Office de la Propriété Intellectuelle du Canada
Un organisme d'Industrie Canada

Canadian Intellectual Property Office
An agency of Industry Canada

CA 2542407 A1 2005/05/06

(21) 2 542 407

(12) DEMANDE DE BREVET CANADIEN
CANADIAN PATENT APPLICATION

(13) A1

(86) Date de dépôt PCT/PCT Filing Date: 2004/10/04
(87) Date publication PCT/PCT Publication Date: 2005/05/06
(85) Entrée phase nationale/National Entry: 2006/04/11
(86) N° demande PCT/PCT Application No.: EP 2004/052433
(87) N° publication PCT/PCT Publication No.: 2005/041523
(30) Priorité/Priority: 2003/10/14 (US10/605,621)

(51) Cl.Int./Int.Cl. *H04L 29/06* (2006.01)

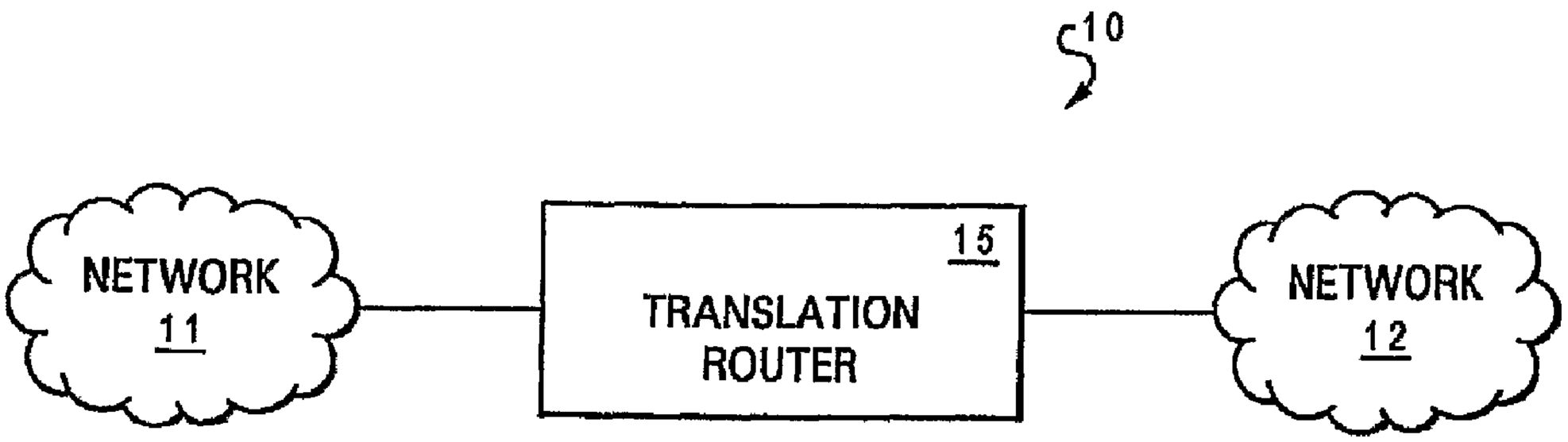
(71) Demandeur/Applicant:
INTERNATIONAL BUSINESS MACHINES CORPORATION, US

(72) Inventeurs/Inventors:
ANDERSON, RICHARD EUGENE, US;
FAUCHER, MARC RAYMOND, US;
GEORGIOU, CHRISTOS JOHN, US;
NOEL, FRANCIS JR., US

(74) Agent: BARRETT, B.P.

(54) Titre : PROCEDE ET APPAREIL DE TRADUCTION DE PAQUETS DE DONNEES D'UN PROTOCOLE DE RESEAU EN UN AUTRE

(54) Title: METHOD AND APPARATUS FOR TRANSLATING DATA PACKETS FROM ONE NETWORK PROTOCOL TO ANOTHER



(57) Abrégé/Abstract:
A method for translating data packets from one network protocol to another is disclosed. A set of translation templates is constructed. The translation templates are then loaded into a translation template cache. In response to a data packet from a first network arriving at a translation router, an appropriate translation template is selected from the set of translation templates within the translation template cache according to the translation context of the data packet. Next, a new header for transmission into a second network is constructed by reading header fields of the data packet from the first network along with the appropriate translation template in the translation template cache. The data payload of the data packet from the first network is subsequently removed from the header of the data packet and then appended to the constructed header of the second network. Finally, the newly constructed data packet is transmitted to the second network.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
6 May 2005 (06.05.2005)

PCT

(10) International Publication Number
WO 2005/041523 A1

(51) International Patent Classification[7]: H04L 29/06

(21) International Application Number:
PCT/EP2004/052433

(22) International Filing Date: 4 October 2004 (04.10.2004)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/605,621 14 October 2003 (14.10.2003) US

(71) Applicant *(for all designated States except US)*: **INTERNATIONAL BUSINESS MACHINES CORPORATION** [US/US]; New Orchard Road, Armonk, NY 10504 (US).

