



- (51) **International Patent Classification:**
H04L 12/801 (2013.01) *H04L 12/707* (2013.01)
- (21) **International Application Number:**
PCT/EP2016/051561
- (22) **International Filing Date:**
26 January 2016 (26.01.2016)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (71) **Applicant:** NEC EUROPE LTD. [DE/DE]; Kurfürsten-Anlage 36, 69115 Heidelberg (DE).
- (72) **Inventors:** BIFULCO, Roberto; Bergheimerstraße 38, 69115 Heidelberg (DE). KUENZER, Simon; Prinz-Eugen-Str. 5, 68735 Waghäusel (DE).
- (74) **Agent:** ULLRICH & NAUMANN; Schneidmühlstraße 21, 69115 Heidelberg (DE).
- (81) **Designated States** (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY,

BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

- (84) **Designated States** (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— with international search report (Art. 21(3))

- (54) **Title:** METHOD AND SERVER FOR ESTABLISHING A TCP CONNECTION

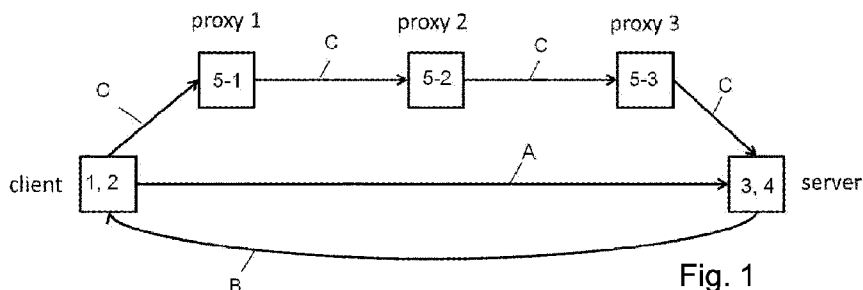


Fig. 1

- (57) **Abstract:** A method for establishing a TCP connection between a first end-point (1) and a second end-point (3) aims at reducing TCP end-to-end transfer time and thereby accelerating the data transfer. The method comprises: establishing a first TCP connection between said first end-point (1) and said second end-point (3), said second end-point (3) dynamically deciding on redirecting said first TCP connection via a chain of proxies (5) that interconnects said first and said second end-points (1; 3), in case of a redirection decision by said second end-point (3), said first end-point (1) establishing a new TCP connection with the first proxy (5-1) of said chain of proxies (5), and establishing a segmented TCP connection between said first end-point (1) and said second end-point (3) via said chain of proxies (5) and transferring data between said first end-point (1) and said second end-point (3) through said chain of proxies (5).



METHOD AND SERVER FOR ESTABLISHING A TCP CONNECTION

The present invention relates to a method for establishing a TCP connection
5 between a first end-point and a second end-point. Furthermore, the present
invention relates to a server being configured to establish a TCP connection with a
client.

TCP (Transmission Control Protocol) is a widely used communication service that
10 operates on top of the network layer of the Internet model and provides host-to-
host connectivity at the transport layer. TCP, which is a rate-adaptive protocol,
implements several control mechanisms for packet deliveries between hosts in a
communications network, among them flow control and congestion control
mechanisms.

15 The TCP flow and congestion control algorithm is dependent on the round trip time
(RTT) between the two parties involved in the communication (typically a client
and a server). When the RTT is large, the time it takes for TCP flow control to
converge to the actual available bandwidth grows. Currently, there are a few
20 solutions to address the issue.

For instance, a possibility is to fine-tune the TCP flow control algorithm for the
greater RTT. However, this requires manual configurations, and a careful selection
of the parameters, while managing those parameters on a per-connection basis.
25 Alternatively, Wide-Area-Network (WAN) optimizer can be deployed at the two
ends of the communication. The optimizers use normal TCP with the client and the
server, while using custom, WAN-optimized protocol configurations between them.
However, this solution has the disadvantage of deploying specialized devices at
both ends of the communication, requiring ad-hoc developed protocols between
30 them.

It is therefore an objective of the present invention to improve and further develop
a method for establishing a TCP connection and a server of the initially specified
types in such a way that a TCP acceleration with reduced end-to-end transfer

times is achieved, while at the same time the problems of prior art solutions mentioned above are being avoided.

