(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 2 031 796 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
04.03.2009 Bulletin 2009/10

(51) Int Cl.:
H04L 12/18 *(2006.01)*

(21) Application number: **07115344.9**

(22) Date of filing: **30.08.2007**

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IS IT LI LT LU LV MC MT NL PL PT RO SE SI SK TR
Designated Extension States:
AL BA HR MK RS

(71) Applicant: **NTT DoCoMo, Inc.**
Tokyo (JP)

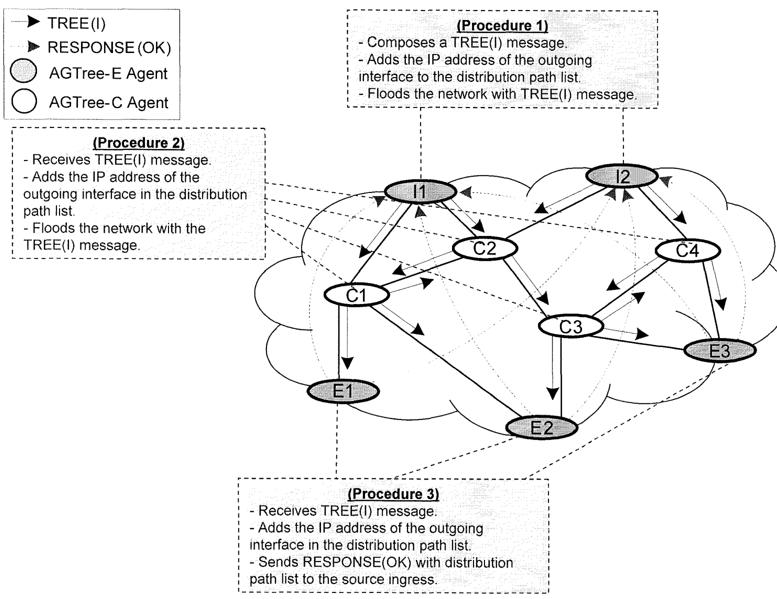
(72) Inventors:
- **Neto, Augusto**
  3030-049, Coimbra (PT)
- **Curado, Marilia**
  3040-688, Coimbra (PT)
- **Monteiro, Edmundo**
  3040-688, Coimbra (PT)
- **Mendes, Paulo**
  81673, Munich (DE)

(74) Representative: **Betten & Resch**
Patentanwälte
Theatinerstrasse 8
(Fünf Höfe)
80333 München (DE)

Remarks:
Amended claims in accordance with Rule 137(2) EPC.

(54) Method and apparatus for multicast tree allocation

(57) A method for multicast tree allocation in a network, said method comprising:
creating during an initialisation phase a plurality of possible multicast distribution trees, wherein said plurality of possible multicast distribution trees is created such that for each of a plurality of combinations of egress nodes and a given ingress node there has been created a corresponding distribution tree;
switching of distribution trees by switching to another distribution tree selected from the distribution trees which have been generated during said initialisation phase, wherein said switching is performed in response to an egress node joining or leaving a session.



(Procedure 1)
- Composes a TREE(I) message.
- Adds the IP address of the outgoing interface to the distribution path list.
- Floods the network with TREE(I) message.

(Procedure 2)
- Receives TREE(I) message.
- Adds the IP address of the outgoing interface in the distribution path list.
- Floods the network with the TREE(I) message.

(Procedure 3)
- Receives TREE(I) message.
- Adds the IP address of the outgoing interface in the distribution path list.
- Sends RESPONSE(OK) with distribution path list to the source ingress.

Paths from I1	
Index	Nodes
0	C1, E1
1	C1, E2
2	C2, C1, E1
3	C2, C1, E2
4	C2, C3, E2
5	C2, C3, E3
6	C2, C3, C4, E3
7	C1, C2, C3, E3
8	C1, C2, C3, E2
9	C1, C2, C3, C4, E3

Paths from I2	
Index	Nodes
0	C4, E3
1	C2, C1, E1
2	C2, C1, E2
3	C2, C3, E2
4	C2, C3, E3
5	C4, C3, E2
6	C4, C3, E3
7	C2, C3, C4, E3
8	C4, C3, C2, C1, E2
9	C4, C3, C2, C1, E1

Discovering Available Distribution Paths

Fig. 3

Printed by Jouve, 75001 PARIS (FR)

Description

FIELD OF THE INVENTION

[0001] The present invention relates to a method for multicast tree allocation in a network..