(71) Applicant *(for MG only)*: **IBM UNITED KINGDOM LIMITED** [GB/GB]; PO Box 41, North Harbour, Portsmouth Hampshire PO6 3AU (GB).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **ANDERSON, Richard, Eugene** [US/US]; 11 Country View Drive, Jericho, VT 05465 (US). **FAUCHER, Marc, Raymond** [US/US]; 44 Winding Brook Drive, South Burlington, VT 05403 (US). **GEORGIOU, Christos, John** [US/US]; 19 Arlington Road, Scarsdale, NY 10583 (US). **NOEL, Francis, Jr.** [US/US]; 4021 Swarthmore Road, Durham, NC 27707 (US).

(74) Agent: **BURT, Roger, James**; IBM United Kingdom Limited, Intellectual Property Law, Hursley Park, Winchester Hampshire SO21 2JN (GB).

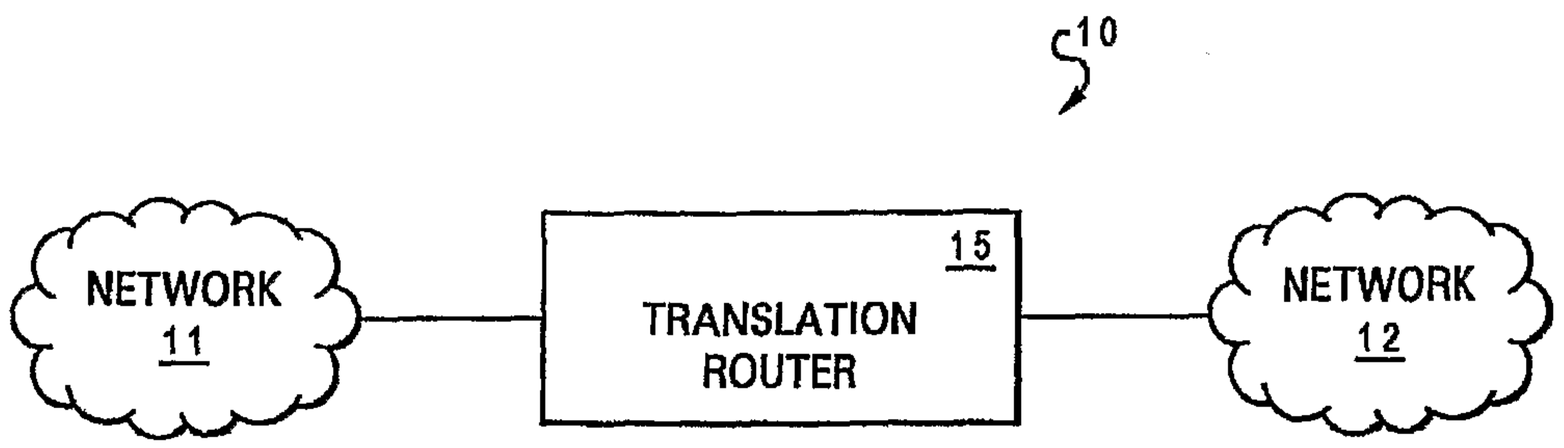
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

[Continued on next page]

(54) Title: METHOD AND APPARATUS FOR TRANSLATING DATA PACKETS FROM ONE NETWORK PROTOCOL TO ANOTHER



(57) Abstract: A method for translating data packets from one network protocol to another is disclosed. A set of translation templates is constructed. The translation templates are then loaded into a translation template cache. In response to a data packet from a first network arriving at a translation router, an appropriate translation template is selected from the set of translation templates within the translation template cache according to the translation context of the data packet. Next, a new header for transmission into a second network is constructed by reading header fields of the data packet from the first network along with the appropriate translation template in the translation template cache. The data payload of the data packet from the first network is subsequently removed from the header of the data packet and then appended to the constructed header of the second network. Finally, the newly constructed data packet is transmitted to the second network.

WO 2005/041523 A1

WO 2005/041523 A1

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Description

METHOD AND APPARATUS FOR TRANSLATING DATA PACKETS FROM ONE NETWORK PROTOCOL TO ANOTHER

Technical Field

[001] The present invention relates to network communications in general, and, in particular, to a method and apparatus for translating data packets. Still more particularly, the present invention relates to a method and apparatus for translating data packets from one network protocol to another.

Background Art

[002] In a mixed protocol network environment, a significant amount of processing power of a network router is spent on translating data packet information from one protocol to another. Packet translation is a relatively time-consuming process. For a network storage environment, it is important to perform packet translation in an efficient manner in order to maintain an overall high system performance.

[003] Conventionally, there are hardware solutions and software solutions for facilitating the packet translation process. One hardware solution is to construct packet headers with a state machine, but the problem with such solution is that there is no flexibility if network protocols change or migrate. Another hardware solution is to construct packet headers with a general purpose processor (GPP), but the problem with such solution is that long sequences of GPP instructions are required to execute protocol translations.