In accordance with the invention, the aforementioned objective is accomplished by
5 a method for establishing a TCP connection between a first end-point and a second end-point, the method comprising:

establishing a first TCP connection between said first end-point and said second end-point,

10 said second end-point dynamically deciding on redirecting said first TCP connection via a chain of proxies that interconnects said first and said second end-points,

in case of a redirection decision by said second end-point, said first end-point establishing a new TCP connection with the first proxy of said chain of proxies, and

15 establishing a segmented TCP connection between said first end-point and said second end-point via said chain of proxies and transferring data between said first end-point and said second end-point through said chain of proxies.

Furthermore, the above objective is accomplished by a server, being configured to
20 establish a TCP connection with a client, the server comprising:

a receiving component being configured to receive a request for TCP connection establishment from a client,

25 an analyzing component being configured to determine and/or evaluate, based on said request for TCP connection establishment from said client (2), one or more variables – end-to-end performance variables – that may affect the end-to-end flow performance towards said client, and

decision means for dynamically deciding on redirecting a first TCP connection via a chain of proxies that interconnects the server with said client.

30 According to embodiments of the invention it has first been recognized that TCP acceleration with reduced end-to-end transfer times can be achieved by deploying one or more TCP proxies in the network and adopting a number of these TCP proxies on the end-to-end path of the connection to form a chain of TCP proxies. In this way, a segmented TCP connection between the first end-point and the

- second end-point is established via the chain of proxies and data between the first end-point and the second end-point can be transferred through this chain of proxies. According to embodiments of the invention the redirection of the first TCP connection to the chain of proxies is not applied mandatorily or consistently, but
- 5 the second end-point dynamically decides on such redirection, for instance on a case-by-case (e.g. flow-by-flow) basis. Consequently, the present invention enables a dynamic acceleration of TCP flows without requiring any change to the end-hosts TCP network stack.
- 10 The invention requires the availability of TCP proxies deployed in the middle of the path between the communication's end-points. The more (performance enhancing) TCP proxies are/can be deployed, the more effective the solution is. In case of MPTCP, the TCP proxies have to be able to proxy an MPTCP subflow.
- 15 The solution according to the present invention does not require any changes to the TCP configuration and does not introduce typical disadvantages of TCP tuning (e.g., buffer bloat, increased recovery cost in case of losses, etc.). It optimizes TCP of unmodified end-hosts, using explicit redirection of TCP flows through a chain of TCP proxies. Since the present invention does not require any
- 20 modifications, neither of the TCP configuration nor of the end-hosts (in particular, no special purpose network stacks are required at the end-hosts), the solution works also with L4+ encryption. Consequently, it can be suitably applied for improving (i.e. accelerating) TLS (Transport Layer Security) performance.
- 25 According to a preferred embodiment the first end-point may be a client and the second end-point may be a server, wherein the client contacts the server by sending a request to download a given resource, e.g. a particular data file, from the server, in accordance with a typical practical use case. For instance, the client may be a Web browser implemented on a desktop computer or on a tablet, and
- 30 the server may be a Web server. However, as will be easily appreciated by those skilled in the art, different client/server architectures may be used likewise.

Each proxy of the chain of proxies between the first end-point and the second end-point of the segmented TCP connection as well as the second end-point itself

function as end-points of the respective segment of the segmented TCP connection. According to a preferred embodiment it may be provided that each segment end-point of the segmented TCP connection, i.e. each proxy of the chain of proxies as well as the second connection end-point (e.g. server), runs a TCP
5 flow control algorithm independently from the other segment end-points of the segmented TCP connection. Advantageously, since in this case the RTT between the segment end-points is smaller than the RTT between the connection end-points, the convergence time of the individual TCP flow control algorithms of the segments is lower than the convergence time of a single (i.e. 'global') TCP flow
10 control algorithm applied on the end-to-end path.

According to a preferred embodiment the second end-point, upon being contacted by the first end-point, may determine and/or evaluate one or more variables – end-to-end performance variables – that may affect the end-to-end flow performance
15 between the first and the second end-points. In this regard it should be noted that an improvement on the flow transfer time may depend on a number of variables, which include the end-to-end round trip time (RTT) of the path between the first (e.g. client) and the second (e.g. server) end-point (RTT-CS), the bandwidth of the path between the end-points (BW-CS), the size (usually expressed in bytes) of the
20 data transferred by a flow (SIZE). As will be easily appreciated by those skilled in the art, there may be many additional variables (e.g. delay or the like) that may influence the flow transfer performance. However, for the sake of explanation clarity, the description hereinafter will generally focus on the mentioned three variables (RTT-CS, BW-CS, SIZE), and will use the general term 'end-to-end
25 performance variables' to refer to all the variables that may affect the end-to-end flow performance.

