

A server, a method for operating a server and a system

Jukka Manner

Le Wang

Aalto University

Teknologiakuvaus
Technology description
14.5.2012

A server, a method for operating a server and a system**Field of the invention**

The invention relates generally to data processing and more
5 particularly to retrieving of content.

Background art

In the following, with content we mean any content that is
loaded in bursts. In particular, content may be loaded from the
Internet and include at least one dynamic web page that is a web
10 page that is prepared with fresh information (content and/or
layout) for each individual viewing. In this context, the term
content would refer specifically to web content.

Content is normally loaded at a terminal (the term "terminal" in
the following may in particular but not necessarily refer to a
15 mobile terminal or mobile device but it can be a desktop or a
laptop or any other client device) to a browser piece by piece.
First, a description of the dynamic page, such as a Hypertext
Markup Language (HTML) page is loaded. The browser parses the
HTML page and submits then separate requests for each piece that
20 with the description of the dynamic page thus form the dynamic
content. Currently, it is normal that a single content contains
some 30 - 100 web objects that must be loaded, not rarely from
different locations. The web objects may include images,
advertisement banners, flash animations etc.

25 The energy consumption of a terminal that is loading a content
is usually higher during the loading as compared with the
terminal in its idle state, that is, when the user is reading
the content that already has been loaded.

In addition, it may be that the loading itself takes longer than
30 expected, in particular when some of the web objects cannot be
retrieved. Even if the web objects could be retrieved, it may be
that they must be retrieved from a remote server which is prone
to increase the duration of the retrieval, especially if the

retrieving is carried out over a slow link that may be a wireless link.

There are some methods that can be used to reduce energy consumption during loading of contents. In the GPRS or 3GPP standards, the wireless terminal loading a dynamic content should enter in idle state after a predetermined time has lapsed, if the requested content is not being transmitted to the terminal. Since transforming the terminal to the idle state requires some signalling, some mobile phone models are believed (cf. Network Efficiency Task Force Fast Dormancy Best Practices, GSM Association, May, 2010) to enter a sleep state aggressively in order to extend battery life and minimize network congestion but may cause signal load problems in the operator network. During browsing, the mobile device has no chance to enter idle or low power state due to HTTP requests and replies going forth and back.

RRC state transition model has remained rather unchanged through several releases, namely, Rel. 99, Rel. 05 and Rel. 06. Perala et al. present in "Theory and practice of RRC state transitions in UMTS networks", in Proc. Fifth IEEE Broadband Wireless Access Workshop (BWA), co-located with IEEE GLOBECOM 2009, Hawaii, USA, November-December 2009 a methodology to discover RRC configuration parameters without operator involvement or cooperation, which shows how the operator network settings may differ drastically from each other.

Feng Qian et al. in "Characterizing Radio Resource Allocation for 3G Networks", in ACM Internet Measurement Conference (IMC) 2010, Melbourne, Australia, and Feng Qian et al. in "TOP: Tail Optimization Protocol for Cellular Radio Resource Allocation" in IEEE ICNP 2010, Kyoto Japan characterize the impact of operational state machine settings and show that tail time period matching the in-activity timer value before a state demotion. During a tail time, a user equipment still occupies transmission channels and its radio power consumption is kept at

the corresponding level of the state, even through there is no traffic transmitted during the period.

Signals Research Group shows in "Reducing the impact of smartphone- generated signaling traffic while increasing the battery life of the phone through the use of network optimization techniques" (May 2010) how signaling traffic generated by smartphones affects battery life and causes network congestion.

In addition to the Fast Dormancy Best Practices paper, "Understanding Smartphone Behavior in the Network" by Nokia Siemens Networks Smart labs (2011) also discusses the timer problem and shows that fast dormancy is one of the solutions, which forces devices to hop back to low power consumption state quickly to save energy and reduce traffic load.

Objective of the invention

The inventors have found out that it is advantageous if the referenced objects belonging to a content are first retrieved by a server and then sent to the terminal requesting the content only after the server has received all or at least most referenced objects belonging to the content. This arrangement enables a more efficient transmission of data from the server to the terminal. The solution has been disclosed in more detail in patent application PCT/EP2012/053214, still unpublished at the date of the first patent filing of the present invention.

