-15 No.16
2017/1/20

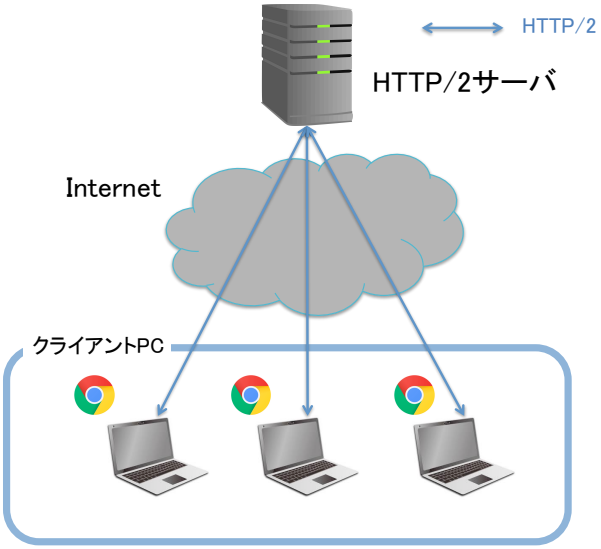

図 2　ネットワーク構成

これらの優先度を Web ブラウザで指定する仕組みは，Web ブラウザごとに実装状況が異なる．これに加えて，優先度は HTML や CSS などのコンテンツの種類ごとに固定されているため，協調作業アプリケーションの同期データごとに優先度を変更することができない．したがって，本研究で実現する同期機構では，アプリケーション開発者が優先度を指定することを可能にするため，X-Priority という項目を HEADERS フレームに追加する．この X-Priority を Web サーバで解析することで，優先度を細かく指定することができる優先度制御を実現する．

3. HTTP/2 を用いた同期機構

3.1 X-Priority による優先度制御

はじめに，対象とするネットワーク構成について述べる．本研究で対象とするネットワーク構成を図 2 に示す．各クライアント間でのデータの同期は Web サーバを介して行うこととする．この HTTP/2 サーバには，HTTP/2 の参照実装である H2O[7] を用いる．

次に，優先度をアプリケーションで指定することができるように追加する X-Priority の概要について述べる．HTTP/2 では，より効率的に情報をやりとりするために，フレームと呼ばれるバイナリ形式のフォーマットを採用している．このフレームの中に存在する Weight の値を使用して，優先度制御が行われる．本手法では，Web ブラウザ側で，HEADERS フレームに X-Priority という項目を追加し，Web サーバがこの項目を解析することで，優先度制御を行う．この X-Priority で指定できる優先度情報の種類を表 1 に示す．これらの種類のうち 1 つを，JavaScript を用いて Web サーバへのリクエストを生成する際に，引数により指定する．

次に，Web サーバ側で X-Priority を解析する手順について説明する．H2O では，優先度に基づいた送信データの選

表 1　X-Priority で指定する優先度情報

種類	重み
Lowest	50
Low	100
Middle	150
High	200
Highest	250

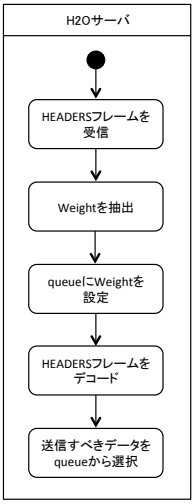

図 3　優先度に基づいた送信データの選択

択を図 3 のように行っている．まず，クライアントから送信された HEADERS フレームを受信すると，HEADERS フレーム内の Weight の値を抽出し，この値を queue に設定する．この後，HEADERS フレームの解析を行い，queue から Weight の値を基に，送信するデータを決定する．本手法では，X-Priority を HEADERS フレームに追加しているため，優先度情報を queue に設定する前に，HEADERS フレームのデコードを行う必要がある．したがって，Weight の値をデコードした後，X-Priority の値を取り出すために，HEADERS フレームのデコードを先に行うように処理を改変する．X-Priority の値を抽出することができた際に，優先度情報を queue に設定することで，X-priority を用いた優先度制御を実現している．

3.2 アプリケーション

リアルタイム協調作業環境において，優先度制御が機能するかを検証するために，複数人で画像を同期できる Web アプリケーションを作成した．このアプリケーションは，テキストの挿入や画像のアップロード，画像のドラッグ機能を持つ．画像のアップロード時には，画像ファイルと画像の位置情報を別々に Web サーバへアップロードしている．

