

Appendix A-1: Written Description Support for Proposed, Substitute Claims

Claim I.D.	Claim Limitation	Exemplary Support in Provisional App. No. 61/870,815	Exemplary Support in Nonprovisional App. No. 14/468,863
29 pre, pt. 1	29. (proposed substitute for claim 15) 45 . A method for fetching a [fresh] content over the Internet[, requested by a first device, identified in the Internet by a first identifier,] from a first server identified in the Internet by a second identifier via a group of multiple [tunnel] devices, each identified in the Internet by an associated group device identifier, ...	<p>See, e.g., Fig. 58 and associated description on Page 2 : “Figure 58 is a diagram of how Tunnels can be used by a Client. The "Tunnel" 16 is used by the Client 2 to fetch data from the Data Server 5, when such data is not found in the acceleration network's elements Caches, or the data is found in the caches but does not make sense for the Client 2 to fetch it for practical reasons, such as because the time to fetch it from the acceleration network's caches would be longer than to fetch the data directly from the Data Server 5. In such a case where it is beneficial for the Client 2 to fetch the data directly from the Data Server 5, the Client 2 may use one or more Tunnels 16 to carry out the request to the Data Server 5 on its behalf.”</p> <p>See, e.g., Figure 55 and associated description on Page 1: “Figure 55 describes how an element takes on one of the roles that it can assume: 5506, 5508 and 5510 show that when the system is starting up, a connection is established with the Acceleration Server 1, and a list of Tunnels for this communication device</p>	<p>See, e.g., original claim 134</p> <p>See, e.g., Figs. 5, 5a, and 5b; see also Figs. 11, 11a, 11b</p> <p>E.g., Page 81: “A method is disclosed for fetching a content over the Internet by a first device identified in the Internet by a first identifier, from a first server identified in the Internet by a second identifier [v]ia a group of multiple devices, each identified in the Internet by an associated group device identifier, ...”</p> <p>E.g., Abstract: “A method for fetching a content from a web server to a client device is disclosed, using tunnel devices serving as intermediate devices.... The tunnel devices in turn fetch the slices from the data server, and send the slices to the client device, where the content is reconstructed from the received slices... The partition into slices may be overlapping or non-overlapping, and the same slice (or the whole content) may be fetched via multiple tunnel devices.”</p>

		<p>is requested. Once received, in 5510 the device establishes connections (e.g. TCP connect) to each of the Tunnels in the list, so that once messages are needed to be sent to/from these Tunnels, it can be done quickly due to this pre-connection.”</p> <p>See, e.g., Fig. 50 and associated description on Page 1: “In the network of figure 50, there is a new type of element called a Tunnel.”</p> <p>E.g., Page 2: “...the Tunnels receives a data request from a Client, including the data server to get the request from and the associated other data required for such a data request, and fetches this from the data server and provides back to the Client[.]”</p> <p>See, e.g., Fig. 69 and associated description on Page 7: “Figure 69 describes the PATH_DIRECT- i.e. how to retrieve the data from the Tunnels 16 when this option is chosen.”</p> <p>See, e.g., Fig. 69 and associated description on Page 7: “In 6902, the requested data is split in to ranges, such that there is one range per participating Tunnel. It is possible to simply choose ranges of equal size to implement this. A possibly more efficient implementation is to provide each Tunnel a range that is</p>	<p>E.g., Page 82: “In one example, the second device may consist of, comprise, or be part of, a tunnel device, such as the tunnel device #1.”</p> <p>E.g., Page 118: “A flowchart 100 relating to the client device #1 31 a when employing three tunnel devices is shown in FIG. 10, based on the flowchart 60 described above. Upon receiving a list of available tunnel in a ‘Receive Tunnels List’ step 62 b from the Acceleration server 32, the client device #1 31 a selects multiple tunnels from the received list, rather than selecting a single tunnel as described in the ‘Select Tunnel’ step 62 c described above.”</p> <p>E.g., Page 114: “The request is received at the tunnel device #1 33 a at a ‘Request Received’ state 54 c, corresponding to a ‘Receive Content Request’ 73 b in the flowchart 70). In response, the tunnel device 33 a sends a ‘Content Request’ message 56 h to the data server #1 22 a (corresponding to a ‘Send Request To Server’ step 73 c), requesting the content that was requested by the client device #1 31 a.”</p> <p>E.g., Page 118: “In turn the data server #1 replies and sends the content in a ‘Send Content’ message 111 c (corresponding to the ‘Send Content’ message 56 i in the timing chart 50) to the requesting tunnel</p>
--	--	--	--

		<p>proportional to that Tunnel's ability to provide the data (for example, based on that Tunnel's upstream and downstream traffic to the Client, as measured in previous communications between them).”</p> <p>See also, e.g., Fig. 70 and associated description on Pages 7-8: “Figure 70 shows an example of how a tunnel may be implemented. The Tunnel receives a request (7002) from another network element (e.g. a Client) which includes the standard information of a data request (DR) including the address of the data server (such as the name or IP of a web server), the information requested (such as a URL), the range of the data that is requested (e.g. all of the data, or only a certain byte range, for example bytes 1000 to 1020), and additional information about the request (such as the cookies in HTIP). The Tunnel the[n] creates a request (7004) to the data server based on the DR and sends it to the data server. Once the response is received (7006), the Tunnel sends the data (7010) back to the requesting Client.”</p>	<p>device #1 33 a, which in turn forward the fetched content to the asking client device #1 31 a using a ‘Send Content’ message 111 d (corresponding to the ‘Send Content’ message 56 j in the timing chart 50).”</p> <p>See also, e.g., Page 73: “A client device may be a first device identified in the Internet by a first identifier, executing a method for fetching over the Internet a first content, identified by a first content, from a second server identified in the Internet by a third identifier, via a second device identified in the Internet by a second identifier, using a first server.”</p>
29 pre, pt. 2	the method comprising the step of partitioning the [fresh] content into a plurality of content slices, each content slice containing at least part of	E.g., Page 2: “The Client 2 will also determine how much of the data it will request from each of the Tunnels 16 based on their speed, and thus assign a ‘range’ of the data (for example for a data of 1200	<p>See, e.g., original claim 134</p> <p>See, e.g., Fig. 10, 11, 11a, 11b and 21</p>

	<p>the [fresh] content, and identified using a content slice identifier, and for each of the content slices, comprising the steps of:</p>	<p>bytes where two agents A and Bare selected, a range assignment may be that A is assigned the bytes between byte 0 and byte 856, and B is assigned bytes 857 to 1200). The Client 2 will then send the request to all these Tunnels 16 along with the requested range for them to fetch from the Data Server.”</p> <p>E.g., Page 2: “...where a range is also provided to the Tunnel, and the Tunnel only fetches this range of data from the data server.”</p> <p>See, e.g., Fig. 69 and associated description on Page 7: “In 6904 the data request is sent to each of the tunnels along with the <i>segment</i> that this Tunnel needs to fetch from the Data Server.” (emphasis added)</p> <p>E.g., Page 23: “Data servers typically limit the number of active connections that a host can keep open with them concurrently. This limitation is detrimental in various scenarios. One such scenario as an example is where a communications device with a web browser needs to load <i>many URLs from a Data Server in order to display a web page...</i>” (emphasis added)</p>	<p>E.g., Page 81: “...the method comprising the step of partitioning the content into a plurality of content slices, each content slice containing at least part of the content, and identified using a content slice identifier, and for each of the content slices.”</p> <p>E.g., Page 125: “The content requested by the client device #1 31 a may be partitioned into multiple parts or ‘slices’. Any number of slices may be used...”</p> <p>E.g., Page 138: “The content stored in a data server, such as the data server #1 22a, which may be requested by a client device such as the client device #1 201a, may be partitioned into multiple parts or ‘slices’. Any number of slices may be used. The slicing may be in a bit, nibble (4-bits), byte (8-bits), word (multiple bytes), character, string, or file level. For example, in a case wherein the content includes 240 bytes designated byte #1 to byte #240, using a byte level partitioning into two slices results in a first slice (slice #1) including byte #1 to byte #120, and a second slice (slice #2) including byte #121 to byte #240.</p> <p>E.g., Page 138: “In one example, the content itself is made of inherent or identifiable parts or segments, and the partition may make use of these parts. In one example, the content may be a website content composed of</p>
--	---	--	---