According to a preferred embodiment it may be provided that the first TCP connection between the first end-point and the second end-point is established by
30 the first end-point performing a three-way TCP or MPTCP (Multipath TCP) handshake with the second end-point. From this handshake the second end-point may already extract one or more of the end-to-end performance variables, for instance an estimation of the RTT-CS.

According to a preferred embodiment the above specified determination and/or evaluation of one or more of the end-to-end performance variables may form the basis for the second end-point to decide on whether to redirect or not an incoming network flow from the first end-point through the chain of proxies. Similarly,
5 according to embodiments explicit redirection of network flows through a chain of proxies may be performed based on the analysis of end-to-end network path's and data transfer's performance metrics.

According to a preferred embodiment the second end-point, after deciding to
10 perform a connection redirection, may send a redirection command to the first end-point, wherein the redirection command contains an indication of the first proxy of the chain of proxies, for instance by including the IP address and/or port number of the first proxy of the chain into the redirection command message.

According to a preferred embodiment the redirection command may contain an instruction for the first end-point to download a requested resource from the first proxy of the chain of proxies. Alternatively, in case of both parties supporting and applying MPTCP, it may be provided that the redirection command contains an instruction for the first end-point to add an MPTCP subflow through the chain of
20 proxies. In any case, it may be provided that the first end-point is configured to establish a new TCP connection with the first proxy in the chain of proxies upon reception of a redirection command from the second end-point.

In particular in case of applying MPTCP, according to a preferred embodiment it
25 may be provided that, while the additional MPTCP subflow (i.e. the segmented TCP connection) is being established via the chain of proxies, the second end-point of the connection (i.e. typically the server) starts already to transmit data through the original subflow (i.e. the first TCP connection). Once the new subflow is connected to the chain of proxies, the server may close the original subflow.

According to an embodiment the chain of proxies may be pre-established which
30 means, for instance, that the server (or the server's administrator) triggers the deployment of a number of proxies proactively, and it configures each proxy to retrieve a given resource from the next proxy in the chain, with the last one

configured to retrieve the resource from the server (i.e. the second endpoint of the first TCP connection). Alternatively, it may be provided that the chain of proxies is established dynamically, with each proxy autonomously deciding on the next hop of the chain for retrieving a given resource. Likewise, the chain's proxies may be
5 created proactively or dynamically, e.g. by booting virtual machines on the fly when the server receives a TCP connection request.

There are several ways how to design and further develop the teaching of the present invention in an advantageous way. To this end it is to be referred to the
10 dependent patent claims on the one hand and to the following explanation of preferred embodiments of the invention by way of example, illustrated by the drawing on the other hand. In connection with the explanation of the preferred
embodiments of the invention by the aid of the drawing, generally preferred
embodiments and further developments of the teaching will be explained. In the
15 drawing

Fig. 1 is a schematic view illustrating the general concept of TCP flow establishment in accordance with an embodiment of the present invention, and
20

Fig. 2 is a schematic view illustrating the establishment of an MP-capable TCP flow in accordance with an embodiment of the present invention.

25 Embodiments of the present invention as described hereinafter build on the observation that lowering the RTT between the end-points of a TCP connection makes the flow control algorithm's convergence time smaller.

For example, assuming a client downloads a 400 Kbit file from a server, over a
30 10 Mbit/s link, with 40 ms as round trip time. If the flow control algorithm would converge to the 10 Mbit/s bandwidth instantaneously, then the file would be downloaded in 40 ms (400 Kbit/10 Mbit). However, since TCP is a rate-adaptive protocol, the standard TCP's flow control algorithm slowly increases the connection speed depending on the received Acknowledgement (ACK) during the

connection. Assuming that the server starts sending the first 100 Kbit of a file and then doubles the amount of sent data after the reception of an ACK (e.g., as it happens during the TCP slow start phase) until convergence, i.e., the maximum bandwidth, is reached. That is, the sender will send 100 Kbit then wait for ACKs,
5 then it will send 200 Kbit and again wait for the ACKs, then it will send 400 Kbit and so on.

Considering that each increment happens after an ACK is received, and that ACKs are sent only upon reception of data, then, the server performs an increment in the
10 sending rate every 40 ms, i.e., every RTT. To send 400 Kbit under this assumption, the server would then take 3 RTT, sending the following amount of data at each transmission: 100 Kbit, 200 Kbit, 100 Kbit (It could send up to 400 Kbit but the file size is 400 Kbit in total and only 100 Kbit are left to send).