The arrangement described in the preceding paragraph may be used to improve the energy efficiency at the terminal when contents are received at the terminal.

The solutions referred to in the previous section may require modifications at the terminal. Even though the proposed solution can reduce power consumption significantly, there still exists the need for more flexibility of deploying the bundling concept. Having the way of operation of a terminal -in

particular a mobile terminal- reprogrammed, can be difficult in view of the rather large number of different operating systems and versions, manufacturers and device types involved. In addition, reprogramming a terminal is not considered to be a transparent method since different data processing would be required at the server depending on whether a particular terminal has been reprogrammed or not.

Therefore, it is an objective of the present invention to increase transparency in downloading content to a terminal from a server.

Summary of the invention

The objective can be met with a server according to independent claim 1, with a method according to parallel independent claim 6, and with a system according to parallel independent claim 11.

The dependent claims describe various advantageous aspects of the server and the method.

Advantages of the invention

With a server that is configured to:

- a) in response to receiving a request for content from a terminal, to respond to the terminal with a response containing instructions configured to set up a bi-directional communication channel at the terminal for communication between the server and the terminal;
- b) to load the content comprising at least one referencing object and a plurality of referenced objects belonging to said content;
- c) to generate a modified referencing object by replacing in the referencing object such links to referenced objects that are external links with modified links that point to a locally stored version of the same object;

d) to send the referenced objects to the terminal over the bi-directional communication channel; and

5 e) to send the modified referencing object to the terminal, in such a manner that the modified links in the modified referencing object point to referenced objects already sent to the terminal,

the downloading of content to the terminal can be made in a manner completely transparent at the terminal which was not the case with any of the solutions referred to above.

10 Analogously, with a method for operating a server that comprises the steps of:

15 a) in response to receiving a request for content from a terminal, responding to the terminal with a response containing instructions configured to set up a bi-directional communication channel at the terminal for communication between the server and the terminal;

b) loading the content comprising at least one referencing object and a plurality of referenced objects belonging to said content;

20 c) generating a modified referencing object by replacing in the referencing object such links to referenced objects that are external links with modified links that point to a locally stored version of the same object;

25 d) sending the referenced objects to the terminal over the bi-directional communication channel; and

e) sending the modified referencing object to the terminal, in such a manner that the modified links in the modified referencing object point to referenced objects already sent to the terminal,

the downloading of content to the terminal can be made in a manner completely transparent at the terminal which was not the case with any of the solutions referred to above.

Our main idea is that, by using the server and the method, the terminal receives a response from the server. Because the response contains instructions configured to set up a bi-directional communication channel at the terminal for communication between the server and the terminal, the referenced objects can be transferred over the bi-directional configuration channel between the server and the terminal i.e. not over the same protocol that is used to transmit requests for content and responses to such requests. This separation makes it possible to transmit the referenced objects together in a single transfer. This can surprisingly be used, since when the bi-directional configuration channel is not active, also the terminal does not need to be active, to save energy during such non-active periods. This is different from HTTP pipelining. HTTP pipelining allows the terminal to request and receive multiple objects from the same originating server over the same transport connection, while in our idea the whole content is collected together from potentially multiple sources and then sent through a single transport connection. According to our idea, there are much less HTTP GET requests coming from the terminal necessary, since the server sends all objects directly to the client.

When the server is configured to send the modified referencing object to the terminal only after the server has sent the referenced objects to the terminal, the terminal already has the referenced objects locally available and can thus use the referencing object without delay.

Most advantageously, the server is configured to set up the bi-directional communication channel as a WebSocket. WebSocket is being standardized by the W3C, and the WebSocket protocol has been standardized by the IETF as RFC 6455. "File API: Directories and System" is now W3C working draft, which defines

an API to navigate file system hierarchies, and defines a means by which a user agent may expose sandboxed sections of a user's local filesystem to web applications (cf.