[004] As for software solutions, information from the header or context block of a packet can be used to construct headers or frames. But the problem with such solution is that there are performance constraints from recalculations needed to be made for each translation.

[005] Consequently, it is desirable to provide an improved method and apparatus for translating data packets between different network protocols.

Disclosure of Invention

[006] The present invention provides a method as claimed in claim 1, an apparatus as claimed in claim 6, and a computer program product as claimed in claim 11.

Brief Description of the Drawings

[007] The invention itself, as well as a preferred mode of use, further objects, and advantages thereof, will best be understood by reference to the following detailed description of an illustrative embodiment when read in conjunction with the accompanying drawings, wherein:

[008] Figure 1 is a block diagram of a network environment to which a preferred embodiment of the present invention is applicable;

[009] Figure **2** is a block diagram of an apparatus for translating data packets between different network protocols, in accordance with a preferred embodiment of the present invention;

[010] Figures **3a-3b** are block diagrams of a translation template cache, in accordance with a preferred embodiment of the present invention;

[011] Figure **4** is a block diagram of a translation template for facilitating the translation of a received data packet to a Fibre Channel protocol, in accordance with a preferred embodiment of the present invention; and

[012] Figure **5** is a high-level logic flow diagram of a method for translating data packets from one protocol to another by using the apparatus from Figure **2**, in accordance with a preferred embodiment of the present invention.

Best Mode for Carrying Out the Invention

[013] Referring now to the drawings and in particular to Figure **1**, there is depicted a block diagram of a network environment to which a preferred embodiment of the present invention is applicable. As shown, a network environment **10** includes a network **11** and a network **12**. Each of network **11** and network **12** may have its own separate network protocol. For example, network **11** can be Ethernet while network **12** can be Fibre Channel, or network **11** can be Fibre Channel while network **12** can be InfiniBand. All communications between network **11** and network **12** are handled by a translation router **15**. Translation router **15** is preferably a multi-protocol router. As a multi-protocol router, each port of translation router **15** is associated with a specific network protocol. For example, if translation router **15** has three ports, port 1 of translation router **15** can be associated with Fiber Channel, port 2 of translation router **15** can be associated with Transmission Control Protocol/Internet Protocol (TCP/IP), and port 3 of translation router **15** can be associated with an InfiniBand. Thus, translation router **15** is aware of the type of network protocol to which a network packet belongs based on the incoming port number from which the network packet comes. Similarly, translation router **15** is aware of the type of network protocol to which a network packet should be translated based on the outgoing port number to which the network packet is to be transmitted.

[014] Within a network environment, such as network environment **10**, the processing overhead can be relatively high because of the large number of data packets that needed to be dealt with. However, within the network environment, there are also a couple of features that can be used to alleviate the above-mentioned problem. First, although there are numerous network protocols that are being in-use, the total number of different network protocols are relatively small and the specifics of each network protocol are well defined by various governing standards. Second, most network

protocols use a header structure ahead of the payload in a packet containing packet description information such as network source and destination addresses, type of packet, etc.

[015] With reference now to Figure **2**, there is illustrated a block diagram of an apparatus for translating data packets from one network protocol to another, in accordance with a preferred embodiment of the present invention. As shown, an apparatus **20** includes a translation template cache **21** and a translation engine **22**. Translation engine **22** is coupled to a general purpose processor **25**, and general purpose processor **25** is coupled to an instruction cache **23**, a data cache **24** and a system memory **26**. Apparatus **20** is preferably located within a translation router, such as translation router **15** from Figure **1**. Translation template cache **21** contains translation templates for facilitating the translation of data packets from one network protocol to another. Basically, each translation template within translation template cache **21** represents a unique network protocol supported by the translation router. The use of translation templates requires far lesser instructions to perform a network protocol translation as compared to using general purpose processor **25**.

[016] Translation engine **22** may be implemented as software, firmware, hardware, or combinations thereof, and is capable of executing operations necessary to manipulate information in packet headers for protocol translations. Translation template cache **21** may be implemented in a dedicated on-chip memory, whether as one or more locked cache lines or a private memory accessible to translation engine **22**.

[017] Referring now to Figure **3a**, there is illustrated a block diagram of translation template cache **21**, in accordance with a preferred embodiment of the present invention. As shown, translation template cache **21** contains multiple entries, each entry contains a translation template associated with a network protocol supported by a translation router, such as translation router **15** from Figure **1**. Each translation template is preferably 64 bytes wide, which should be long enough to contain headers of data packets of various network protocols.