次に，データの同期方法について述べる．各クライアントがデータの更新を行った際に，Web サーバは他のクライアントに対して，データの更新を通知する必要がある．このデータの更新通知は Server-sent events[8] を利用して

Vol.2017-GN-100 No.16
Vol.2017-CDS-18 No.16
Vol.2017-DCC-15 No.16
2017/1/20

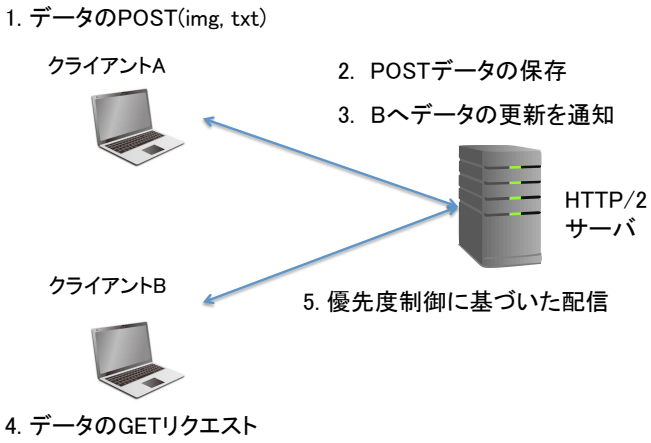

図 4　更新データの同期

行う．まず，はじめに Web サーバへアクセスした際に，Server-sent events を用いて，クライアントと Web サーバ間にコネクションを確立する．このコネクションを用いて，データの同期を行う際の概要を図 4 に示す．クライアント A でデータの更新があった時，画像データやテキスト情報，画像の位置情報などをサーバへ POST する．これを受け取った HTTP/2 サーバは POST データの保存を行い，Server-sent events によってクライアント B へデータの更新を通知する．データの更新を受け取ったクライアント B は，更新データに対する GET リクエストを生成し，HTTP/2 サーバへ送信する．この GET リクエストに X-Priority を付加することで，同期データごとに優先度情報を割り当てることが可能になる．最後に，GET リクエストを受け取った HTTP/2 サーバが X-Priority を解析し，優先度に基づいた配信を行うことで，データの同期を行う．

4. 評価実験

4.1 X-Priority による優先度制御

4.1.1 実験概要

提案手法による優先度制御が機能するかを検証するための実験を行った．今回の実験では，優先度の高い情報と低い情報の 2 種類を用意し，それぞれのレスポンスタイムの検証を行う．このレスポンスタイムは，クライアントが GET リクエストを行ってからコンテンツをダウンロードできるまでの時間を意味する (図 4 の 4 および 5)．また，優先度情報をクライアント側で付加しない方法との比較も行う．

次に，実験環境について述べる．本実験では，Web サーバとクライアントが同一無線 LAN 環境下に存在し，1 対 1 でコンテンツの受信および配信を行う．クライアントは Google Chrome を用いて Web サーバにアクセスし，コンテンツを取得することとする．また，検証を行うためのデータとして，テキストファイル (5KB) と，画像ファイル (5MB) を用意した．優先度をクライアント側で付加する場合は，テキストファイルの優先度を High, 画像ファイルの優先度を Low に設定することとする．

表 2　各リクエストの順番

順番	ファイルの種類
1〜3	テキストファイル
4	画像ファイル
5〜7	テキストファイル
8	画像ファイル
9, 10	テキストファイル

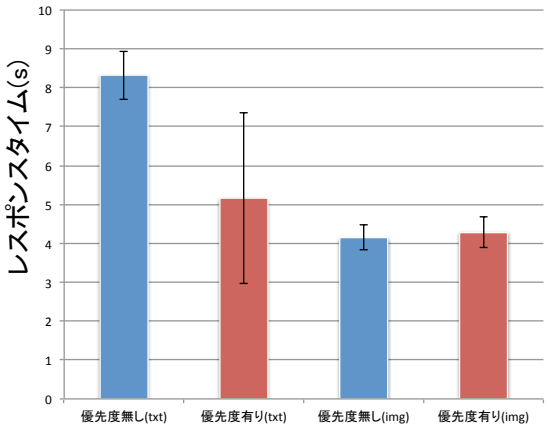

図 5　X-Priority による優先度制御

最後に，実験の手順について述べる．まず，はじめにクライアントが Web サーバに対してそれぞれのファイルのリクエストを行う．このときのリクエストを行う順番を表 2 に示す．これらの一連のリクエストを 5 度試行し，画像ファイルとテキストファイル，それぞれのレスポンスタイムを計測する．