		<p>See, e.g., Fig. 136 and associated description on Page 23: “Figure 136 describes such a system. In it, the Client 2 would like to open more connections with the Data Server 5 than the Data Server 5 will allow (<i>where each of these connections is loading one or more requests, where each request is a resource, a range of a resource, or multiple resources</i>). The Client then communicates with participating Tunnels 4 (i.e. network devices on which software is installed for this purpose), and requesting from each of these Tunnels to load <i>several of these requests</i>, such that the each of the participating elements (the Client and the Tunnels from which it requested the loads) does not open more connections vs. the Data Server 5 that the Data Server 5 allows.” (emphasis added)</p> <p>See, e.g., Fig. 73</p> <p>See also, e.g., Page 21: “a system that comprises of a software module, a database, where upon getting a uri as an input, the system looks up the domain of the URL within the database, retrieves the schema of the URL and can identify the identity of the video, and the offset to view within the video according to the schema[.]”</p>	<p>multiple webpages, and thus the partition may be such that each slice includes one (or few) webpages. Further, the partitioning may be sequential or non-sequential in the content.”</p>
--	--	---	---

		<p>See also, e.g., Page 2: "...wherein a Client chooses one or more Tunnels and a range per Tunnel, sends the request to these Tunnels, and receives back the information from the Tunnel that the Tunnel received from the Data Server[.]"</p> <p>See also, e.g., Page 2: "This approach has multiple benefits, including the ability to fetch multiple segments of this data in parallel (which may result in a faster load time) and also by creating multiple paths to the Data Server 5 (which may result in a more effective use of the network's resources)."</p> <p>See also Pages 33-34 ("Microchunks" section)</p>	
29a	(a) selecting a [tunnel] device from the group;	<p>E.g., Page 2: "Therefore, once the Client 2 has determined that it is not worthwhile to use the caches within the various acceleration systems' network elements, it will determine how many Tunnels 16 to use for this request based on the amount of data to fetch, and the speed of the available Tunnels 16. The Client 2 will also determine how much of the data it will request from each of the Tunnels 16 based on their speed, and thus assign a 'range' of the data (for example for a data of 1200 bytes where two agents A and Bare selected, a range assignment may be</p>	<p>See, e.g., original claim 134</p> <p>See, e.g., Figs. 5, 5a, 5b, and 6</p> <p>E.g., Page 81: "The method may comprise the steps of selecting a device from the group..."</p> <p>E.g., Page 113: "Based on pre-set criteria, a tunnel device (or multiple tunnel devices) is selected by the client device #1 31 a in a 'Tunnel Select' step 53 c (corresponding to a 'Select Tunnel' step 62 c in the flowchart</p>

		<p>that A is assigned the bytes between byte 0 and byte 856, and B is assigned bytes 857 to 1200). The Client 2 will then send the request to all these Tunnels 16 along with the requested range for them to fetch from the Data Server.”</p> <p>E.g., Page 7: “In 6901 the Client 2 chooses which of the Tunnels 16 that were allocated to it by the Acceleration Server 1 it will use for this current transaction.”</p> <p>See also, e.g., Page 7: “...alternative tunnel can be chosen by several methods; a random tunnel from the list of Tunnels for this client, or a Tunnel that is currently not in use, or a Tunnel that the Client has the highest BW/RTI ratio to, etc.”</p> <p>See also, e.g., Page 5: “[I]n a system where one or more network elements (tunnels) are used to load portions of data on behalf of another network element (a client), a method of choosing tunnels for the clients, where the network elements that are the closest to the Client are chosen as tunnels[.]”</p>	<p>60). For example, the tunnel device #1 33 a may be selected.”</p> <p>E.g., Page 76: “A distinct device may be selected for each content slice.”</p> <p>See also, e.g., Page 118: “Upon receiving a list of available tunnel in a ‘Receive Tunnels List’ step 62 b from the Acceleration server 32, the client device #1 31 a selects multiple tunnels from the received list, rather than selecting a single tunnel as described in the ‘Select Tunnel’ step 62 c described above.”</p> <p>See also, e.g., Page 118: “In the described example, three distinct tunnel devices are selected from the list, such as the tunnel device #1 33 a (as before), the tunnel device #2 33 b, and the tunnel device #3 33 c. The client device 31 a executes three pre-connection processes in a ‘Pre-Connection Tunnel #1’ step 64 a, a ‘Pre-Connection Tunnel #2’ step 64 b, and a ‘Pre-Connection Tunnel #3’ step 64 c (each corresponding to the ‘Pre-connection’ flow chart 64 above), followed by a ‘Content Fetch Tunnel #1’ step 65 a, a ‘Content Fetch Tunnel #2’ step 65 b, and a ‘Content Fetch Tunnel #3’ step 65 c, respectively (each corresponding to the ‘Content Fetch’ flow chart 65 above).”</p> <p>See also, e.g., Page 112: “Then the tunnel device #1 33 a sign in with the acceleration</p>
--	--	--	---

			server 32 in a step ‘Sign-in as Tunnel’ 71 b, which corresponds to a message ‘Sign In’ 56 a in the chart 50. The message comprises the device functionality as ‘tunnel’, and the device 33 a identification on the Internet 113, such as its IP address (for example 125.12.67.0).”
29b	(b) sending over the Internet a first request [from the first device] to the selected [tunnel] device using [a tunneling protocol and]the group device identifier of the selected [tunnel] device, the first request including the content slice identifier and the second identifier[, the first request identifies the source of the fresh content as the first server];	<p>See, e.g., Fig. 70 and associated description on Pages 7-8: “Figure 70 shows an example of how a tunnel may be implemented. The Tunnel receives a request (7002) from another network element (e.g. a Client) which includes the standard information of a data request (DR) including the address of the data server (such as the name or IP of a web server), the information requested (such as a URL), the range of the data that is requested (e.g. all of the data, or only a certain byte range, for example bytes 1000 to 1020), and additional information about the request (such as the cookies in HTTP). The Tunnel the[n] creates a request (7004) to the data server based on the DR and sends it to the data server. Once the response is received (7006}, the Tunnel sends the data (7010) back to the requesting Client.”</p> <p>See, e.g., Fig. 58 and associated description on Page 2 : “Figure 58 is a diagram of how Tunnels can be used by a</p>	<p>See, e.g., original claim 134</p> <p>See, e.g., Figs. 5, 5a, 5b, and 7</p> <p>E.g., Page 81: “...the first device sending a first request to the selected device using the selected device identifier, the first request including the content slice identifier and the second identifier...”</p> <p>E.g., Page 114: “Once the client device #1 31 a determines that external content from a data server is required, as shown in a ‘Content Required’ state 53 d, a ‘Content Request’ message 56 g (shown in the messaging chart 50) is sent (corresponding to a ‘Send Content Request’ step 63 b in the flowchart 60) to the selected tunnel device #1 33 a.”</p> <p>E.g., Pages 117-118: “A ‘Content Request’ message 111 a (corresponding to the ‘Content Request’ message 56 g in the timing chart 50) is first sent from the client device #1 31 a to the selected tunnel device</p>