[018] A translation template is required for each protocol translation. For example, consider a network A and a network B having a protocol A and a protocol B, respectively. One translation template is required for generating data packets suitable for transmission in network B from data packets received from network A. Another translation template is required to generate data packets suitable for transmission in network A from packets received from network B. If there is a third network C having a network protocol C needed to be supported, four more translation templates must be constructed and loaded into the translation template cache. Basically, two translation templates are required for protocol translations between network A and network C, and two translation templates are required for translations between network B and network

C.

[019] Five different translation templates associated with their respective network protocols are shown in Figure **3b**. As shown, entry 1 is a translation template for Fibre Channel frame type #1, entry 2 is a translation template for Fibre Channel frame type #2, entry 3 is a translation template for InfiniBand local routing, entry 4 is a translation template for InfiniBand global routing, and entry 5 is a translation template for TCP/IP.

[020] The details of a translation template can be illustrated with an example. The control and information bits of a typical Fibre Channel frame header include:

[021] D_ID: destination identification (ID)

[022] S_ID: source ID

[023] TYPE: protocol of frame content (*e.g.*, 0000 1000 = SCSI FCP)

[024] F_CTL: flow control

[025] SEQ_ID: sequence ID

[026] DF_CTL: data_field control

[027] SEQ_CNT: sequence count

[028] OX_ID: originator exchange ID

[029] R: responder exchange ID

[030] Parameter: carries information specific to Link_Control frames

[031] Thus, a translation template for facilitating the translation of a received data packet to a Fibre Channel protocol should include most, if not all, of the above-mentioned control and information bits. With reference now Figure **4**, there is illustrated a block diagram of a translation template for facilitating the translation of a received data packet to a Fibre Channel protocol, in accordance with a preferred embodiment of the present invention. As shown, a translate template **40** includes a destination ID field **41**, a source ID field **42**, a type field **43**, a flow control field **44**, a sequence ID field **45**, a data control field **46**, a sequence count field **47**, an originator exchange ID field **48**, and a responder exchanged ID field **49**. As shown, the fields within translation template **40** correspond to the fields within the Fibre Channel frame header.

[032] Referring now to Figure **5**, there is depicted a high-level logic flow diagram of a method for translating data packets from one protocol to another by using apparatus **20** from Figure **2**, in accordance with a preferred embodiment of the present invention. Starting at block **30**, a group of translation templates are constructed and the translation templates are loaded into a translation template cache, as shown in block **31**. The translation templates, such as translation template **40** from Figure **4**, are preferably constructed during initial configuration of a system, and the translation templates are then loaded into the translation template cache. Otherwise, the translation templates can be constructed "on-the-fly" as they are being loaded into the translation template

cache.

[033] Next, exchange attributes are identified by a translation router, such as translation router **15** from Figure **1**, as depicted in block **32**. At this time, network protocols are also identified, usually as part of the translation engine configuration, and the translation templates are verified as available for use. As mentioned previously, a multi-protocol translation router is capable of identifying the type of network protocol to which a network packet belongs based on the incoming port number the network packet comes from, and is also capable of identifying the type of network protocol to which a network packet should be translated based on the outgoing port number the network packet is to be transmitted.

[034] As each data packet arriving into the translation router, an appropriate translation template is selected from the translation template cache according to the translation context of the data packet by an translation engine, as shown in block **33**.

[035] Appropriate header fields from the data packets from a first network are read and used by the translation engine along with the appropriate translation template in the translation template cache to generate new headers for transmission into a second network, as depicted in block **34**.

[036] The data payload of the data packet from the first network is removed from the header of the data packet, and is then appended to the recently constructed header for the second network, as shown in block **35**. The resultant data packet intended for the second network is then placed into a local memory of the network router.

[037] The data packet intended for the second network is subsequently pulled out of the local memory within the network router. This is usually performed by using a direct memory access (DMA) or by the general-purpose processor, as depicted in block **36**. The data packet intended for the second network is sent to the second network.

[038] As has been described, the present invention pro□vides an improved method and apparatus for translating data packets between different network protocols. Control blocks are constructed in a system memory to facilitate translation from one protocol domain to another. They must be repeatedly accessed as part of the translation process. With the present invention, the control blocks are stored on-chip in a manner that fa-cilitates repeated use over a number of data packets. Further, the on-chip storage is not bound to one protocol format or another. In this way, packets may be translated across such protocols as Fibre Channel, Ethernet and InfiniBand.

[039] It is also important to note that although the present invention has been described in the context of a fully functional translation router, those skilled in the art will appreciate that the mechanisms of the present invention are capable of being distributed as a program product in a variety of forms, and that the present invention applies equally regardless of the particular type of signal bearing media utilized to

actually carry out the distribution. Examples of signal bearing media include, without limitation, recordable type media such as floppy disks or CD ROMs and transmission type media such as analog or digital communi□cations links.