<http://www.w3.org/TR/file-system-api/>). Advantageously, the

5 instructions of setting up WebSocket and Storing referenced objects are all generated in one processing step (P201, cf. FIG 4). Currently, WebSocket is supported by almost all desktop browsers, and of mobile broesers by IOS Safari and Opera Mobile (cf. <http://caniuse.com/websockets>). It seems File API is now
10 only support by Google Chrome and no mobile browser supports it (cf. <http://www.html5rocks.com/en/tutorials/file/filesystem/>). In our current implementation, the referencing object is a hypertext markup language page and the referenced objects are web objects referred to in said hypertext language page.

15 If the server is configured to send the referenced objects to the terminal in a bundle, the communication over the WebSocket may be done more efficiently as the sending of the bundle can be done at one go in contrast to sending the referenced objects one by one. In addition, now the terminal has a chance to enter low
20 power consumption state while the server is loading and bundling and before it sends the reply to the terminal. The chances are increased even more when this concept is used together with the fast dormancy concept as referred to above. Most advantageously the bundle is sent only after having sent instructions to set up
25 a WebSocket and instructions to save the referenced objects locally at the terminal.

A system comprises at least one server according to the previous aspect of the present invention, and a plurality of terminals that preferably are mobile terminals. The system is configured
30 to use the at least one server for responding to requests for content that have been sent by the terminals.

List of drawings

In the following, the invention is described in more detail with reference to the example in the attached drawings in FIG 1 to 4, of which:

5

FIG 1 illustrates the server setting up a WebSocket between terminal and server;

10

FIG 2 illustrates the server loading of objects over HTTP protocol and sending the objects to the terminal via WebSocket;

15

FIG 3 illustrates the server sending the modified referencing object to the terminal for the terminal then to load the objects to browser;

20

FIG 4 illustrates signalling between the terminal, server and a remote server; and

FIG 5 illustrates the principle of generating a modified referencing object.

25

The same reference numerals and symbols refer to same components in all FIG.

Detailed description

FIG 1 illustrates server 120 setting up a WebSocket WS between terminal 10 and server 120. The server 120 may in particular be a proxy server (network proxy) but, alternatively, it may be a content server. The terminal 10 may be any client device, in particular it may be a wireless or mobile device or wireless or mobile terminal such as a smart phone or PDA, or a computer device, such as in particular a desktop, notebook or tablet computer.

The terminal 10 and server 120 both have at least one protocol stack suitable for bi-directional communication. In particular, in the protocol stack, Media Access Control (MAC), Internet Protocol (IP), Transfer Control Protocol (TCP), Hypertext
5 Transfer Protocol (HTTP), have preferably been implemented.

The preferred embodiment of the invention is in the following discussed with reference to the signalling diagram in FIG 4 and the protocol stack illustrations in FIG 1 to 3.

Terminal 10 sends to server 120 request M201 that most
10 preferably is a HTTP request, in particular request M201 may be a request for a web page, but in principle it can be a request for any content C.

In return, server 120 generates (step P201) and sends response M203 to terminal 10. Response M203 comprises most preferably a
15 Hypertext Markup Language (HTML) page that comprises instructions to set up a WebSocket WS between terminal 10 and server 120, and instructions to store referenced objects B1, B2, B3, ... at terminal 10. Terminal 10 sets up the WebSocket WS in step P203. In addition, if the terminal 10 receives the
20 referenced objects B1, B2, B3, ... in at least one bundle, the response M203 may contain instructions to the terminal 10 to unbundle the referenced objects B1, B2, B3, ... from the bundle.

Server 120 sends a set of requests M205 to at least one web server 140 (or to any other remote server) and receives a set of
25 responses M207. It is to be understood that there may be one or more remote or web servers 140 that each receive at least one request M205.

Requests M205 are most preferably HTTP Requests and responses M207 are most preferably HTTP Replies. The responses M207 may
30 comprise at least one referencing object A and at least one but most preferably a number of referenced objects B1, B2, B3, ...

In our Example (cf. FIG 5), the referencing object A is the original web page and the referenced object B1 is object www.example.com/image.png. In the Example, the referencing object A and the referenced objects B1 make out the requested content C.