		<p>Client. The "Tunnel" 16 is used by the Client 2 to fetch data from the Data Server 5, when such data is not found in the acceleration network's elements Caches, or the data is found in the caches but does not make sense for the Client 2 to fetch it for practical reasons, such as because the time to fetch it from the acceleration network's caches would be longer than to fetch the data directly from the Data Server 5. In such a case where it is beneficial for the Client 2 to fetch the data directly from the Data Server 5, the Client 2 may use one or more Tunnels 16 to carry out the request to the Data Server 5 on its behalf."</p> <p>See, e.g., Fig. 69 and associated description on Page 7: "Figure 69 describes the PATH_DIRECT- i.e. how to retrieve the data from the Tunnels 16 when this option is chosen."</p> <p>E.g., Page 1: "Once received, in 5510 the device establishes connections (e.g. TCP connect) to each of the Tunnels in the list, so that once messages are needed to be sent to/from these Tunnels, it can be done quickly due to this pre-connection."</p> <p>See, e.g., Figs. 136 and 138 and associated description on Page 23: "In it, the Client 2 would like to open more connections with the Data Server 5 than the Data Server 5</p>	<p>#1 33 a, which responds by forwarding the request to the data server #1 22 a using a 'Content Request' message 111 b (corresponding to the 'Content Request' message 56 h in the timing chart 50)."</p> <p>E.g., Page 145: "A content, such as an URL (or a web-page, or a web-site) which is typically stored in a data server, such as the data server #1 22 a, may be requested by the client device, such as the client device 201 a, as shown in a state 'Content Needed' 223 b in the chart 220. The client device sends a 'Request List' message 226 e to the acceleration server 202, corresponding to a 'Request Agents List' step 231 c in the flowchart 230. This request includes the URL or any other identifier of the requested content."</p> <p>E.g., Pages 34-35: "Layer 3 (Network Layer) and lower layer encryption based protocols include IPsec, L2TP (Layer 2 Tunneling Protocol) over IPsec, and Ethernet over IPsec. The IPsec is a protocol suite for securing IP communication by encrypting and authenticating each IP packet of a communication session."</p> <p>E.g., Page 85: "Alternatively or in addition, the first device may be communicating with the second device using a VPN or a</p>
--	--	---	---

		<p>will allow (where each of these connections is loading one or more requests, where each request is a resource, a range of a resource, or multiple resources). The Client then communicates with participating Tunnels 4 (i.e. network devices on which software is installed for this purpose), and requesting from each of these Tunnels to load several of these requests, such that the each of the participating elements (the Client and the Tunnels from which it requested the loads) does not open more connections vs. the Data Server 5 that the Data Server 5 allows.”</p> <p>See also, e.g, Fig. 226 and associated description on Page 44: “It is thus useful to create a system or a method that would allow to always send the data from an available socket. Figure 226 is a diagram of a system that makes it possible to move data between sockets so that if data is waiting to be sent through a socket and during that time a different socket is ready for sending data, the data is moved to that other socket.”</p> <p>See also, e.g., Fig. 69 and associated description on Page 7: “In 6904 the data request is sent to each of the tunnels along with the segment that this Tunnel needs to fetch from the Data Server. The Tunnel</p>	<p>tunneling protocol, and the connection may be established using authentication.”</p> <p>E.g., Page 99: “Any communication between any two network elements may use TCP, and wherein the connection may be established by performing ‘Active OPEN’ or ‘Passive OPEN’, may use a VPN, or may use a tunneling protocol.”</p>
--	--	--	--

		will then make that request to the Data Server and return the information received to the Client.”	
29c	(c) in response to receiving the sent first request by the selected [tunnel] device, receiving over the Internet the content slice [from the first server via] from the selected [tunnel] device; and	<p>See, e.g., Fig. 70 and associated description on Pages 7-8: “The Tunnel the creates a request (7004) to the data server based on the DR and sends it to the data server. Once the response is received (7006), the Tunnel sends the data (7010) back to the requesting Client.”</p> <p>See, e.g., Fig. 58 and associated description on Page 2 : “Figure 58 is a diagram of how Tunnels can be used by a Client. The "Tunnel" 16 is used by the Client 2 to fetch data from the Data Server 5, when such data is not found in the acceleration network's elements Caches, or the data is found in the caches but does not make sense for the Client 2 to fetch it for practical reasons, such as because the time to fetch it from the acceleration network's caches would be longer than to fetch the data directly from the Data Server 5. In such a case where it is beneficial for the Client 2 to fetch the data directly from the Data Server 5, the Client 2 may use one or more Tunnels 16 to carry out the request to the Data Server 5 on its behalf.”</p> <p>See, e.g., Fig. 62 and associated description on Page 4: “6128 sends the</p>	<p>See, e.g., original claim 134</p> <p>See, e.g., Figs. 5 and 5b; see also Fig. 7</p> <p>E.g., Page 81: “...in response to receiving the first request, the selected device sending a second request to the first server using the second identifier, the second request including the content slice identifier; in response to receiving the second request, the first server sending the content slice to the selected device...”</p> <p>E.g., Page 114: “The request is received at the tunnel device #1 33 a at a ‘Request Received’ state 54 c, corresponding to a ‘Receive Content Request’ 73 b in the flowchart 70). In response, the tunnel device 33 a sends a ‘Content Request’ message 56 h to the data server #1 22 a (corresponding to a ‘Send Request To Server’ step 73 c), requesting the content that was requested by the client device #1 31 a.”</p> <p>E.g., Page 118: “In turn the data server #1 replies and sends the content in a ‘Send Content’ message 111 c (corresponding to the ‘Send Content’ message 56 i in the timing chart 50) to the requesting tunnel</p>

		<p>resulting data (received directly from the Data Server 5 through the Tunnels 16, or from the acceleration network through the Agents 4 and Peers 3, to the requesting Application.”</p> <p>See, e.g., Fig. 69 and associated description on Page 7: “Figure 69 describes the PATH_DIRECT- i.e. how to retrieve the data from the Tunnels 16 when this option is chosen.”</p> <p>See also, e.g., Fig. 69 and associated description on Page 7: “In 6904 the data request is sent to each of the tunnels along with the segment that this Tunnel needs to fetch from the Data Server. The Tunnel will then make that request to the Data Server and return the information received to the Client.”</p>	<p>device #1 33 a, which in turn forward the fetched content to the asking client device #1 31 a using a ‘Send Content’ message 111 d (corresponding to the ‘Send Content’ message 56 j in the timing chart 50).”</p> <p>E.g., Page 114: “The data server #1 22 a receives the request and prepares the requested content in a ‘Content Prepared’ state 55 a, and sends the requested content back to the tunnel device #1 33 a in a ‘Send Content’ message 56 i, received by the tunnel device #1 33 a in a ‘Receive Content from Server’ step 73 d. The received content is prepared in a ‘Content Prepared’ state 54 d, and then sent, in a ‘Send Content’ message 56 j (corresponding to a ‘Send Content To Client’ step 73 e), to the client device #1 31 a.”</p>
29d	[(d) in response to receiving the content slice, sending the content slice to the first device using the first identifier and;]	<p>See, e.g., Fig. 70 and associated description on Pages 7-8: The Tunnel the creates a request (7004) to the data server based on the DR and sends it to the data server. Once the response is received (7006}, the Tunnel sends the data (7010) back to the requesting Client.</p> <p>See, e.g., Fig. 58 and associated description on Page 2 : “Figure 58 is a diagram of how Tunnels can be used by a Client. The "Tunnel" 16 is used by the</p>	<p>See, e.g., Figs. 5, 5a, 5b, and 7</p> <p>E.g., Page 81: “...and in response to receiving the content slice, the selected device sending the content slice to the first device.”</p> <p>E.g., Page 114: “The request is received at the tunnel device #1 33 a at a ‘Request Received’ state 54 c, corresponding to a ‘Receive Content Request’ 73 b in the flowchart 70). In response, the tunnel device</p>