Claims

[001] A method for translating data packets from one network protocol to another, said method comprising: constructing a plurality of translation templates; loading said plurality of translation templates into a translation template cache; in response to a data packet from a first network arriving at a translation router, selecting an appropriate one of said plurality of translation templates from said translation template cache according to the translation context of said data packet; generating a new header for transmission into a second network by reading header fields of said data packet from said first network along with said appropriate one of said plurality of translation templates in said translation template cache; removing data payload of said data packet from said first network from its header; appending said data payload of said data packet to said generated header for said second network; and transmitting said data packet to said second network.

[002] The method of Claim 1, wherein said plurality of translation templates includes translation templates for Fibre Channel, translation templates for Ethernet and translation templates for InfiniBand.

[003] The method of Claim 1, wherein said translation template cache is a dedicated on-chip memory.

[004] The method of Claim 1, wherein said selecting further includes selecting an appropriate one of said plurality of translation templates from said translation template cache according to an incoming port number from which said data packet comes.

[005] The method of Claim 1, wherein said generating further includes generating a new header for transmission into a second network according to an outgoing port number to which said data packet is to be transmitted.

[006] An apparatus for translating data packets from one network protocol to another, said apparatus comprising: means for constructing a plurality of translation templates; means for loading said translation templates into a translation template cache; in response to a data packet from a first network arriving into a translation router, means for selecting an appropriate translation template from said translation template cache according to the translation context of said data packet; means for generating a new header for transmission into a second network by reading header fields of said data packet from said first network along with said appropriate translation template in said translation template cache; means for removing data payload of said data packet from said first network from its header; means for appending said data payload of said data

packet to said generated header for said second network; and means for transmitting said data packet to said second network.

[007] The apparatus of Claim 6, wherein said wherein said plurality of translation templates includes translation templates for Fibre Channel, translation templates for Ethernet and translation templates for InfiniBand.

[008] The apparatus of Claim 6 wherein said translation template cache is a dedicated on-chip memory.

[009] The apparatus of Claim 6, wherein said means for selecting further includes means for selecting an appropriate one of said plurality of translation templates from said translation template cache according to an incoming port number from which said data packet comes.

[010] The apparatus of Claim 6, wherein said means for generating further includes means for generating a new header for transmission into a second network according to an outgoing port number to which said data packet is to be transmitted.

[011] A computer program product residing on a computer usable medium for, when run on a data processing system, instructing the system to carry out the method of any of claims 1 to 5.

[Fig.]

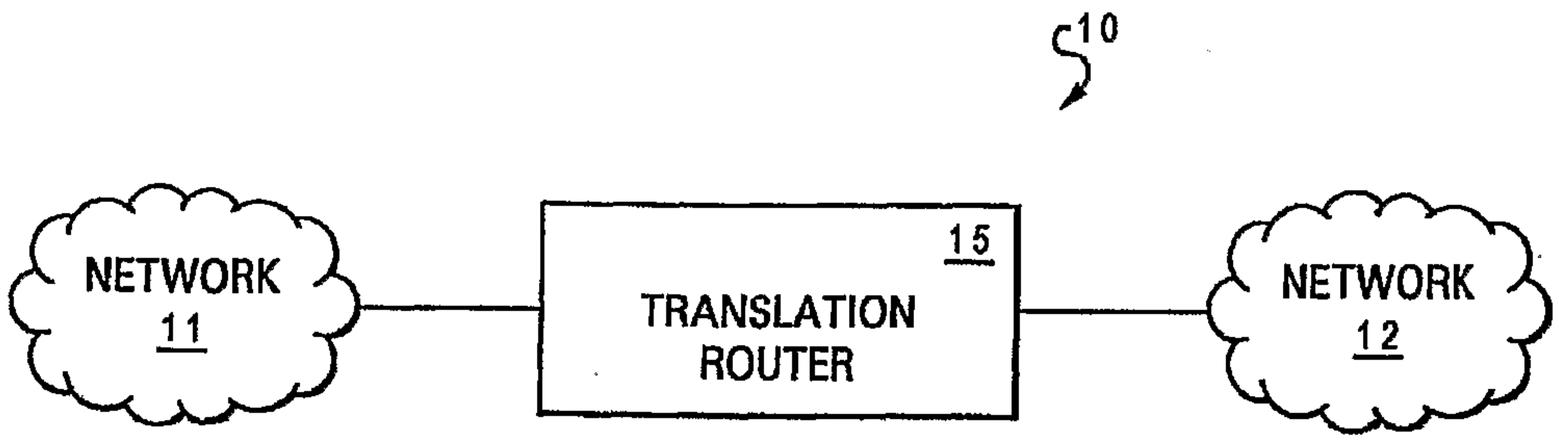


Fig. 1

[Fig.]