15 Thus, in the previous case it would take $3 \text{ RTT} = 40 \text{ ms} \times 3 = 120 \text{ ms}$ to transfer the 400 Kbit, which corresponds to an average transfer rate of 3.3Mbit/s.

Considering now the case in which the transfer happens over a link with 20 ms as RTT, the server would still require 3 RTT to send the data, but, this time, it would
20 take $3 \text{ RTT} = 20 \text{ ms} \times 3 = 60 \text{ ms}$ to transfer the 400 Kbit, which corresponds to an average transfer rate of 6,6 Mbit/s.

In most of the cases the RTT cannot be arbitrarily changed, in particular when it is mainly caused by the propagation delay (i.e., the time it takes for a signal to travel
25 from a point to another point). However, dividing the end-to-end connection in a number of segments guarantees that each of the segments has a smaller RTT between the segments' endpoints than the RTT between the connection's end-points. Thus, if each segment end-point runs a TCP flow control algorithm independently from the other segments end-points, then, the convergence time for
30 each segment is lower than the convergence time on the end-to-end path.

To achieve a segmented TCP connection and to thereby accelerate the transfer time of a TCP flow, it is assumed that a number of TCP proxies are deployed in

the network, and embodiments of the invention adopt a variable number of these TCP proxies on the end-to-end path of the connection, as shown in Fig. 1.

According to the illustrated embodiment, the general procedure for achieving a reduced TCP end-to-end transfer time is as follows: a first end-point 1, which is a client 2, initiates a TCP connection – first TCP connection, denoted ‘A’ in Fig. 1 – with a second end-point 3, which is a server 4. The server 4 dynamically takes the decision of redirecting the flow through a chain of TCP proxies 5 and communicates such decision to the client 2 (in Fig. 1, the respective message is denoted ‘B’). The client 2 establishes a connection with the first proxy 5-1 in the chain, which in turn is connected to the second proxy 5-2, which in turn is connected to the third proxy 5-3. The last proxy 5-3 in the chain is connected to the server 4. Once the client 2 is connected to the proxy chain, the TCP flow data transfer can start via this segmented TCP connection (denoted ‘C’ in Fig. 1).

Since the end-to-end performance variables change for each flow, in the moment in which a flow is first established, the knowledge of such variables may be unknown to the flow initiator, i.e., to the client 2 (it should be recalled that a TCP connection is always initiated by the client). For example, the client 2 may ignore the RTT-CS since no packets have been exchanged with the server 4 yet. The server 4 may instead have more chances to value the performance variables, since it may be the destination of several other flows, thus, it may use previous knowledge to infer the end-to-end performance variables. Without loss of generality, it is assumed in the embodiments described hereinafter that the client 2 always starts a new TCP flow directly with the server 4. Likewise, it is assumed that the server 4 is the end-point 3 that always decides if the transfer of data for such TCP flow should happen using a chain of TCP proxies 5. However, as will be easily appreciated by those skilled in the art, solutions that are similar to the ones described herein may be applied to the case in which the client 2 is the one deciding to use a chain of TCP proxies 5 to perform the data transfer on the basis of an accelerated TCP flow.

In more detail, the operations according to the embodiment illustrated in Fig. 1 unfold as follows:

1) Connection establishment:

In this phase the client 2 contacts the server 4, which learns the client-related information, such as some of the end-to-end performance variables. For instance
5 the client 2 may perform the three-way TCP handshake, from which the server 4 extracts, e.g., an estimation of the RTT-CS. Then, the client 2 may send a HTTP REQUEST (denoted 'A') to the server 4 (where HTTP is used on top of TCP), from which the server 4 is able to infer the SIZE of the data transfer (e.g., the HTTP REQUEST's URL points to a specific server's 4 resource). Although not explicitly
10 shown in Fig. 1, the server 4 may comprise a receiving component that is configured to receive the request for a TCP connection establishment (i.e. message 'A' in Fig. 1) from client 2. Furthermore, the server 4 may comprise an analyzing component that may be configured to determine and/or evaluate, based on the request for TCP connection establishment from client 2, one or more of the
15 end-to-end performance variables (i.e. the parameters that may affect the end-to-end flow performance towards the client 2).