		<p>Client 2 to fetch data from the Data Server 5, when such data is not found in the acceleration network's elements Caches, or the data is found in the caches but does not make sense for the Client 2 to fetch it for practical reasons, such as because the time to fetch it from the acceleration network's caches would be longer than to fetch the data directly from the Data Server 5. In such a case where it is beneficial for the Client 2 to fetch the data directly from the Data Server 5, the Client 2 may use one or more Tunnels 16 to carry out the request to the Data Server 5 on its behalf.”</p> <p>See, e.g., Fig. 62 and associated description on Page 4: “6128 sends the resulting data (received directly from the Data Server 5 through the Tunnels 16, or from the acceleration network through the Agents 4 and Peers 3, to the requesting Application.”</p> <p>See, e.g., Fig. 69 and associated description on Page 7: “Figure 69 describes the PATH_DIRECT- i.e. how to retrieve the data from the Tunnels 16 when this option is chosen.”</p> <p>See also, e.g., Fig. 69 and associated description on Page 7: “In 6904 the data request is sent to each of the tunnels along</p>	<p>33 a sends a ‘Content Request’ message 56 h to the data server #1 22 a (corresponding to a ‘Send Request To Server’ step 73 c), requesting the content that was requested by the client device #1 31 a.”</p> <p>E.g., Page 118: “In turn the data server #1 replies and sends the content in a ‘Send Content’ message 111 c (corresponding to the ‘Send Content’ message 56 i in the timing chart 50) to the requesting tunnel device #1 33 a, which in turn forward the fetched content to the asking client device #1 31 a using a ‘Send Content’ message 111 d (corresponding to the ‘Send Content’ message 56 j in the timing chart 50).”</p> <p>E.g., Page 114: “The data server #1 22 a receives the request and prepares the requested content in a ‘Content Prepared’ state 55 a, and sends the requested content back to the tunnel device #1 33 a in a ‘Send Content’ message 56 i, received by the tunnel device #1 33 a in a ‘Receive Content from Server’ step 73 d. The received content is prepared in a ‘Content Prepared’ state 54 d, and then sent, in a ‘Send Content’ message 56 j (corresponding to a ‘Send Content To Client’ step 73 e), to the client device #1 31 a.”</p>
--	--	--	---

		with the segment that this Tunnel needs to fetch from the Data Server. The Tunnel will then make that request to the Data Server and return the information received to the Client.”	
29e	wherein the method further comprising[comprises] the step of constructing the [fresh] content from the received plurality of content slices, and	<p>See, e.g., Fig. 70 and associated description on Pages 7-8: “Figure 70 shows an example of how a tunnel may be implemented. The Tunnel receives a request (7002) from another network element (e.g. a Client) which includes the standard information of a data request (DR) including the address of the data server (such as the name or IP of a web server), the information requested (such as a URL), the range of the data that is requested (e.g. all of the data, or only a certain byte range, for example bytes 1000 to 1020), and additional information about the request (such as the cookies in HTTP). The Tunnel the[n] creates a request (7004) to the data server based on the DR and sends it to the data server. Once the response is received (7006), the Tunnel sends the data (7010) back to the requesting Client.”</p> <p>E.g., Page 21: “a system that comprises of a software module, a database, where upon getting a uri as an input, the system looks up the domain of the URL within the database, retrieves the schema of the</p>	<p>See, e.g., original claim 134</p> <p>See, e.g., Figs. 5 and 5b</p> <p>E.g., Page 81: “...receiving the content slice from the selected device; and constructing the content from the received content slices.”</p> <p>See, e.g., Fig. 21 and associated description on Page 139: “For example, multiple chunks may be combined to reconstruct the original content, such as website or content, as schematically shown in an arrangement 210 shown in FIG. 21.”</p> <p>E.g., Page 139: “In one example, the content is a website or a webpage, or may be identified as a URL, and consists of, or comprises, non-overlapping and equally-sized parts, referred to as chunks. For example, multiple chunks may be combined to reconstruct the original content, such as website or content.”</p> <p>E.g., Page 126: “It is noted that in such a partition, the content may be fully reconstructed from any two of the slices,</p>

		<p>URL and can identify the identity of the video, and the offset to view within the video according to the schema[.]”</p> <p>See, e.g., Fig. 62 and associated description on Page 4: “6128 sends the resulting data (received directly from the Data Server 5 through the Tunnels 16, or from the acceleration network through the Agents 4 and Peers 3, to the requesting Application.”</p> <p>See, e.g., Fig. 69 and associated description on Page 7: “Figure 69 describes the PATH_DIRECT- i.e. how to retrieve the data from the Tunnels 16 when this option is chosen.”</p> <p>See also, e.g., Fig. 69 and associated description on Page 7: “In 6904 the data request is sent to each of the tunnels along with the segment that this Tunnel needs to fetch from the Data Server. The Tunnel will then make that request to the Data Server and return the information received to the Client.”</p> <p>See also, e.g., Page 13: “Next, the agent decides on a chunk size {8006} which will be the size of the pieces in to which the data is broken down to, for the sake of assigning pieces to the Peers to deliver to the Client (chunk_size). The chunk_size</p>	<p>hence providing a degree of redundancy. For example, in case of carrying the three slices over the Internet and a failure to receive one of the slices, the remaining two slices may be used to fully reconstruct the whole content.”</p>
--	--	---	--

		<p>can either be a set value (for example, 16KB), or a number that changes according to the size of the data that is requested (for example 1/10th of the size of the data itself).”</p> <p>E.g., Page 33: “In 17402 the client is going to load a file which is available via multiple sources (source(1) ... source(n_sources). In 17404, the requested file is split up in toN chunks of equal size (chunk_size) that was pre-determined in the system. For example, this chunk size can be 16KB.”</p> <p>See also Pages 33-34 (“Microchunks” section)</p> <p>See also Fig. 73</p>	
29f	wherein each of the [tunnel] devices in the group is a client device [comprising a consumer electronic device with a client operating system and	<p>E.g., Page 1: “In the network of figure 50, there is a new type of element called a Tunnel. As in patent 12836059, each of the communication devices in the network may store the same software that under different conditions enables that communication device to assume a different role, and may also take multiple roles at the same time (e.g. the communication device can be both a Client and a Tunnel for example), and thus a device may for example communicate with itself under different roles (for</p>	<p>E.g., Page 56: “A device herein (such as device 11) may consist of, be part of, or include, a Personal Computer (PC), a desktop computer, a mobile computer, a laptop computer, a notebook computer, a tablet computer, a server computer, a handheld computer, a handheld device, a Personal Digital Assistant (PDA) device, or a cellular handset.”</p> <p>E.g., Page 100: “Each of the network elements herein, such as the first, second, and third</p>

		<p>example a Client may use itself also as a Tunnel).”</p> <p>See, e.g., Fig. 234 and associated description on Page 46: “Figure 234 is a diagram that shows such an architecture. A network element 23402 has an operating system 10 and applications 8, with a virtual application gateway module 23404 running on it as well.”</p> <p>See also, e.g., Pages 47-48: “In a computing device that is connected to a network and includes an operating system and applications, a method by which a program on the device provides the communication configuration to the operating system instead of the operating system getting it from a gateway on the network, where the program communicates with the external gateway to get the configuration information so that it can communicate to other network elements, and provides separate configuration information to the operating system, thereby having the operating system communicate with the program, and the program communicating with the external network elements[.]”</p>	<p>devices, may store, operate, or use, a client operating system, that may consist or, comprise of, or may be based on, Microsoft Windows 7, Microsoft Windows XP, Microsoft Windows 8, Microsoft Windows 8.1, Linux, or Google Chrome OS. The client operating system may be a mobile operating system, such as Android version 2.2 (Froyo), Android version 2.3 (Gingerbread), Android version 4.0 (Ice Cream Sandwich), Android Version 4.2 (Jelly Bean), Android version 4.4 (KitKat)), Apple iOS version 3, Apple iOS version 4, Apple iOS version 5, Apple iOS version 6, Apple iOS version 7, Microsoft Windows® Phone version 7, Microsoft Windows® Phone version 8, Microsoft Windows® Phone version 9, or Blackberry® operating system.”</p> <p>E.g., Abstract: “A client device may also serve as a tunnel device, serving as an intermediate device to other client devices. Similarly, a tunnel device may also serve as a client device for fetching content from a data server.”</p> <p>E.g., Page 167: “A client device such as a client device #1 341 may assume the role of the tunnel-using client device #1 31 a, the role of the agent/peers-using client device #1 201 a, or both.”</p>
--	--	---	---