4.1.2 実験結果

HTTP/2 では複数のリクエストを，レスポンスを待たずに行えるため，本実験における一連のリクエストは一斉にクライアントから Web サーバに対して送信されている．リクエストを行ってからデータがダウンロードできるまでのレスポンスタイムの平均を図 5 に示す．ここで，エラーバーは分散値を表している．この結果から，優先度有りのテキストファイルのレスポンスタイムが優先度情報を付加しない場合と比較して短いことがわかる．また，画像ファイルのページロードタイムについては，優先度無しの場合の方が，優先度有りの場合と比較して，短くなっている．

実験結果から，提案手法における X-Priority をクライアント側で付加し，Web サーバで解析することで，優先度制御に基づいた配信が行われることがわかった．優先度情報を付加した場合，優先度の高いテキストファイルに多くのリソースが割り当てられるため，優先度を付加しない場合に比べてレスポンスタイムが短くなったと考えられる．一方で，優先度を付加しない場合，画像ファイルのダウンロードにリソースが割当てられている間に，テキストファ

Vol.2017-GN-100 No.16
Vol.2017-CDS-18 No.16
Vol.2017-DCC-15 No.16
2017/1/20

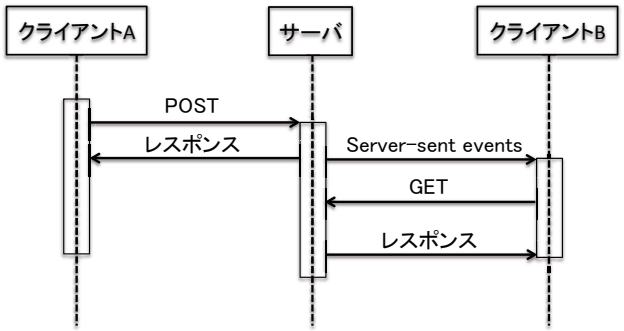

図 6　クライアントからのリクエストについて

イルのダウンロードが待たされてしまうため，レスポンスタイムが長くなる．また，優先度有りのテキストファイルの分散値が大きいことは，優先度制御が機能しない場合があることが関係していると考えられる．HTTP/2 の優先度制御では，クライアントからの優先度情報を参考に行うため，必ずしも優先度制御が機能するわけではない．したがって，優先度制御が機能しない際に，レスポンスタイムが長くなるため，分散値が大きくなったと考えられる．また，画像ファイルのレスポンスタイムに大きな差が見られないことは，テキストファイルのデータ量が小さいことが関係している．テキストファイルのデータ量が小さい場合，テキストファイル自体のダウンロード時間が短いため，画像ファイルのダウンロードが待たされる時間も短くなる．したがって，テキストファイルの優先度を付加した場合においても，画像ファイルのレスポンスタイムがそれほど長くならないことがわかる．

4.2　協調作業環境における優先度制御

4.2.1　実験概要

協調作業環境において，提案手法が有効に機能するかを検証するために評価実験を行った．協調作業時に行われるテキストや画像のアップロードを行い，他のクライアントがこれらのコンテンツを取得するまでのレスポンスタイムを計測する．ここで，本実験で発生させるリクエストのタイミングを図 6 に示す．計測するレスポンスタイムは，クライアント PC が Web サーバからデータの更新通知を受けた後の GET リクエストからレスポンスとしてコンテンツをダウンロードするまでとする．また，クライアント PC は 3 台用意し，すべて無線 LAN に接続されているものとする．

次に，実験内容について述べる．2 台のクライアント PC がテキストや画像のアップロードを行い，1 台のクライアント PC で発生する GET リクエストのレスポンスタイムを計測する．ここで，アップロードを行うデータを表 3 に示す．この制御情報は，テキストや画像の挿入位置を示すテキストデータを意味する．また，これらのデータは，表 4 に示す順番で 3 回アップロードされる．この試行を，優先度を付加した場合と付加しない場合のそれぞれで，5 回試行した．