2) Connection redirection:

The server 4, based on the end-to-end performance variables derived in the above
20 step 1) and/or inferred from previous knowledge, takes a decision to redirect (or not) the incoming TCP flow from the client 2. To this end, the server 4 may comprise decision means that may be configured to receive evaluation results of the end-to-end performance variables from the server's 4 analyzing component and to dynamically decide on the redirection based thereupon. If a proxy chain
25 should be used, the decision is communicated back to the client 2. For instance, in case of using HTTP, the server 4, based on the end-to-end performance variables, takes a decision to redirect the incoming HTTP request 'A' to a chain of TCP proxies 5. Thus, the server 4 sends to the client 2 a HTTP REDIRECT message (denoted 'B'), which informs the client 2 that the requested resource should be
30 downloaded from a different server, which is the chain's first proxy 5-1.

3) Chain establishment:

The client 2, upon reception of redirection command 'B', establishes a new connection 'C' with the proxy chain (i.e., with the first proxy 5-1 in the chain). Each

proxy 5 of the chain connects to the next proxy in the chain, with the exception of the last chain's proxy 5-3, which connects to the server 4. Here, it should be noted that the number of three proxies to be part of the chain of proxies has been arbitrarily chosen and that this number may be higher or lower.

5

The establishment of the chain could happen in different ways: According to one embodiment the chain is pre-established which means, for instance, the server 4 (or the server's administrator) triggers the deployment of a number of proxies 5 proactively, and it configures each proxy 5 to retrieve a given resource from the next proxy 5 in the chain, with the last proxy 5-3 configured to retrieve the resource from the server 4. According to another embodiment the chain is dynamically-established which means, for instance, the server 4 (or the server's administrator) triggers the deployment of a number of proxies 5 proactively, but it does not configure the proxies 5 in advance. Each proxy 5 autonomously decides if the next hop for requesting a given resource should be another proxy 5 (and which one) or the server 4.

4) Data transfer:

Once the connection 'C' through the chain of proxies 5 is established, data is transferred through the chain.

As will be easily appreciated by those skilled in the art each of the aforementioned phases can be performed in different implementations and adopting different techniques and/or technologies, without departing from the scope of the present invention.

Fig. 2 is a schematic view illustrating the establishment of an MP (Multipath)-capable TCP flow in accordance with an embodiment of the present invention. In this embodiment, the general procedure for achieving a reduced TCP end-to-end transfer time is as follows (like numerals denote like components as in Fig. 1): The client 2 establishes a MP-capable TCP connection with the server 4 and sends the requests ('A'). The server 4 takes the decision of redirecting the sessions through a chain of TCP proxies 5 and informs the client 2 to establish a new subflow with the first proxy 5-1 of the chain depending on this decision. The server 4 may

directly start with the data transfer. At the same time, the client 2 establishes the subflow with the first proxy 5-1 in the chain, which in turn is connected to the second proxy 5-2, which in turn is connected to the third proxy 5-3. The last proxy 5-3 in the chain is connected to the server 4 (again, it should be noted that the number of proxies 5 employed to be part of the chain of proxies 5 could be different, i.e. both higher and lower than 3). Once the new subflow is connected to the proxy chain, the server 4 may close the initial subflow.

In more detail, the operations of MPTCP subflow redirection through a chain of proxies according to the embodiment illustrated in Fig. 2 unfold as follows:

1) Connection establishment:

The client 2 performs the three-way MPTCP handshake, from which the server 4 extracts, e.g., an estimation of the RTT-CS. Then, the client 2 sends a HTTP REQUEST (denoted 'A' in Fig. 2), where HTTP is used on top of a multi-path TCP implementation (as described, e.g., in Internet Engineering Task Force (IETF), RFC6824: "TCP Extensions for Multipath Operation with Multiple Addresses", 2013). From this the server 4 is able to infer the SIZE of the data transfer (e.g., the HTTP REQUEST's URL points to a specific server's 4 resource).

2) Connection redirection:

The server 4, based on the end-to-end performance variables, takes a decision to redirect the HTTP connection 'A' to a TCP proxy chain. Thus, the server 4 instructs the client 2 to add an MPTCP subflow through the proxy chain. For this purpose, the server 4 passes the IP address and port number of the first proxy 5-1 of the chain to the client 2. For this purpose, the MPTCP ADD_ADDR option may be used (denoted 'B' in Fig. 2). Depending on the implementation, the server 4 may start replying with the HTTP REPLY followed by data through the original subflow already.