Example: The original web page

```
<!DOCTYPE html>
<html lang="en">

10  <head>
      <meta charset="utf-8">
      <title>Cats Who Code demo</title>
    </head>

15  <body>
      <div>
          <header>
              <h1 id="logo"><a href="#">Cats Who
Code</a></h1>
20      </header>
      <!-- original line -->
          <img src='http://www.example.com/image.png' />
      <!-- original line -->
          <div>
25      <h2>Paris, France</h2>
          <p>A cat that can program computers is a
particularly interesting animal. For those who do not know
yet, also this text was written by the smart cat.</p>
          </div>
30      </div>
      </body>
    </html>
```

The referencing object A may be stored locally at the server 120 so that it does not necessarily need to be retrieved, in particular when the server 120 is a web server. If the server 120 is a proxy server, though, then the server 120 most probably
 5 needs to request also the referencing object A from the remote server 140 that may be a web server or any other server.

In step P205 server 120 parses the referencing object A, i.e. the main HTML page, and replaces all or at least some external links L1, L2, ... with internal links ML1, ML2, ... i.e. links
 10 with the "filesystem" prefix. The result is the modified referencing object AM.

In our Example, link L1 thus becomes modified link ML1: "http://www.example.com/image.png" is changed to "filesystem:http://www.example.com/temporary/image.png". We see
 15 that it suffices to add to the external link L1, L2, ... a suffice only to obtain the modified link ML1, ML2, ...; the suffice may in particular be as simple as the text "filesystem:" or any other suitable identifier.

Example (cont'd): The modified web page

```

20      <!DOCTYPE html>
      <html lang="en">

      <head>
          <meta charset="utf-8">
25      <title>Cats Who Code demo</title>
      </head>

      <body>
          <div>
30              <header>
                  <h1 id="logo"><a href="#">Cats Who
Code</a></h1>
                  </header>
          <!-- modified line -->
```

```

                    <img
src='filesystem:http://www.example.com/temporary/image.png'
/>
                    <!-- modified line -->
5                    <div>
                        <h2>Paris, France</h2>
                        <p>A cat that can program computers is a
particularly interesting animal. For those who do not know
yet, also this text was written by the smart cat.</p>
10                    </div>
                        </div>
                    </body>
                    </html>

```

In step P207 server 120 bundles some or all referenced objects
15 B1, B2, B3, ... The bundling can be carried out by using an
archiving program that do not use compression (such as TAR) or
that uses compression (such as ZIP).

Then server 120 sends the at least one bundle by using the
WebSocket WS to the terminal 10 (denoted by message M209). The
20 terminal 10 in step P209 receives the bundle and saves the
bundle locally, extracts the objects B1, B2, B3, ... from the
bundle and stores them locally in browser filesystem.

It must be understood that the bundling in step P207 is
optional. If the objects B1, B2, B3, ... are not stored in at
25 least one bundle, they are transmitted one by one from server
120 to terminal 10 as messages M209. We see that also in this
case the transmission is carried out by using the WebSocket WS.

After finishing the storing of the objects B1, B2, B3, ...,
terminal 10 sends notification M211 to server 120 that it has
30 finished the storing.

Upon receiving notification M211, server 120 sends (denoted by
message M213) the modified referencing object AM to terminal 10
as a HTTP response (cf. FIG 3).

For storing, the terminal 10 saves the referenced objects B1, B2, B3, ... in a filesystem at the terminal 10, such as in the temporary storage of the browser.

5 When the modified referencing object AM is received at the terminal 10, the modified links ML1, ML2, ML3, ... point to the referenced objects B1, B2, B3, ... stored locally in the filesystem of the terminal 10. In other words, the modified links ML1, ML2, ML3, ... give references for the browser at the
10 terminal 10 to locate the referenced objects B1, B2, B3, ... from the local filesystem of the terminal 10.

In step P211, the terminal 10 retrieves and renders the referenced objects B1, B2, B3, ... based on the modified referencing object AM. One sees that the result is that the
15 content C requested by the terminal 10 with message M201 is reproduced at terminal 10 but in such a manner that the referencing object A is received at terminal 10 only after the referenced objects B1, B2, B3, ... were first loaded at the terminal 10 via the WebSocket WS.