BACKGROUND OF THE INVENTION

[0002] Next generation networks are expected to simultaneously distribute the session content to multiple users, independent of the connectivity control schemes supported inside or between the networks along the distribution path. The multi-user sessions (e.g., IPTV, radio, file distribution and push media), will be available anywhere and anytime for users attached to networks with different access technologies. The accommodation of a high number of simultaneous multi-user sessions and the assurance of their continuity along their entire lifetime without jeopardising the system scalability requires efficient network provision.. With this in mind, the control scheme must be chosen taking into account the current network capability. For instance, per-flow approaches which operate on-demand for configurations may jeopardise the network performance through signalling and state overhead

[0003] From the transport point of view, multicast is the most suitable technology for group communication, since it allows several users to get the content of the subscribed sessions at the same time without packet duplication, which optimises the utilisation of the network bandwidth.. From the routing point of view, the session path must be computed from the source to the receivers, since the best routes are computed in this direction. However, the multicast resources are controlled in the opposite direction (i.e., from the receiver to the sources), which jeopardizes, for instance, the creation of Quality of Service (QoS) aware multicast trees, once the QoS path is computed in the opposite direction of the multicast processing. This way, asymmetric routes may influence the creation of the multicast tree in a path different than the QoS path. Another drawback of multicast is its per-flow (per tree) basis, which affects negatively the system performance in terms of state storage. For example, the performance of packet forwarding processing is overloaded with the increasing number of active multicast trees, overloading the look-up operations for packet replication.. Therefore, the multicast resources must be efficiently provisioned to ensure the distribution of all session-flows to the member users without damage the system performance..

[0004] Dynamic multicast provisioning fits better in high used networks than static provisioning, since the resources are configured on-demand and automatically. For instance, this approach avoids constant human interventions to implement the required configurations and updates. However, the network performance may be negatively affected in terms of bandwidth consumption, since signalling messages are required to inform the agents along the path about the resources which need configuration.

[0005] Once the distribution trees are assigned per-flow, they must be over-provisioned carefully so they don't jeopardise the system scalability in terms of state storage (e..g., overloading the forwarding nodes).

[0006] The state storage may be optimised by implementing aggregation, which allocates into a single distribution tree the sessions which match some condition, such as when sharing the same segment to distribute packets.. With this in mind, two main nodes manage the connectivity between the incoming flows and the aggregation trees, the aggregator and the deaggregator, which are generally placed at the root and leaf nodes respectively. Therefore, the forwarding nodes only store state regarding aggregation trees, instead of per-flow distribution trees.. For example, the packet forwarding processing is optimised by significantly decreasing the number of options in the look-up operations for packet replication in multicast aware networks.

[0007] Most solutions proposed by the research community that optimise signalling and state overhead through over-provisioning, are centred in QoS resources, such as by implementing bandwidth over-reservation. Available solutions which support the over-provisioning of multicast trees create surplus multicast resources aiming to protect against link breaks, node failures and member group leaves.. The solutions presented e.g. in T. Braun, V. Arya, and T. Turletti, "A Backup Tree Algorithm for Multicast Overlay Networks", Networking 2005, Waterloo, Canada, May 2005. and M. Kodialem and T. Lakshman, "Dynamic routing of bandwidth guaranteed multicasts with failure backup", IEEE INFOCOM, New York, USA, Jun, 2002, build a primary and a backup tree to each request (i..e., in a per-flow basis), taking into account resource availability. The configuration of the resources in the new path is fast since the backup tree is created in advance, and moreover the quality of the section may not be damaged. However, these approaches jeopardise the system scalability since they control the creation of the multicast trees in a per-flow basis. Therefore, whereas the system robustness is enhanced, the system scalability is seriously affected once the amount of multicast state is twice the number of active flows.

[0008] The solution presented in A. Striegel and G. Manimaran, "DSMCast: a scalable approach for DiffServ multicasting", Computer Networks, vol. 44, New York, NY, USA, Apr. 2004, proposes an application layer multicast in DiffServ domains, where stateless core routers forward the data packets based on extra headers. The information filled in the extra headers carries identification for core nodes, appropriate branching information, tunnelling bit to bypass DiffServ-non-capable nodes, and adaptive DiffServ field to adapt to the heterogeneous DSCP requirements by the different receivers. However, this solution is based on unicast connections to distribute session packets, not supporting thus IP multicast.

[0009] The solution presented in J. Cui et al, "Aggregated Multicast - A Comparative Study", Cluster Computing, vol. 8, Hingham, MA, USA, Jan.2005 proposes a novel scheme to improve multicast state scalability.. In this scheme, multiple multicast groups are forced to share one distribution tree, called aggregated tree, significantly reducing the number of trees in the network. Consequently the forwarding state is also reduced since the core routers only need to keep state per aggregated tree instead of per group. The solution considers perfect, leaky and incomplete matches of multicast trees. This way, although reducing the multicast state overhead, this approach do not ensure waste of bandwidth avoidance, once leaky and incomplete matches enable delivering multicast data to non-group-member nodes. In addition, the solution may endanger the network performance in terms of signalling overhead, where the network must be signalled to set-up state in appropriate routers on demand.