3) Chain establishment:

The client 2, upon reception of the subflow addition command 'B', establishes a new subflow with the proxy chain by performing a connection ('C') to the first proxy 5-1 of the chain. For this purpose, the MPTCP MP_JOIN option may be used

Again, the establishment of the chain could happen in different ways: According to one embodiment the chain may be pre-established, i.e. for instance, the server (or the server's administrator) triggers the deployment of a number of proxies 5 proactively, and it configures each proxy 5 to establish a connection to the next proxy 5 in the chain, with the last proxy 5-3 configured to connect to the origin server4. Alternatively, the chain may be dynamically-established, i.e. for instance, the server (or the server's administrator) triggers the deployment of a number of proxies 5 for the MPTCP subflow proactively, but it does not configure the proxies 5 in advance. Each proxy 5 autonomously decides if the next hop for connecting the subflow should be another proxy 5 (and which one) or the server4. Likewise, the chain's proxies 5 may be proactively created or dynamically created, e.g., booting virtual machines on the fly when a request comes.

4) Data transfer:

After the subflow 'C' through the proxy chain is established, the server 4 closes the original subflow (indicated at 'D'). The transfer of the HTTP REPLY followed by data is started or continued on the subflow 'C' through the chain.

Finally, to be noted that the embodiments described in connection with Figs. 1 and 2 can both be used at the same time if the MPTCP implementation supports falling back to standard TCP operation. The server 4 is then able to fall back to HTTP request/redirect/reply approach in the first phase when MPTCP is not supported by the client 2.

Many modifications and other embodiments of the invention set forth herein will come to mind the one skilled in the art to which the invention pertains having the benefit of the teachings presented in the foregoing description and the associated drawings. Therefore, it is to be understood that the invention is not to be limited to the specific embodiments disclosed and that modifications and other embodiments are intended to be included within the scope of the appended claims. Although specific terms are employed herein, they are used in a generic and descriptive sense only and not for purposes of limitation.

Claims

1. Method for establishing a TCP connection between a first end-point (1) and a second end-point (3), the method comprising:
 - 5 establishing a first TCP connection between said first end-point (1) and said second end-point (3),
said second end-point (3) dynamically deciding on redirecting said first TCP connection via a chain of proxies (5) that interconnects said first and said second end-points (1; 3),
10 in case of a redirection decision by said second end-point (3), said first end-point (1) establishing a new TCP connection with the first proxy (5-1) of said chain of proxies (5), and
establishing a segmented TCP connection between said first end-point (1) and said second end-point (3) via said chain of proxies (5) and transferring data
15 between said first end-point (1) and said second end-point (3) through said chain of proxies (5).
2. Method according to claim 1, wherein said first end-point (1) is a client (2), wherein said second end-point (3) is a server (4), and wherein said client (2)
20 contacts said server (4) requesting download of a given resource.
3. Method according to claim 1 or 2, wherein each proxy (5) of said chain of proxies (5) between said first end-point (1) and said second end-point (3) runs a TCP flow control algorithm independently from the other proxies (5) of said chain
25 of proxies (5).
4. Method according to any of claims 1 to 3, wherein said second end-point (3), upon being contacted by said first end-point (1), determines and/or evaluates one or more variables – end-to-end performance variables – that may affect the
30 end-to-end flow performance between said first and said second end-points (1; 3).
5. Method according to claim 4, wherein said end-to-end performance variables include the end-to-end round trip time, RTT, of the path between said first end-point (1) and said second end-point, the bandwidth of the path between

said first end-point (1) and said second end-point, the size of the data transferred by said network flow and/or delay.

- 5 6. Method according to any of claims 1 to 5, wherein said first TCP connection between said first end-point (1) and said second end-point (3) is established by said first end-point (1) performing a three-way TCP or MPTCP handshake with said second end-point.
- 10 7. Method according to any of claims 1 to 6, wherein said second end-point (3) decides on the redirection of an incoming network flow from said first end-point (1) through said chain of proxies based on the determination and/or evaluation of said one or more end-to-end performance variables.
- 15 8. Method according to any of claims 1 to 7, wherein said second end-point, after deciding to perform a connection redirection, sends a redirection command to said first end-point, wherein said redirection command contains an indication of the first proxy (5-1) of said chain of proxies (5).
- 20 9. Method according to claim 8, wherein said redirection command contains an instruction for said first end-point (1) to download a requested resource from the first proxy (5-1) of said chain of proxies (5) or to add an MPTCP subflow through said chain of proxies (5).
- 25 10. Method according to claim 7 or 8, wherein said first end-point (1) establishes the new TCP connection with the first proxy (5-1) in said chain of proxies (5) upon reception of a redirection command from said second end-point (3).
- 30 11. Method according to any of claims 1 to 10, wherein said second end-point (3) starts sending data to said first end-point (1) via said first TCP connection prior to the establishment of said segmented TCP connection.
12. Method according to any of claims 1 to 11, wherein said chain of proxies is pre-established, with each proxy (5) of said chain being configured to retrieve a

given resource from the next proxy (5) in said chain and with the last proxy (5) of said chain being configured to retrieve the resource from said second end-point (3).