20 To summarize, FIG 2 illustrates the principle in more detail: the server 120 retrieves, in addition to the referencing object A that may optionally be retrieved, the referenced objects B1, B2, B3, ... from at least one remote server 140 by using the HTTP protocol. Then the server 120 transmits the referenced
25 objects B1, B2, B3, ... over the WebSocket WS to the terminal 10.

In our implementation of the server 120 and method according to the invention, all instructions from the server 120 to client are sent in response M203, which contains the instructions of

30 1) to set up a WebSocket WS following the WebSocket standard

Example javascript code to set up WebSocket:

```
var websocket = new  
WebSocket("ws://192.168.0.1:8888")
```

2) to set up filesystem following the draft standard of
5 FileSystem.

Example javascript code to set up FileSystem:

```
window.requestFileSystem(window.TEMPORARY, 1024*1024,  
function(filesystem)
```

3) to receive web objects via WebSocket WS and store them
10 in the memory allocated by browser

Example in pseudo code:

```
// when one object is received via websocket WS, it  
is written into FileSystem
```

```
15 websocket.onmessage = function (evt) {  
  
    // Create a FileWriter object for our  
    FileEntry (fileName).  
  
    fs.root.getFile(fileName, {create: true},  
function(fileEntry) {  
  
20     fileEntry.createWriter(function(fileWriter) {  
  
        fileWriter.onwriteend = function(event) {  
  
            debug('Write completed.');  
        };  
  
25     var writer = new BlobBuilder();  
  
    writer.append(content);
```

```
fileWriter.write(writer.getBlob(type));  
  
    }, errorHandler);  
  
};
```

5 4) when all referenced objects B1, B2, B3, ... are stored
in filesystem, the code generates notification M211, which
is for informing the server 120 to send the modified
referencing object AM.

10 **Example javascript code to sent notification M211
that is basically a HTTP request appended with a
special string "//filesystem//":**

```
window.location.href =  
'http://www.example.com//filesystem//';)
```

15 The functionality of the instructions in response M203 is like
the client software and all the necessary functions are just
reinterpreted in javascript.

Instead of installing client-side software in the operating
system of terminal 10, in our implementation the functions are
installed on-the-air. Since all referenced objects B1, B2, B3,
... are already inside the browser and modified links ML1, ML2,
20 ML3, ... already give enough indication for the browser to find
where these referenced objects B1, B2, B3, ... are, modified
referencing object AM only needs to contain the referencing
object A (original HTML page) wherein links L1, L2, L3, ... have
been replaced with modified links ML1, ML2, ML3,

25 The invention is not to be understood to be limited in the
attached patent claims but must be understood to encompass all
their legal equivalents.

Claims:1. A server (120) **configured to:**

- 5 a) in response to receiving a request (M201) for content (C) from a terminal (10), to respond (P201) to the terminal (10) with a response (M203) containing instructions configured to set up a bi-directional communication channel (WS) at the terminal (10) for communication between the server (120) and the terminal (10);
- 10 b) to load (M205, M207) the content comprising at least one referencing object (A) and a plurality of referenced objects (B1, B2, B3, ...) belonging to said content (C);
- 15 c) to generate a modified referencing object (AM) by replacing in the referencing object (A) such links (L1, L2, L3, ...) to referenced objects (B1, B2, B3, ...) that are external links with modified links (ML1, ML2, ML3, ...) that point to a locally stored version of the same object (B1, B2, B3, ...);
- 20 d) to send the referenced objects (B1, B2, B3, ...) to the terminal (10) over the bi-directional communication channel (WS); and
- 25 e) to send the modified referencing object (AM) to the terminal (10), in such a manner that the modified links (ML1, ML2, ML3) in the modified referencing object (AM) point to referenced objects (B1, B2, B3) already sent to the terminal (10).

2. A server (120) according to claim 1, **wherein:** the server (120) is configured to send the modified referencing object (AM) to the terminal (10) only after the server (120) has sent the
- 30 referenced objects (B1, B2, B3, ...) to the terminal (10).

3. A server (120) according to claim 1, **wherein:** the server (120) is configured to set up the bi-directional communication channel as a web socket.

4. A server (120) according to any one of claims 1 to 3,
5 **wherein:** the referencing object (A) is a hypertext markup language page and the referenced objects (B1, B2, B3, ...) are web objects referred to in said hypertext language page.