29g	wherein the second identifier is a complete URL and	<p>See, e.g., Fig. 124 and associated description on Page 20: “Figure 124 is a flowchart for an example of such a specific back end implementation, for analytic servers. In 12402, a request to a specific analytic server is identified. This may be identified by the IP of the server, by the URL of the file that the browser is requesting from the analytics server, or by any other method.”</p> <p>See also, e.g., Fig. 130, block 13008: “URL with offset is requested or received (e.g., http://www.youtube.com/watch?v=9MSB3P7CZIE?ST=1:48)</p> <p>E.g., Page 21: “a system that comprises of a software module, a database, where upon getting a uri as an input, the system looks up the domain of the URL within the database, retrieves the schema of the URL and can identify the identity of the video, and the offset to view within the video according to the schema[.]”</p> <p>E.g., Pages 18-19: “Here are some examples of who may benefit from different treatment as described: 1. by URL: specific files on the Internet (e.g. “Wikipedia.org/contact.html”) 2. by Domain: specific web sites (e.g. “Wikipedia.org”) 3. by Data server IP: specific servers (e.g. “208.80.152.201”) 4.</p>	<p>E.g., Pages 72-73: “Each of the identifiers herein may be an IP address (in IPv4 or IPv6 form) or a URL. Each of the servers may be a web server using HyperText Transfer Protocol (HTTP) that responds to HTTP requests via the Internet, and the first and second requests may be HTTP requests. Each communication with a server may be based on, or using, HTTP persistent connection.”</p> <p>E.g., Page 145: “A content, such as an URL (or a web-page, or a web-site) which is typically stored in a data server, such as the data server #1 22 a, may be requested by the client device, such as the client device 201 a, as shown in a state ‘Content Needed’ 223 b in the chart 220. The client device sends a ‘Request List’ message 226 e to the acceleration server 202, corresponding to a ‘Request Agents List’ step 231 c in the flowchart 230. This request includes the URL or any other identifier of the requested content.”</p> <p>E.g., Page 115: “In one example, the content may include, be a part of, or a whole of, a URL or a website page.”</p> <p>See, e.g., Fig. 21</p> <p>E.g., Page 169: “For example, in a video identifier such as</p>
------------	---	---	--

		<p>by Type of file: specific file types (e.g. ".flv files") 5. by time of day: specific handling of all files or a group of files during certain hours of the day (e.g. "all files between 11pm to 4am") 6. by Geography of Client: specific handling according to where the Client is (e.g. "for all Clients in Germany")"</p> <p>E.g., Page 68: "In 60002, the module receives from its caller a URL that the communication device needs to fetch..."</p> <p>See also Page 22</p>	<p>https://www.youtube.com/watch?v=9mSb3P7cZIE?ST=1:48, the field 'https://www.youtube.com' is the URL domain, which identify the server from which the video can be fetched, , the part '9mSb3P7cZIE' identifies the video data (such as a movie) as a whole, and the offset '1:48' part in the video starting point, in this example after 1 minute and 48 seconds after the video start point."</p> <p>See, e.g., Fig. 15a and associated description on Page 131: "...the URL fetched is shown in the fourth column 153 d..."</p>
29h	wherein the first device remains anonymous as to the first tunnel device and the first server].	<p>See, e.g., Fig. 58 and associated description on Page 2 : "In such a case where it is beneficial for the Client 2 to fetch the data directly from the Data Server 5, the Client 2 may use one or more Tunnels 16 to carry out the request to the Data Server 5 <i>on its behalf</i>." (emphasis added)</p> <p>E.g., Page 3: "[I]t is always possible for the Client 2 to fetch the information through Tunnels 16 by requesting that they fetch the information <i>on the Client's behalf</i> from the Data Server 5." (emphasis added)</p> <p>See, e.g., Figs. 61 and 62 and associated description on Page 3: "Figures 61 and 62</p>	<p>E.g., Page 117: "Since the data server #1 22 a is accessed by, and sends information only to, tunnel devices (such as the tunnel device #1 33 a), and is not aware of the final content destination being the client device #1 31 a, the identity (such as the IP address) of the client device #1 31 a is concealed from the data server #1 22 a, thus providing anonymity and untraceability. Further, in a case where the data server #1 22 a is a web server, the method and system described may provide for an anonymous web browsing."</p> <p>E.g., Page 128: "The service allows the client device (such as client device #1 31 a) to quickly and anonymously fetch content from the data server #1 22 a."</p>

		<p>are flowcharts that suggest an implementation of a method of choosing the optimal method- whether to fetch the data from the acceleration network (called in the figures "PATH_ACCEL"), or <i>directly via the Tunnels</i> (called in the figures "PATH_DIRECT")." (emphasis added)</p> <p>See, e.g., Fig. 69 and associated description on Page 7: "In 6904 the data request is sent to each of the tunnels along with the segment that this Tunnel needs to fetch from the Data Server. The Tunnel will then make that request to the Data Server and return the information received to the Client."</p> <p>See, e.g., Fig. 234 associated description on Page46: "When the network element 23402 is connected to a gateway 23410, the virtual application gateway 23404 intercepts the operating system's request to the network for receiving configuration information, and receives on its own the configuration information from the gateway 23410 and provides that information back to the operating system, thereby acting as a sort of proxy for the initial configuration information. Some of the requests that the virtual application gateway module 23404 receives may require it to do actions on its own which</p>	<p>See, e.g., Fig. 13 and associated description on Page 128 : "In one example shown as a system 130 in FIG. 13, the client device #1 31 a functionality is implemented in as client device #1 system 133, comprising a computer 132 (may be used for GUI or as a client), is communicating with a proxy server 131."</p>
--	--	---	---

		<p>are not in the form of being a proxy. One such action is the IP request from the operating system in that case the virtual application gateway module 23404 will need to provide a local IP to the operating system.”</p> <p>See, e.g., Fig. 236 and associated description on Page 47: “Then, the VAGM sets 23604 itself up for intercepting the IP requests of the operating system that would normally have gone out to the gateway... Once such a request has been trapped 23610 the VAGM provides an IP to the OS in a similar way to how NAT devices provide IPs to requesting clients. The VAGM then acts 23606 as a proxy/nat for the operating system as NAT devices do.”</p>	
30	<p>30. (proposed substitute for claim 16)16. The method according to claim 15[29], wherein the content is composed of bits, nibbles, bytes, characters, words, or strings, and the partitioning is based on bit, nibble, byte, multi-byte, number, character, word, or string level.</p>	<p>E.g., Page 68: “If the cached URL has expired, then in 60012 a range request for X bytes is sent to the external source (could be 16kb for example, or could be a random range of 5% of the size (in bytes) of the cached URL's size.”</p> <p>E.g., Page 2: “The Client 2 will also determine how much of the data it will request from each of the Tunnels 16 based on their speed, and thus assign a 'range' of the data (for example for a data of 1200</p>	<p>See, e.g., original claim 135</p> <p>E.g., Page 82: “A content herein may be composed of bits, nibbles, bytes, characters, words, or strings, and the partitioning may be based on bit, nibble, byte, multi-byte, number, character, word, or string level.”</p> <p>E.g., Page 138: “The slicing may be in a bit, nibble (4-bits), byte (8-bits), word (multiple bytes), character, string, or file level. For example, in a case wherein the content</p>