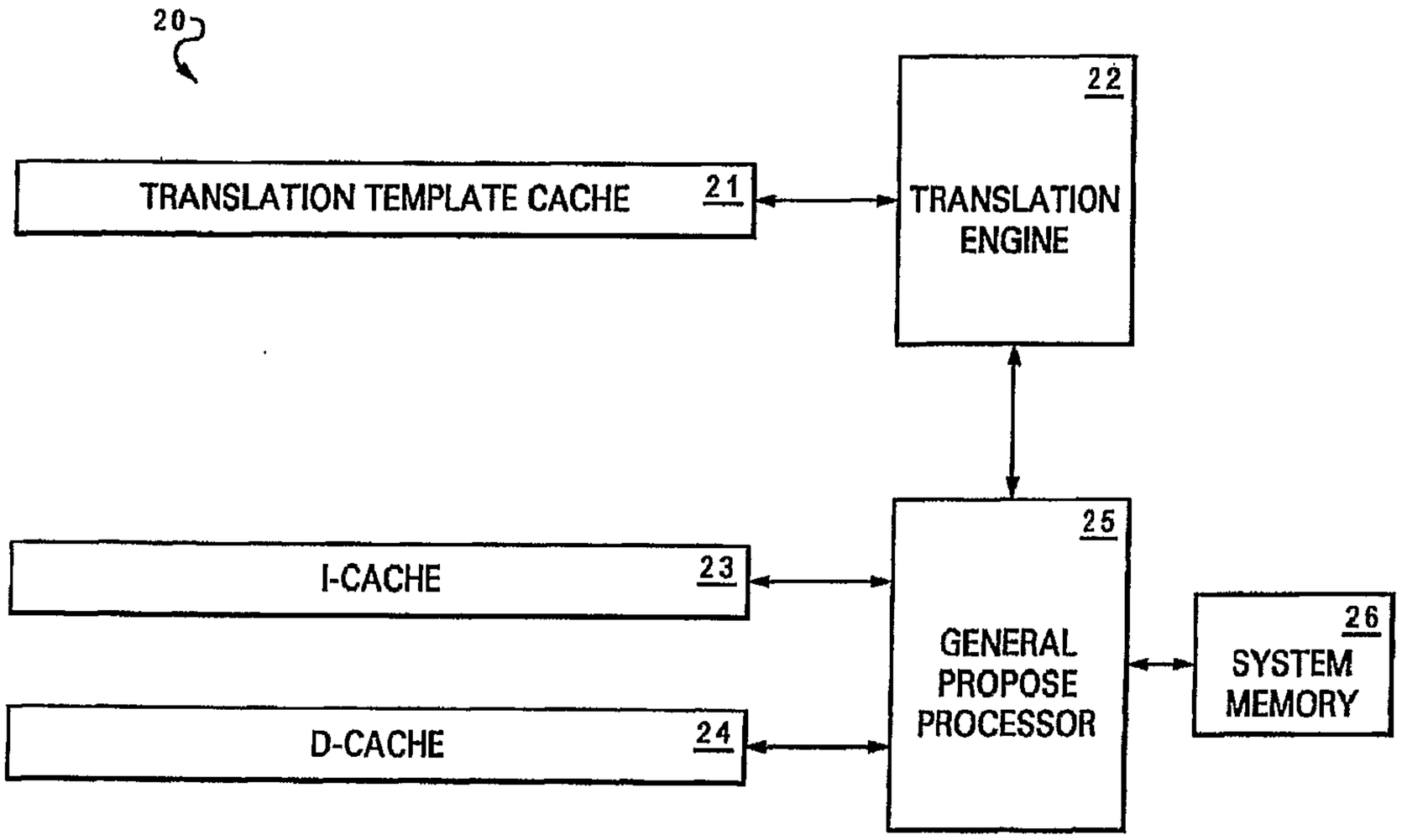


Fig. 2

[Fig.]