5. A server (120) according to any one of the preceding claims,
10 **wherein:** the server (120) is configured to send the referenced objects (B1, B2, B3, ...) to the terminal (10) in at least one bundle, most preferably after having sent instructions (M203) to set up a WebSocket (WS) and instructions to save the referenced objects (B1, B2, B3, ...) locally at the terminal (10).

6. A method for operating a server (120), comprising the steps
15 of:

a) in response to receiving a request (M201) for content (C) from a terminal (10), responding to the terminal (10) with a response (M203) containing instructions configured to set up a bi-directional communication channel (WS) at the
20 terminal (10) for communication between the server (120) and the terminal (10);

b) to load (M205, M207) the content comprising at least one referencing object (A) and a plurality of referenced objects (B1, B2, B3, ...) belonging to said content (C);

25 c) generating a modified referencing object (AM) by replacing in the referencing object (A) such links (L1, L2, L3, ...) to referenced objects (B1, B2, B3, ...) that are external links with modified links (ML1, ML2, ML3, ...) that point to a locally stored version of the same object (B1, B2,
30 B3, ...);

d) sending the referenced objects (B1, B2, B3, ...) to the terminal (10) over the bi-directional communication channel (WS); and

5 e) sending the modified referencing object (AM) to the terminal (10), in such a manner that the modified links (ML1, ML2, ML3, ...) in the modified referencing object (AM) point to referenced objects (B1, B2, B3, ...) already sent to the terminal (10).

7. A method according to claim 6, **wherein:** the step e) is
10 performed only after the step d) has been performed.

8. A method according to claim 6 or 7, **wherein:** the bi-directional communication channel is a WebSocket (WS).

9. A method according to any one of claims 6 to 8, **wherein:** the referencing object (A) is a hypertext markup language page and
15 the referenced objects (B1, B2, B3, ...) are web objects referred to in said hypertext language page.

10. A method according to any one of the preceding claims 6 to 10, **wherein:** the referenced objects (B1, B2, B3, ...) are sent to the terminal (10) in at least one bundle, most preferably
20 after having sent instructions (M203) to set up a WebSocket (WS) and instructions to save the referenced objects (B1, B2, B3, ...) locally at the terminal (10).

11. A system comprising at least one server (120) according to any one of the claims 1 to 5 and a plurality of terminals (10)
25 that preferably are mobile terminals, **wherein:** the system is configured to use the at least one server (120) for responding to requests (M201) for content that have been sent by the terminals (10).

Abstract

A server, a method for operating a server and a system

5 A server (120) is configured to:

- a) in response to receiving a request (M201) for content (C) from a terminal (10), to respond (P201) to the terminal (10) with a response (M203) containing instructions configured to set up a bi-directional communication channel (WS) at the terminal (10) for communication between the server (120) and the terminal (10);
- b) to load (M205, M207) the content comprising at least one referencing object (A) and a plurality of referenced objects (B1, B2, B3, ...) belonging to said content (C);
- 15 c) to generate a modified referencing object (AM) by replacing in the referencing object (A) such links (L1, L2, L3, ...) to referenced objects (B1, B2, B3, ...) that are external links with modified links (ML1, ML2, ML3, ...) that point to a locally stored version of the same object (B1, B2, B3, ...);
- 20 d) to send the referenced objects (B1, B2, B3, ...) to the terminal (10) over the bi-directional communication channel (WS); and

to send the modified referencing object (AM) to the terminal (10), in such a manner that the modified links (ML1, ML2, ML3) in the modified referencing object (AM) point to referenced objects (B1, B2, B3) already sent to the terminal (10).

The patent application comprises also parallel independent claims for a method and a system.