		<p>bytes where two agents A and Bare selected, a range assignment may be that A is assigned the bytes between byte 0 and byte 856, and B is assigned bytes 857 to 1200). The Client 2 will then send the request to all these Tunnels 16 along with the requested range for them to fetch from the Data Server.”</p> <p>See, e.g., Fig. 70 and associated description on Pages 7-8: “Figure 70 shows an example of how a tunnel may be implemented. The Tunnel receives a request (7002) from another network element (e.g. a Client) which includes the standard information of a data request (DR) including the address of the data server (such as the name or IP of a web server), the information requested (such as a URL), the range of the data that is requested (e.g. all of the data, or only a certain byte range, for example bytes 1000 to 1020), and additional information about the request (such as the cookies in HTTP). The Tunnel the[n] creates a request (7004) to the data server based on the DR and sends it to the data server. Once the response is received (7006}, the Tunnel sends the data (7010) back to the requesting Client.”</p> <p>E.g., Pages 18-19: “Here are some examples of who may benefit from</p>	<p>includes 240 bytes designated byte #1 to byte #240, using a byte level partitioning into two slices results in a first slice (slice #1) including byte #1 to byte #120, and a second slice (slice #2) including byte #121 to byte #240. In the case of byte-level partitioning into three slices (referred as slice #1, slice #2, and slice #3), a first slice (slice #1) may be including byte #1 to byte #80, a second slice (slice #2) may be including byte #81 to byte #160, and a third slice (slice #3) may be including byte #161 to byte #240. Similarly, in a case wherein the content include 3 bytes designated byte #1 to byte #3 representing 24 bits, using a bit-level partitioning into four slices results in a slice #1 including the first 6 bits, slice #2 including the next 6 bits, slice #3 including the next 6 bits, and slice #4 including the last 6 bits.”</p> <p>E.g., Page 138: “In one example, the content itself is made of inherent or identifiable parts or segments, and the partition may make use of these parts. In one example, the content may be a website content composed of multiple webpages, and thus the partition may be such that each slice includes one (or few) webpages. Further, the partitioning may be sequential or non-sequential in the content.”</p> <p>See also, e.g., Fig. 21</p>
--	--	--	--

		<p>different treatment as described: 1. by URL: specific files on the Internet (e.g. "Wikipedia.org/contact.html") 2. by Domain: specific web sites (e.g. "Wikipedia.org") 3. by Data server IP: specific servers (e.g. "208.80.152.201") 4. by Type of file: specific file types (e.g. ".flv files") 5. by time of day: specific handling of all files or a group of files during certain hours of the day (e.g. "all files between 11pm to 4am") 6. by Geography of Client: specific handling according to where the Client is (e.g. "for all Clients in Germany")"</p> <p>See, e.g., Fig. 73</p> <p>See also, e.g., Page 13: "Next, the agent decides on a chunk size {8006} which will be the size of the pieces in to which the data is broken down to, for the sake of assigning pieces to the Peers to deliver to the Client (chunk_size). The chunk_size can either be a set value (for example, 16KB), or a number that changes according to the size of the data that is requested (for example 1/10th of the size of the data itself)."</p>	
31	31. (proposed substitute for claim 17) 47. The method according to claim 45[29],	E.g., Pages 18-19: "Here are some examples of who may benefit from different treatment as described: 1. by	See, e.g., original claim 136

	<p>wherein the content is composed of files, or programs, and the partitioning is based on file or program level.</p>	<p>URL: specific files on the Internet (e.g. "Wikipedia.org/contact.html") 2. by Domain: specific web sites (e.g. "Wikipedia.org") 3. by Data server IP: specific servers (e.g. "208.80.152.201") 4. by Type of file: specific file types (e.g. ".flv files") 5. by time of day: specific handling of all files or a group of files during certain hours of the day (e.g. "all files between 11pm to 4am") 6. by Geography of Client: specific handling according to where the Client is (e.g. "for all Clients in Germany")"</p> <p>See, e.g., Fig. 73</p> <p>See, e.g., Fig. 69 and associated description on Page 7: "In 6904 the data request is sent to each of the tunnels along with the <i>segment</i> that this Tunnel needs to fetch from the Data Server." (emphasis added)</p> <p>E.g., Page 23: "Data servers typically limit the number of active connections that a host can keep open with them concurrently. This limitation is detrimental in various scenarios. One such scenario as an example is where a communications device with a web browser needs to load <i>many URLs from a Data Server in order to display a web page...</i>" (emphasis added)</p>	<p>E.g., Page 82: "Alternatively or in addition, a content herein may be composed of files or programs, and the partitioning may be based on file or program level."</p> <p>E.g., Page 138: "The slicing may be in a bit, nibble (4-bits), byte (8-bits), word (multiple bytes), character, string, or file level..."</p> <p>E.g., Page 138: "In one example, the content itself is made of inherent or identifiable parts or segments, and the partition may make use of these parts. In one example, the content may be a website content composed of multiple webpages, and thus the partition may be such that each slice includes one (or few) webpages. Further, the partitioning may be sequential or non-sequential in the content."</p> <p>See also, e.g., Fig. 21</p>
--	---	--	---

32	32. (proposed substitute for claim 18) 18 . The method according to claim 17 [31], wherein the content is a website content comprising multiple webpages, and the partitioning is based webpages level.	<p>E.g., Page 80: “68002 are a group of pages in the physical memory of a computer (page #1, Page #2, ... Page #N), in this example each of the pages is a range of 4KB.”</p> <p>E.g., Pages 18-19: “Here are some examples of who may benefit from different treatment as described: 1. by URL: specific files on the Internet (e.g. "Wikipedia.org/contact.html") 2. by Domain: specific web sites (e.g. "Wikipedia.org") 3. by Data server IP: specific servers (e.g. "208.80.152.201") 4. by Type of file: specific file types (e.g. ".flv files") 5. by time of day: specific handling of all files or a group of files during certain hours of the day (e.g. "all files between 11pm to 4am") 6. by Geography of Client: specific handling according to where the Client is (e.g. "for all Clients in Germany")”</p> <p>See, e.g., Fig. 73</p> <p>See, e.g., Fig. 69 and associated description on Page 7: “In 6904 the data request is sent to each of the tunnels along with the <i>segment</i> that this Tunnel needs to fetch from the Data Server.” (emphasis added)</p>	<p>See, e.g., original claim 137</p> <p>E.g., Page 82: “Further, the content may be a website content comprising multiple webpages, and the partitioning may be based on webpages level.”</p> <p>E.g., Page 138: “In one example, the content itself is made of inherent or identifiable parts or segments, and the partition may make use of these parts. In one example, the content may be a website content composed of multiple webpages, and thus the partition may be such that each slice includes one (or few) webpages. Further, the partitioning may be sequential or non-sequential in the content.”</p> <p>See also, e.g., Fig. 21</p>

		E.g., Page 23: “Data servers typically limit the number of active connections that a host can keep open with them concurrently. This limitation is detrimental in various scenarios. One such scenario as an example is where a communications device with a web browser needs to load many URLs from a Data Server in order to display a web page... ” (emphasis added)	
33	33. (proposed substitute for claim 19) 19 . The method according to claim 15 [29], wherein all of the parts of the content are included in all of the content slices.	<p>E.g., Page 7: “...the range of the data that is requested (e.g. all of the data, or only a certain byte range, for example bytes 1000 to 1020)...” (emphasis added)</p> <p>See, e.g., Fig. 70 and associated description on Pages 7-8: “Figure 70 shows an example of how a tunnel may be implemented. The Tunnel receives a request (7002) from another network element (e.g. a Client) which includes the standard information of a data request (DR) including the address of the data server (such as the name or IP of a web server), the information requested (such as a URL), the range of the data that is requested (e.g. all of the data, or only a certain byte range, for example bytes 1000 to 1020), and additional information about the request (such as the cookies in HTIP).</p>	<p>See, e.g., original claim 138</p> <p>E.g., Page 82: “All parts of the content may be included in all of the content slices, and two or more, or all of the content slices, may be having the same size.”</p> <p>E.g., Page 82: “Two or more of the content slices may include the same information. Further, the same part of the content may be included in two or more content slices.”</p> <p>E.g., Page 84: “Part of, or all of, the content parts may be having the same size, that may be 8 KB, 16 KB, 32 KB, or 64 KB. Two or more content parts may be identical and may contain the same data.”</p> <p>E.g., Page 126: “The same content may be requested and fetched using multiple tunnel</p>