SUMMARY OF THE INVENTION

[0010] According to one embodiment there is provided method for multicast tree allocation in a network, said method comprising:

> creating during an initialisation phase a plurality of possible multicast distribution trees, wherein said plurality of possible multicast distribution trees is created such that for each of a plurality of combinations of egress nodes and a given ingress node there has been created a corresponding distribution tree;
> switching of distribution trees by switching to another distribution tree selected from the distribution trees which have been generated during said initialisation phase, wherein said switching is performed in response to an egress node joining or leaving a session.

[0011] The creation of multiple multicast distribution trees during an initialisation phase, in other words "in advance", leads to an overprovisioning of connectivity resources which can be used for implementing an efficient adaptation mechanism which can adapt to changing circumstances without much signalling being required for providing the necessary connectivity resources.. In this way the mechanism can adapt to changes in the set of egress nodes which participate in the session to achieve an optimum distribution path in any situation.

[0012] According to one embodiment all possible multicast distribution trees are generated during the initialisation phase. This means that for each combination (or for each "realistic" combination) of egress nodes and a given ingress node there is a corresponding multicast distribution tree which has been generated and which can be used to switch between the trees in order to adapt to a change in the set of egress nodes, e.g. by a node joining or leaving a session. The term "all possible" distribution trees therefore should be understood such that it not necessarily encompasses all theoretically possible distribution trees (although this may be possible as well) but such that from this reservoir of "all theoretically possible distribution trees" only those which may from a practical point be chosen are included. This may e.g. exclude some distribution trees for efficiency reasons, e.g. trees where an ingress node or an edge node would form an intermediate node of the tree.

[0013] The creation of multicast distribution trees during initialisation therefore leads to a "reservoir" of distribution trees or a "overprovisioning" of multiple multicast distribution trees which have different combinations of egress nodes for a given ingress node and from which the most suitable one may be chosen and between which a switching can be performed depending on the circumstances, like depending on the egress nodes participating in a session.

[0014] According to one embodiment said multicast trees comprise multicast aggregation trees in which multiple multicast groups are sharing one distribution tree..

[0015] This enables the usage of "ramified" trees of aggregation trees in which multiple multicast groups are sharing one distribution tree.

[0016] According to one embodiment the method comprises:

> controlling the session connectivity by selecting a suitable aggregation tree such that the selected aggregation tree has no empty leave nodes formed by an egress node not participating in the session, and further such that there is a leave node in the selected aggregation tree for each node participating in the session.

[0017] This enables the selection of an aggregation tree which uses the available resources in an optimum manner.

[0018] According to one embodiment said generation of distribution trees further comprises the filtering of the paths, wherein said filtering comprises the selection of the best paths or the discarding of inefficient paths.

[0019] This increases the efficiency of the mechanism.

[0020] According to one embodiment said filtering comprises: discarding paths in which ingress routers are within the forwarding nodes, and/or discarding the paths with a branching point in the ingress routers.

[0021] This avoids particularly inefficient paths.

[0022] According to one embodiment the method comprises:

> creating during an initialisation phase all possible distribution trees such that if a session requires a certain distribution path from an ingress node or a source node to one or more egress nodes, the corresponding distribution path has already been created during said initialisation phase.

[0023] The creation of a reservoir of distribution paths in advance during an initialisation phase makes it possible later during usage of a session to select the most suitable distribution path from this reservoir without large signalling overhead or loss of time due to processing..

[0024] According to one embodiment each distribution tree which has been created is identified by a multicast tuple composed by the source IP address being the address of the ingress router which is the enter point of the session-flows and a destination IP address, being an IP multicast address..

[0025] This enables the usage of the created trees or paths in a multicast aware environment..

[0026] According to one embodiment the method comprises discovering the available distribution paths within the network and creating the un-ramified aggregation trees..

[0027] The (unramified) aggregation trees allow efficient usage of resources.

[0028] According to one embodiment the method comprises: flooding the network during initialisation by the ingress nodes of said network with a TREE message, and upon receiving the TREE message, a router which has received said TREE message propagates the message on each outgoing interface except the one in which the message was received, wherein each message carries the local IP address, filled in the distribution path list, of the outgoing interface used for propagation, said method further comprising:

> after receiving the TREE message, an egress node composes a RESPONSE message corresponding to the TREE message received and sends it to the ingress router which is source of the equivalent TREE message, said RESPONSE message carrying the distribution path list of the equivalent TREE message and the local outgoing interfaces.

[0029] This enables the discovery or creation of the distribution trees.