30 FIG 2

FIG 1

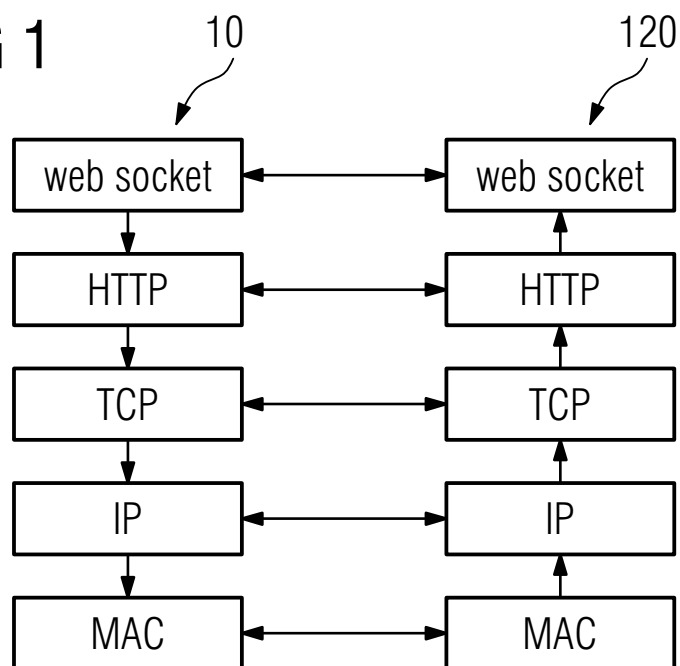


FIG 2

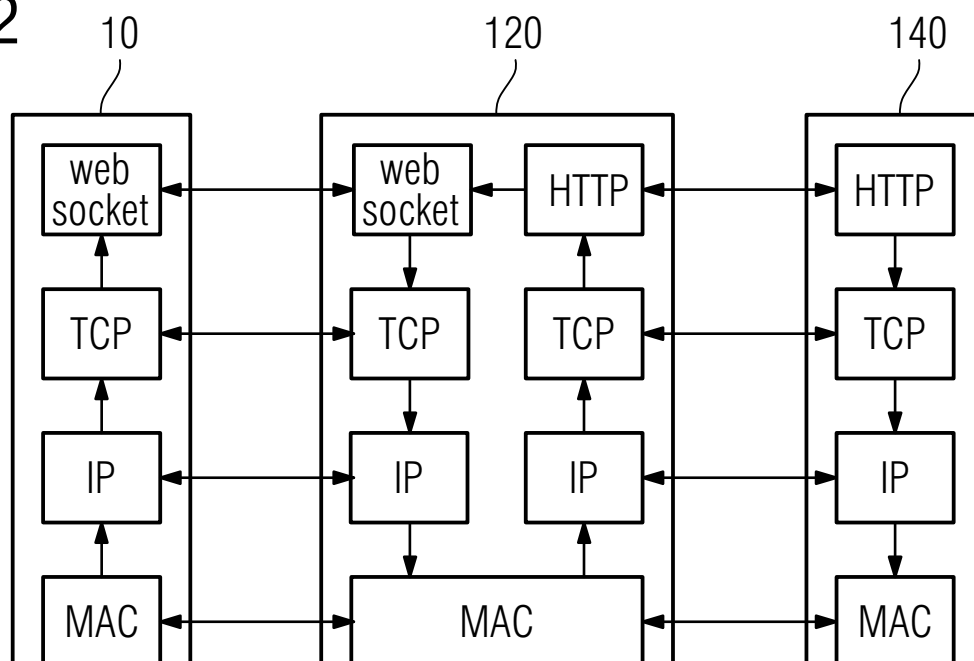