		<p>The Tunnel the[n] creates a request (7004) to the data server based on the DR and sends it to the data server. Once the response is received (7006}, the Tunnel sends the data (7010) back to the requesting Client.” (emphasis added)</p> <p>E.g., Page 50: “In figure 247, a method of transmitting packets in parallel over multiple routes is described, in which one communications device (NE1) wishes to communicate with another communications device (NE2), sending the <i>same data</i> over multiple routes in parallel.” (emphasis added)</p> <p>See, e.g., Fig. 136 and associated description on Page 23: “Figure 136 describes such a system. In it, the Client 2 would like to open more connections with the Data Server 5 than the Data Server 5 will allow (<i>where each of these connections is loading one or more requests, where each request is a resource, a range of a resource, or multiple resources</i>). The Client then communicates with participating Tunnels 4 (i.e. network devices on which software is installed for this purpose), and requesting from each of these Tunnels to load several of these requests, such that the each of the participating elements (the</p>	<p>devices as exemplified above. Alternatively or in addition, the content may be partitioned into multiple slices (overlapping or non-overlapping), where each slice is requested and fetched using a distinct tunnel device (or via the client device serving as its own tunnel).”</p> <p>E.g., Page 139: “The partitioning may be non-overlapping, wherein each slice includes a distinct part of the content, as is exemplified above in a case wherein the content includes 240 bytes designated byte #1 to byte #240, where using a byte level partitioning into three slices (referred as slice #1, slice #2, and slice #3), results in a first slice (slice #1) including byte #1 to byte #80, a second slice (slice #2) including byte #81 to byte #160, and a third slice (slice #3) including byte #161 to byte #240. Alternatively or in addition, an overlapping partitioning may be applied, where the same part of the content is included in multiple slices.”</p>
--	--	--	--

		Client and the Tunnels from which it requested the loads) does not open more connections vs. the Data Server 5 that the Data Server 5 allows.” (emphasis added)	
34	34. (proposed substitute for claim 20) 20 . The method according to claim 15 [29], wherein all of the content slices have the same size.	<p>E.g., Page 13: “For example, this can be done according to the BW/RTT from the Peer to the Client. Next, the agent decides on a chunk size {8006} which will be the size of the pieces in to which the data is broken down to, for the sake of assigning pieces to the Peers to deliver to the Client (chunk_size). The chunk_size can either be a set value (for example, 16KB), or a number that changes according to the size of the data that is requested (for example 1/10th of the size of the data itself).”</p> <p>E.g., Page 33: “In 17402 the client is going to load a file which is available via multiple sources (source(1) ... source(n_sources). In 17404, the requested file is split up in toN chunks of <i>equal size</i> (chunk_size) that was pre-determined in the system. For example, this chunk size can be 16KB.” (emphasis added)</p> <p>See, e.g., Fig. 69 and associated description on Page 7: “In 6902, the requested data is split in to ranges, such that there is one range per participating Tunnel. It is possible to simply choose</p>	<p>See, e.g., original claim 139</p> <p>E.g., Page 82: “All parts of the content may be included in all of the content slices, and two or more, or all of the content slices, may be having the same size.”</p> <p>E.g., Page 82: “Two or more of the content slices may include the same information. Further, the same part of the content may be included in two or more content slices.”</p> <p>E.g., Page 84: “Part of, or all of, the content parts may be having the same size, that may be 8 KB, 16 KB, 32 KB, or 64 KB. Two or more content parts may be identical and may contain the same data.”</p> <p>E.g., Page 126: “The same content may be requested and fetched using multiple tunnel devices as exemplified above. Alternatively or in addition, the content may be partitioned into multiple slices (overlapping or non-overlapping), where each slice is requested and fetched using a distinct tunnel device (or via the client device serving as its own tunnel).”</p>

		ranges of <i>equal size</i> to implement this.” (emphasis added)	E.g., Page 139: “The partitioning may be non-overlapping, wherein each slice includes a distinct part of the content, as is exemplified above in a case wherein the content includes 240 bytes designated byte #1 to byte #240, where using a byte level partitioning into three slices (referred as slice #1, slice #2, and slice #3), results in a first slice (slice #1) including byte #1 to byte #80, a second slice (slice #2) including byte #81 to byte #160, and a third slice (slice #3) including byte #161 to byte #240. Alternatively or in addition, an overlapping partitioning may be applied, where the same part of the content is included in multiple slices.”
35	35. (proposed substitute for claim 23) 23 . The method according to claim 15 [29], wherein the partitioning is sequential in the content.	E.g., Page 2: “The Client 2 will also determine how much of the data it will request from each of the Tunnels 16 based on their speed, and thus assign a 'range' of the data (for example for a data of 1200 bytes where two agents A and Bare selected, a range assignment may be that A is assigned the bytes between byte 0 and byte 856, and B is assigned bytes 857 to 1200). The Client 2 will then send the request to all these Tunnels 16 along with the requested range for them to fetch from the Data Server.”	See, e.g., original claim 142 E.g., Page 76: “The partitioning may be sequential or non-sequential in the content...” E.g., Page 224: “The steps described herein may be sequential, and performed in the described order...” E.g., Page 120: “The tasks relating to the different data paths, such as shown in a flowchart 100 a, relating to communicating with the multiple tunnel devices and/or

		<p>E.g., Page 7: “In 6902, the requested data is split in to ranges, such that there is one range per participating Tunnel. It is possible to simply choose ranges of equal size to implement this. A possibly more efficient implementation is to provide each Tunnel a range that is proportional to that Tunnel's ability to provide the data (for example, based on that Tunnel's upstream and downstream traffic to the Client, as measured in previous communications between them).”</p> <p>E.g., Page 13: “The Agent then assigns the chunks of data to each Peer in the following manner: starting with the first Peer in the list {8010}, which is the fastest Peer because of the sorting in 8004, it then assigns a number of chunks to the Peer {8012} by using a metric of allocation. For example, by providing a set number of chunks per Peer, or by assigning for example BW/RTT chunks to the Peer. If there are more chunks of the data left to assign {8014}, then the next Peer in the list is selected {8016}, and more chunks are assigned to the Peer in 8012, and so on until all chunks are assigned, and the allocation function ends {8018}.”</p> <p>E.g., Page 33: “In 17402 the client is going to load a file which is available via multiple sources (source(1) ...</p>	<p>direct access, may be executed sequentially or in parallel.”</p> <p>E.g., Page 82: “The partitioning may be sequential or non-sequential in the content, and the number of the content slices may be equal to, higher than, or lower than, the number of devices in the group.”</p> <p>E.g., Page 101: “The steps described herein may be sequential, and performed in the described order. For example, in a case where a step is performed in response to another step, or upon completion of another step, the steps are executed one after the other.”</p> <p>E.g., Page 139: “The partitioning may be non-overlapping, wherein each slice includes a distinct part of the content, as is exemplified above in a case wherein the content includes 240 bytes designated byte #1 to byte #240, where using a byte level partitioning into three slices (referred as slice #1, slice #2, and slice #3), results in a first slice (slice #1) including byte #1 to byte #80, a second slice (slice #2) including byte #81 to byte #160, and a third slice (slice #3) including byte #161 to byte #240. Alternatively or in addition, an overlapping partitioning may be applied, where the same part of the content is included in multiple slices.”</p>
--	--	--	---