5 13. Method according to any of claims 1 to 11, wherein said chain of proxies (5) is dynamically established, with each proxy (5) autonomously deciding on the next hop of said chain for retrieving a given resource.

10 14. Method according to any of claims 1 to 13, wherein the proxies (5) of said chain of proxies (5) are created and/or activated proactively or dynamically.

15. Server, being configured to establish a TCP connection with a client (2), the server (4) comprising:

15 a receiving component being configured to receive a request for TCP connection establishment from a client (2),

an analyzing component being configured to determine and/or evaluate, based on said request for TCP connection establishment from said client (2), one or more variables – end-to-end performance variables – that may affect the end-to-end flow performance towards said client (2), and

20 decision means for dynamically deciding on redirecting a first TCP connection via a chain of proxies (5) that interconnects the server (4) with said client (2).

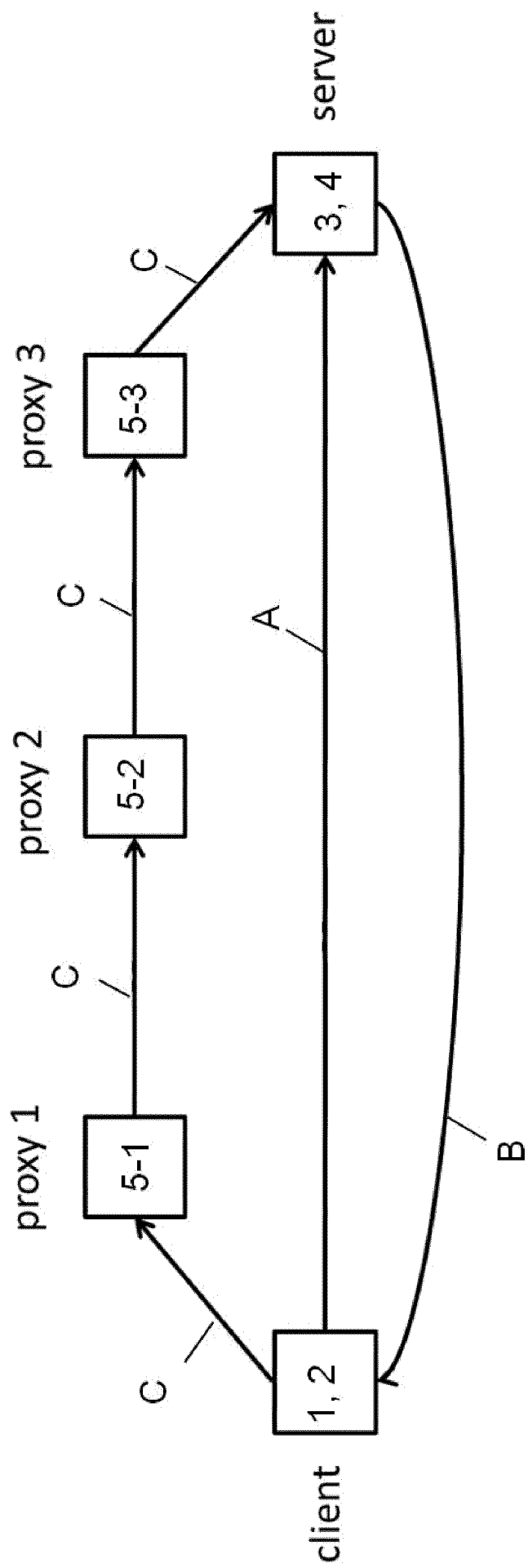


Fig. 1

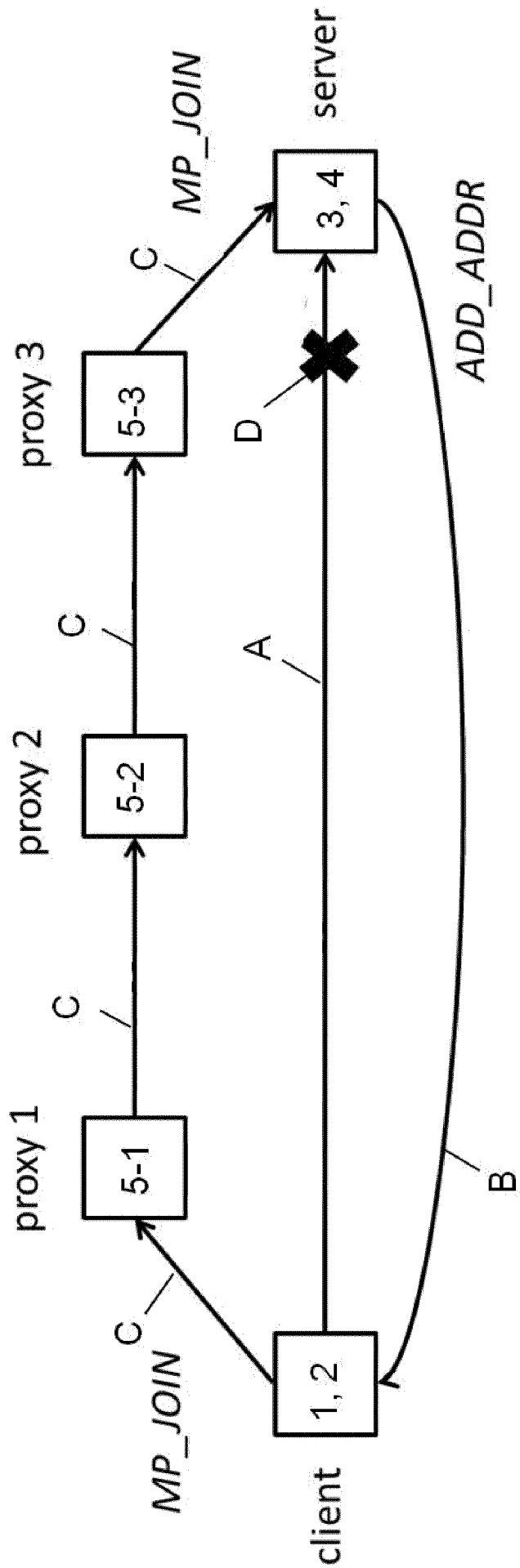


Fig. 2

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2016/051561

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L12/801 H04L12/707
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2005/188101 A1 (LUDWIG REINER [DE] ET AL) 25 August 2005 (2005-08-25) abstract paragraph [0001] - paragraph [0076] paragraph [0085] - paragraph [0086] paragraph [0092] - paragraph [0097] claims 1, 5-6, 8, 12, 15 paragraph [0100] - paragraph [0103] figures 1-2, 4-6	1-15