FIG 3

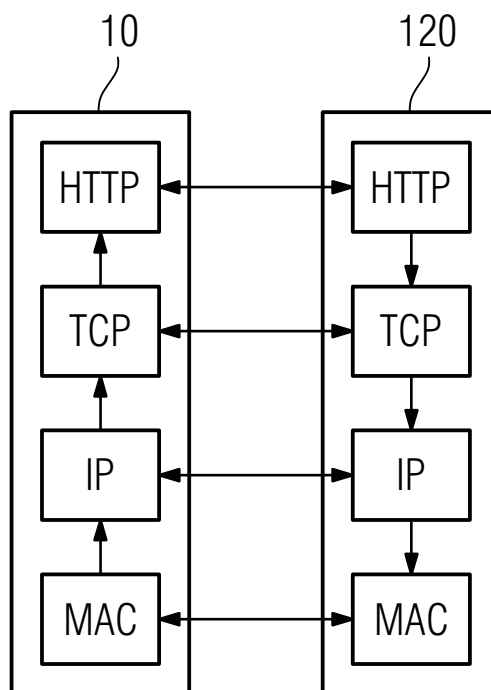


FIG 4

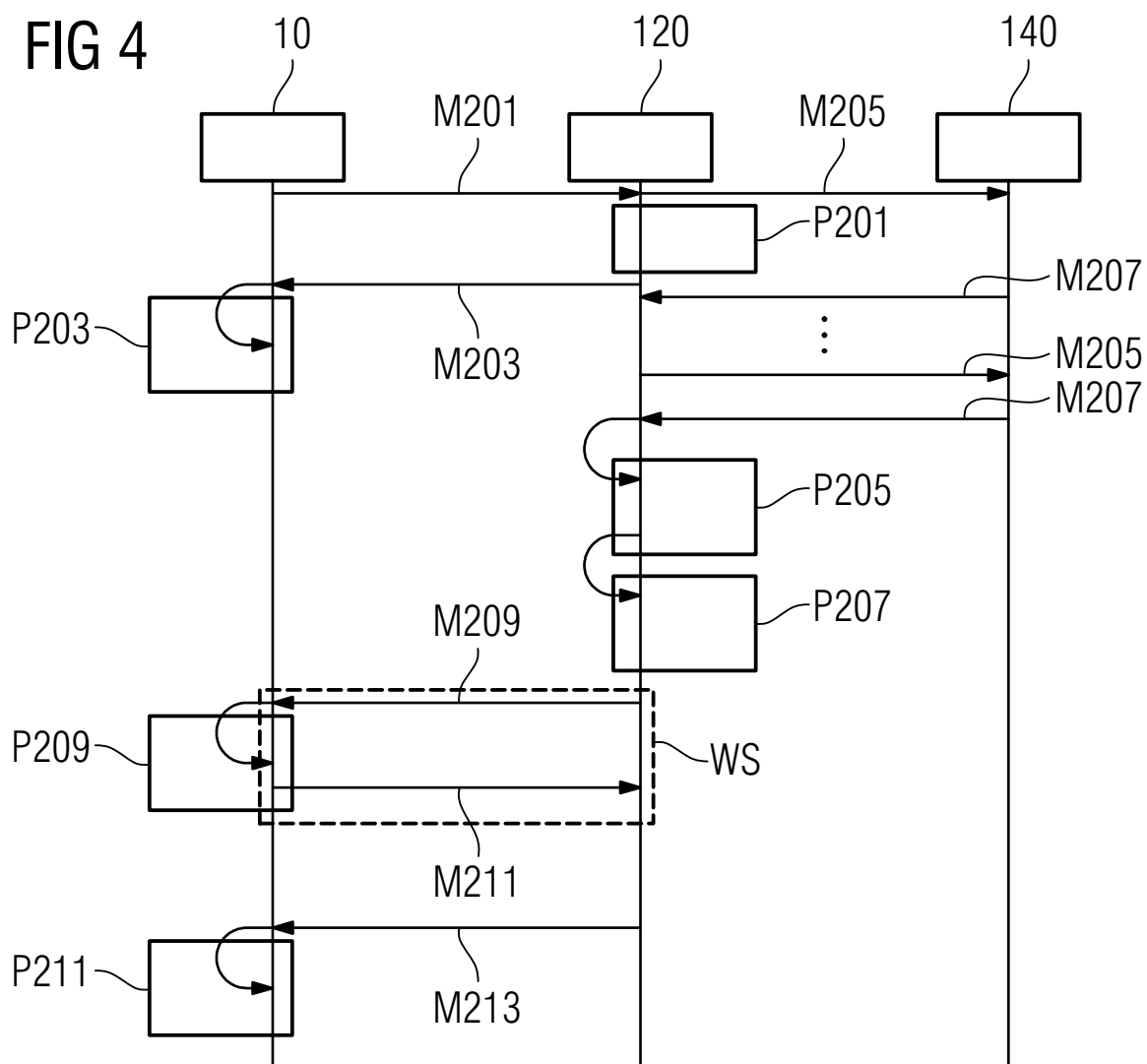


FIG 5