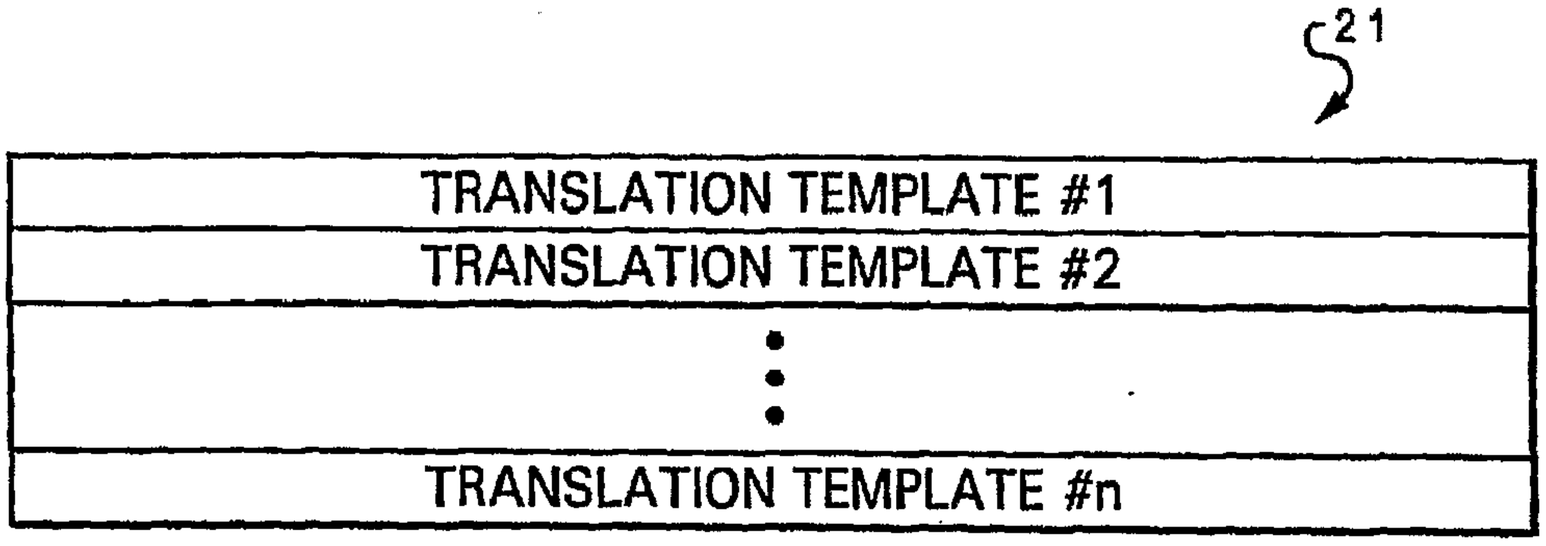


Fig. 3a

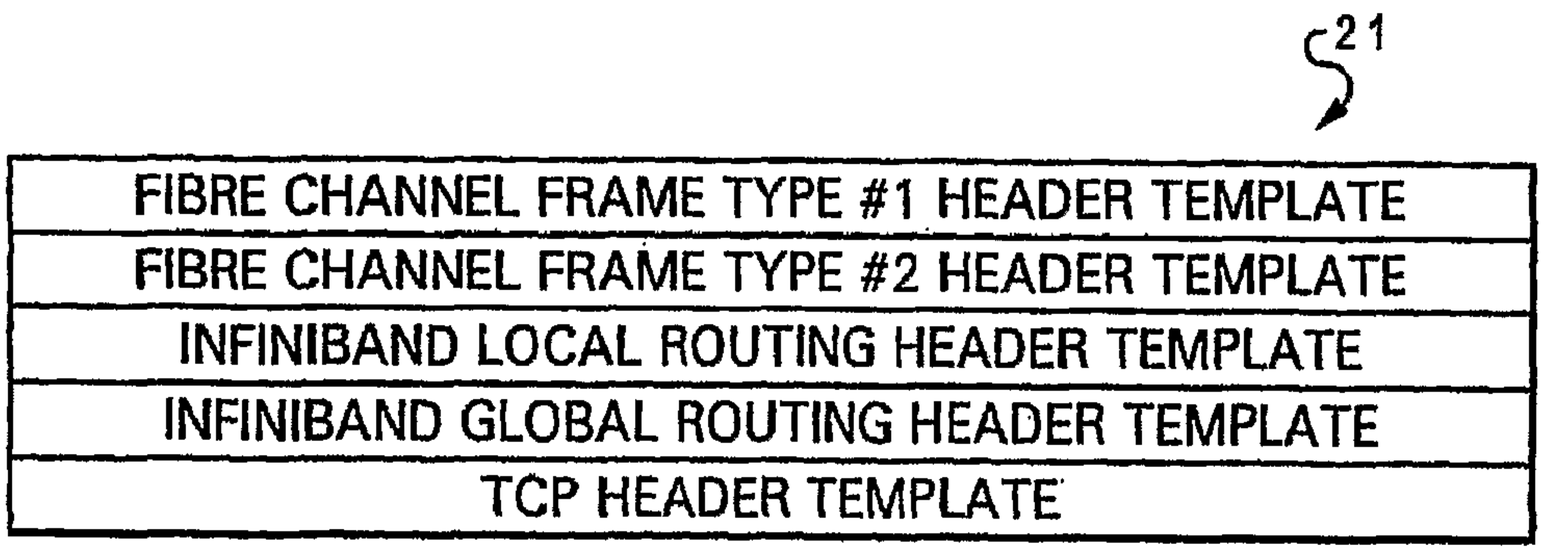


Fig. 3b

[Fig.]

40

41 DEST. ID	42 SOURCE ID	43 TYPE	44 FLOW CONTROL	45 SEQ. ID	46 DATA CONTROL	47 SEQ. COUNT	48 ORG. EXG. ID	49 RESP. EXG. ID

Fig. 4

[Fig.]