A	US 2014/036674 A1 (AGRAWAL DAKSHI [US] ET AL) 6 February 2014 (2014-02-06) abstract paragraph [0001] - paragraph [0012] paragraph [0019] - paragraph [0029] claims 1, 4, 6, 9 figures 1-4 ----- -/-	1-15



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

14 March 2016

Date of mailing of the international search report

23/03/2016

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Schrembs, Gerd

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2016/051561

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2008/235382 A1 (MARWAH MANISH [US] ET AL) 25 September 2008 (2008-09-25) abstract paragraph [0002] - paragraph [0010] paragraph [0013] - paragraph [0026] claims 1, 7, 14 figure 1 -----	1-15
A	US 2015/312383 A1 (ROELAND DINAND [SE] ET AL) 29 October 2015 (2015-10-29) abstract paragraph [0001] - paragraph [0030] paragraph [0056] - paragraph [0070] claims 1-3, 14, 17, 20 figures 1-3, 13-16 -----	1-15

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2016/051561

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2005188101 A1	25-08-2005	AT 471013 T AU 2002309186 A1 EP 1497955 A1 US 2005188101 A1 WO 03090417 A1	15-06-2010 03-11-2003 19-01-2005 25-08-2005 30-10-2003
US 2014036674 A1	06-02-2014	CN 103634288 A US 2014036674 A1	12-03-2014 06-02-2014
US 2008235382 A1	25-09-2008	NONE	
US 2015312383 A1	29-10-2015	CN 104854837 A EP 2932675 A1 KR 20150089057 A US 2015312383 A1 WO 2014090335 A1	19-08-2015 21-10-2015 04-08-2015 29-10-2015 19-06-2014