		<p>source(n_sources). In 17404, the requested file is split up in toN chunks of equal size (chunk_size) that was pre-determined in the system. For example, this chunk size can be 16KB.”</p> <p>E.g., Page 7: “...the range of the data that is requested (e.g. all of the data, or only a certain byte range, for example bytes 1000 to 1020)...”</p>	
36	<p>36. (proposed substitute for claim 24) 24. The method according to claim 15[29], wherein the partitioning is non-sequential in the content.</p>	<p>E.g., Page 13: “The Agent then assigns the chunks of data to each Peer in the following manner: starting with the first Peer in the list {8010}, which is the fastest Peer because of the sorting in 8004, it then assigns a number of chunks to the Peer {8012} by using a metric of allocation. For example, by providing a set number of chunks per Peer, or by assigning for example BW/RTT chunks to the Peer. If there are more chunks of the data left to assign {8014}, then the next Peer in the list is selected {8016}, and more chunks are assigned to the Peer in 8012, and so on until all chunks are assigned, and the allocation function ends {8018}.”</p> <p>E.g., Page 7: “In 6902, the requested data is split in to ranges, such that there is one range per participating Tunnel. It is</p>	<p>See, e.g., original claim 143</p> <p>E.g., Page 76: “The partitioning may be sequential or non-sequential in the content...”</p> <p>E.g., Page 101: “However, in case where two or more steps are not explicitly described as being sequentially executed, these steps may be executed in any order, or may be simultaneously performed.”</p> <p>E.g., Page 120: “The tasks relating to the different data paths, such as shown in a flowchart 100 a, relating to communicating with the multiple tunnel devices and/or direct access, may be executed sequentially or in parallel.”</p>

		possible to simply choose ranges of equal size to implement this. A possibly more efficient implementation is to provide each Tunnel a range that is proportional to that Tunnel's ability to provide the data (for example, based on that Tunnel's upstream and downstream traffic to the Client, as measured in previous communications between them)."	<p>E.g., Page 82: "The partitioning may be sequential or non-sequential in the content, and the number of the content slices may be equal to, higher than, or lower than, the number of devices in the group."</p> <p>E.g., Page 139: "The partitioning may be non-overlapping, wherein each slice includes a distinct part of the content, as is exemplified above in a case wherein the content includes 240 bytes designated byte #1 to byte #240, where using a byte level partitioning into three slices (referred as slice #1, slice #2, and slice #3), results in a first slice (slice #1) including byte #1 to byte #80, a second slice (slice #2) including byte #81 to byte #160, and a third slice (slice #3) including byte #161 to byte #240. Alternatively or in addition, an overlapping partitioning may be applied, where the same part of the content is included in multiple slices."</p>
37	37. (proposed substitute for claim 27) 27. The method according to claim 15[29], wherein the number of content slices is higher than the number of [tunnel] devices in the group.	<p>E.g., Page 33: "In this example, the client 2 loads these chunks from 3 available sources, fetching more chunks from faster sources, so that 3 chunks are requested from source 117002, 1 chunk from source 2 17004 and 2 chunks from source 3 17006."</p> <p>E.g., Page 13: "The Agent then assigns the chunks of data to each Peer in the</p>	<p>See, e.g., original claim 146</p> <p>E.g., Page 82: "The partitioning may be sequential or non-sequential in the content, and the number of the content slices may be equal to, higher than, or lower than, the number of devices in the group."</p> <p>E.g., Page 82: "Two or more of the content slices may include the same information."</p>

		<p>following manner: starting with the first Peer in the list {8010}, which is the fastest Peer because of the sorting in 8004, it then assigns a number of chunks to the Peer {8012} by using a metric of allocation. For example, by providing a set number of chunks per Peer, or by assigning for example BW/RTT chunks to the Peer. If there are more chunks of the data left to assign {8014}, then the next Peer in the list is selected {8016}, and more chunks are assigned to the Peer in 8012, and so on until all chunks are assigned, and the allocation function ends {8018}.”</p> <p>See also, e.g., Fig. 136 and associated description on Page 23: “Figure 136 describes such a system. In it, the Client 2 would like to open more connections with the Data Server 5 than the Data Server 5 will allow (<i>where each of these connections is loading one or more requests, where each request is a resource, a range of a resource, or multiple resources</i>). The Client then communicates with participating Tunnels 4 (i.e. network devices on which software is installed for this purpose), and requesting from each of these Tunnels to load several of these requests, such that the each of the participating elements (the Client and the Tunnels from which it requested the loads) does not open more</p>	<p>Further, the same part of the content may be included in two or more content slices.”</p> <p>E.g., Page 84: “Part of, or all of, the content parts may be having the same size, that may be 8 KB, 16 KB, 32 KB, or 64 KB. Two or more content parts may be identical and may contain the same data.”</p> <p>E.g., Page 126: “The same content may be requested and fetched using multiple tunnel devices as exemplified above. Alternatively or in addition, the content may be partitioned into multiple slices (overlapping or non-overlapping), where each slice is requested and fetched using a distinct tunnel device (or via the client device serving as its own tunnel).”</p> <p>See also, e.g., Fig. 21</p> <p>See also, e.g., Page 126: “It is noted that in such a partition, the content may be fully reconstructed from any two of the slices, hence providing a degree of redundancy. For example, in case of carrying the three slices over the Internet and a failure to receive one of the slices, the remaining two slices may be used to fully reconstruct the whole content.”</p>
--	--	---	---

		<p>connections vs. the Data Server 5 that the Data Server 5 allows.” (emphasis added)</p> <p>See, e.g., Fig. 70 and associated description on Pages 7-8: “Figure 70 shows an example of how a tunnel may be implemented. The Tunnel receives a request (7002) from another network element (e.g. a Client) which includes the standard information of a data request (DR) including the address of the data server (such as the name or IP of a web server), the information requested (such as a URL), the range of the data that is requested (e.g. all of the data, or only a certain byte range, for example bytes 1000 to 1020), and additional information about the request (such as the cookies in HTTP). The Tunnel then creates a request (7004) to the data server based on the DR and sends it to the data server. Once the response is received (7006), the Tunnel sends the data (7010) back to the requesting Client.”</p> <p>See also Pages 33-34 (“Microchunks” section)</p>	
38	38. (proposed substitute for claim 28) 28 . The method according to claim 15[29], wherein the number of content	E.g., Page 7: “...the range of the data that is requested (e.g. all of the data, or only a certain byte range, for example bytes 1000 to 1020)...”	<p>See, e.g., original claim 147</p> <p>E.g., Page 82: “The partitioning may be sequential or non-sequential in the content,</p>

	<p>slices is lower than the number of [tunnel] devices in the group.</p>	<p>E.g., Page 50: “In figure 247, a method of transmitting packets in parallel over multiple routes is described, in which one communications device (NE1) wishes to communicate with another communications device (NE2), sending the same data over multiple routes in parallel.”</p> <p>E.g., Page 33: “In 17402 the client is going to load a file which is available via multiple sources (source(1) ... source(n_sources)). In 17404, the requested file is split up in toN chunks of equal size (chunk_size) that was pre-determined in the system. For example, this chunk size can be 16KB.”</p> <p>See also, e.g., Fig. 136 and associated description on Page 23: “Figure 136 describes such a system. In it, the Client 2 would like to open more connections with the Data Server 5 than the Data Server 5 will allow (<i>where each of these connections is loading one or more requests, where each request is a resource, a range of a resource, or multiple resources</i>). The Client then communicates with participating Tunnels 4 (i.e. network devices on which software is installed for this purpose), and requesting from each of these Tunnels to</p>	<p>and the number of the content slices may be equal to, higher than, or lower than, the number of devices in the group.”</p> <p>E.g., Page 82: “Two or more of the content slices may include the same information. Further, the same part of the content may be included in two or more content slices.”</p> <p>E.g., Page 84: “Part of, or all of, the content parts may be having the same size, that may be 8 KB, 16 KB, 32 KB, or 64 KB. Two or more content parts may be identical and may contain the same data.”</p> <p>E.g., Page 126: “The same content may be requested and fetched using multiple tunnel devices as exemplified above. Alternatively or in addition, the content may be partitioned into multiple slices (overlapping or non-overlapping), where each slice is requested and fetched using a distinct tunnel device (or via the client device serving as its own tunnel).”</p> <p>See also, e.g., Fig. 21</p> <p>See also, e.g., Page 126: “It is noted that in such a partition, the content may be fully reconstructed from any two of the slices, hence providing a degree of redundancy. For example, in case of carrying the three slices over the Internet and a failure to receive one</p>
--	--	---	--

		<p>load several of these requests, such that the each of the participating elements (the Client and the Tunnels from which it requested the loads) does not open more connections vs. the Data Server 5 that the Data Server 5 allows.” (emphasis added)</p> <p>See also Pages 33-34 (“Microchunks” section)</p>	<p>of the slices, the remaining two slices may be used to fully reconstruct the whole content.”</p>
--	--	--	---