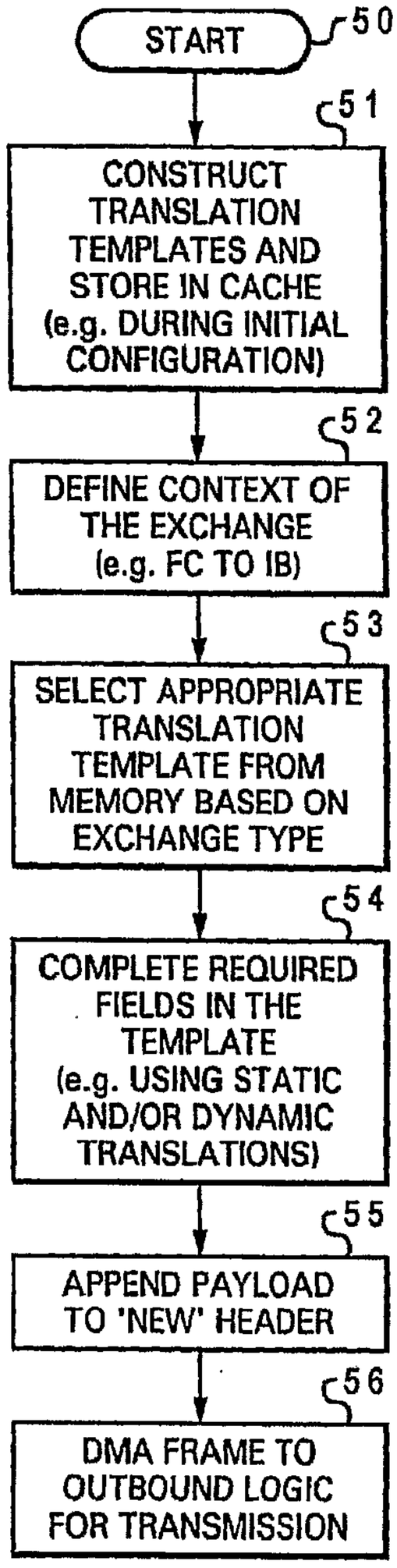
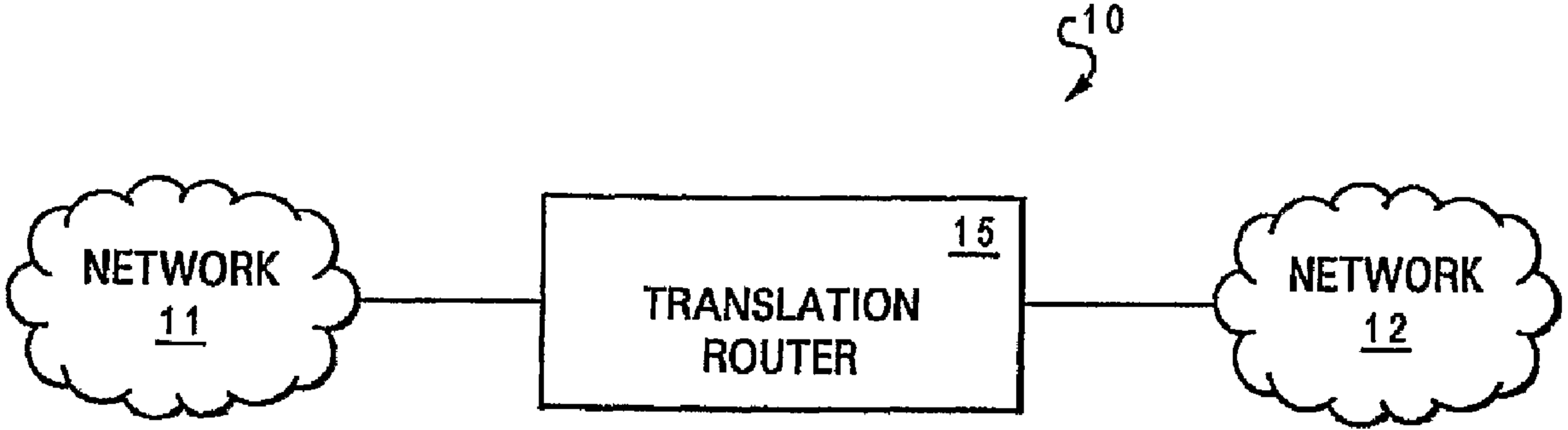

START
50
51
CONSTRUCT TRANSLATION TEMPLATES AND STORE IN CACHE (e.g. DURING INITIAL CONFIGURATION)
52
DEFINE CONTEXT OF THE EXCHANGE (e.g. FC TO IB)
53
SELECT APPROPRIATE TRANSLATION TEMPLATE FROM MEMORY BASED ON EXCHANGE TYPE
54
COMPLETE REQUIRED FIELDS IN THE TEMPLATE (e.g. USING STATIC AND/OR DYNAMIC TRANSLATIONS)
55
APPEND PAYLOAD TO 'NEW' HEADER
56
DMA FRAME TO OUTBOUND LOGIC FOR TRANSMISSION

Fig. 5


10
NETWORK
11
15
TRANSLATION
ROUTER
NETWORK
12