表 3　アップロードデータ

データの種類	データ量	優先度
制御情報	5KB	High
テキスト	5KB	Low
画像	1MB	Low

表 4　アップロードのタイミング

順番	ファイルの種類
1	画像
2	テキスト
3	制御情報

表 5　優先度有り

種類	平均値 (ms)	分散値	中央値	最大値	最小値
画像	0.754	0.393	0.780	1.69	0.19
テキスト	0.256	0.109	0.220	0.598	0.157
制御情報	0.251	0.111	0.206	0.582	0.158

表 6　優先度無し

種類	平均値 (ms)	分散値	中央値	最大値	最小値
画像	0.8886	0.458	0.736	1.92	0.202
テキスト	0.223	0.092	0.195	0.587	0.15
制御情報	0.233	0.100	0.202	0.602	0.15

4.2.2　実験結果

優先度を付加した場合のレスポンスタイムの結果を表 5 に示す．同様に，優先度を付加しない場合の結果を表 6 に示す．この結果から，優先度情報を付加した場合の中央値に注目すると，制御情報のレスポンスタイムがテキストよりも短くなっていることがわかる．また，優先度を付加しない場合の中央値に注目すると，制御情報のレスポンスタイムがテキストよりも長くなっていることがわかる．

この結果から，協調作業環境時において，わずかな効果であるが，X-Priority による優先度制御が機能する場合があることがわかった．画像データに関しては，優先度制御によるレスポンスタイムの変化を観測できなかったが，制御情報とテキストデータに関しては優先度制御によるレスポンスタイムの変化を観測することができた．これは，制御情報とテキストの GET リクエストが連続して行われたことが関係していると考えられる．クライアント PC から各データがアップロードされる際，画像データはデータ量が多いため，Web サーバへのアップロードに時間がかかる．これにより，レスポンスタイムを計測するクライアント PC 上では，制御情報やテキストデータの変更通知が先に到着し，これらのデータの GET リクエストが連続して行われる形式が観測された．したがって，Web サーバ上で制御情報やテキストの GET リクエストを連続して受信した際に，優先度制御が機能したと考えられる．今回の結

果では，制御情報やテキストデータのデータ量が少ないため，レスポンスタイムの差はわずかであるが，データ量の多いファイルなどを扱った際に，より有効に働くことが予想される．

5. おわりに

本研究では，アプリケーション開発者が優先度を設定可能な X-Priority を利用した同期機構の構築を行った．この機構を用いて，Web サーバ上に存在するコンテンツを X-Priority の値に基づいて取得する実験において，優先度制御が機能していることを確認できた．また，Web ブラウザ間でデータの同期を行う協調作業環境においては，優先度の高いデータが優先的にリソースを割り当てられることを確認した．今後は，データの同期のタイミングや同期データのサイズなどを変更し，より優先度制御が有効に働く環境を検証する．

参考文献

[1] Song, Y., Wei, W., Deng, L., Du, P., Zhang, Y. and Nie, D.: 3D-colladesign: A real-time collaborative system for web 3D design, *Computer Supported Cooperative Work in Design (CSCWD), 2015 IEEE 19th International Conference on*, IEEE, pp. 407–412 (2015).
[2] Kruajirayu, H., Tangsomboon, A. and Leelanupab, T.: Cozpace: a proposal for collaborative web search for sharing search records and interactions, *Student Project Conference (ICT-ISPC), 2014 Third ICT International*, IEEE, pp. 165–168 (2014).
[3] Belshe, M., Thomson, M. and Peon, R.: Hypertext transfer protocol version 2 (http/2) (2015).
[4] Inoue, R., Kato, Y., Goda, T., Ozono, T., Shiramatsu, S. and Shintani, T.: A real-time collaborative mechanism for editing a web page and its applications, *2012 Fifth International Symposium on Parallel Architectures, Algorithms and Programming*, IEEE, pp. 186–193 (2012).
[5] 片山真也，合田拓史， 白松俊，大囿忠親，新谷虎松：Web 技術に基づく協調型資料添削支援システムの実現，人工知能学会全国大会論文集，Vol. 27, pp. 1–4 (2013).
[6] de Saxcé, H., Oprescu, I. and Chen, Y.: Is HTTP/2 really faster than HTTP/1.1?, *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, IEEE, pp. 293–299 (2015).
[7] Oku, K.: H2O, https://github.com/h2o/h2o.
[8] Hickson, I.: Server-sent events, *W3C Working Draft WD-eventsource-20091222, latest version available at¡ http://www. w3. org/TR/eventsource* (2009).