[0030] According to one embodiment said initialisation phase further comprises:

> creating the un-ramified multicast aggregation trees, said creation comprising:
>
> > after having allocated a multicast channel to a distribution path, sending a TREE message along the distribution path, wherein each core node adds in the MRIB the IP address of the previous node and propagates the message downstream, said creation further comprising:
> >
> > > after receiving the TREE message, updating by the egress node the MRIB with the IP address of the previous router and triggering the multicast routing protocol for creating the multicast tree.

[0031] In this manner the unramified aggregation trees can be generated.

[0032] According to one embodiment said initialisation phase further comprises:

> computing and creating the ramified multicast aggregation trees that cover more than one egress node, wherein the ramified trees are computed by combining all un-ramified trees taking into consideration the ingress router and the available egress routers,
>
> allocating a multicast channel to each of the ramified multicast aggregation trees.

[0033] In this manner a set of ramified aggregation trees can be generated which can then be used for connectivity control to provide an efficient connectivity control.

[0034] According to one embodiment the method comprises: discarding all trees with branching point at the root, and
sending a signalling message to all trees in order to configure the multicast trees.

[0035] This enables the efficient generation of the ramified aggregation trees.

[0036] According to one embodiment the method comprises:

> controlling the session connectivity in order to propagate the one or more session-flows into the generated distribution trees such as to avoid a waste of resources.

[0037] In this manner there can be made use of the distribution trees generated in advance in order to achieve a particularly efficient connectivity control with short response time and low signalling overhead.

[0038] According to one embodiment there is provided an apparatus for multicast tree multicast tree allocation in a network, said apparatus comprising:

> a module for creating during an initialisation phase a plurality of possible multicast distribution trees, wherein said plurality of possible multicast distribution trees is created such that for each of a plurality of combinations of egress nodes and a given ingress node there has been created a corresponding distribution tree;
>
> a module for switching of distribution trees by switching to another distribution tree selected from the distribution trees which have been generated during said initialisation phase, wherein said switching is performed in response to an egress node joining or leaving a session.

[0039] According to one embodiment the apparatus further comprises:

a module for performing a method according to one of the embodiments of the invention.

[0040] A computer program comprising computer program code which when being executed on a computer enables said computer to carry out a method according to one of the embodiments of the invention.

DESCRIPTION OF THE DRAWINGS

[0041]

Fig.. 1 schematically illustrates an environment where an embodiment of the invention may be used.
Fig. 2 schematically illustrates the interfaces of a node according to an embodiment of the invention..
Fig. 3 schematically illustrates a distribution path generation method according to an embodiment of the invention.
Fig. 4 schematically illustrates a distribution path generation method according to a further embodiment of the invention.
Fig. 5 schematically illustrates a distribution path generation method according to a further embodiment of the invention.
Fig. 6 schematically illustrates a state diagram of an embodiment of the invention.
Fig.. 7 schematically illustrates a multicast tree allocation according to an embodiment of the invention..

DETAILED DESCRIPTION

[0042] Before describing embodiments of the invention some terms which will be used in the following description are explained.

[0043] Access router - Router linked to the access network in which a receiver belongs to;

AS - Autonomous Systems;
BR - Border Router (ingress or egress);
Downstream - Direction from the ingress node to an egress or access node;
Upstream - Direction from the egress node to an ingress node;
Unicast Routing Information Base - The unicast topology table;
Multicast Routing Information Base - The multicast topology table typically derived from unicast RIB. PIM protocols use the MRIB to decide where to send join/prune messages; Reserved Path - The path in which a Reserve message has controlled resources.

[0044] In the following there will be described embodiments of a mechanism to over-provision distribution trees (considered here as connectivity resources).. The mechanism and its embodiments may be referred to in the following as Advanced Aggregation Tree Allocation (AGTree). The AGTree mechanism in one embodiment controls the aggregation of the incoming session-flows and dynamically allocates them into distribution trees which supply all their member users, increasing the system performance significantly reducing signalling and state overhead.

[0045] The Advanced Aggregation Tree Allocation (AGTree) is a mechanism for over-provisioning connectivity resources (i.e., distribution trees) and dynamically controlling the allocation of unicast and multicast session-flows into aggregation trees. The AGTree mechanism in one embodiment fills the gap of the signalling overhead resulting from on-demand approaches, by combining over-provisioning of connectivity resources and admission control, where no per-request signalling is performed. The state overhead due to per-flow approach is overcome by implementing aggregations and, combined with a dynamic session connectivity control, avoids waste of resources due to the over-provisioning. Furthermore, the AGTree mechanism avoids problems due to asymmetric routes by indicating the session path in the *Multicast Routing Information Base* (MRIB) during the resource connectivity over-provisioning work..

[0046] In the AGTree mechanism, according to one embodiment the distribution trees are over-provisioned in the system bootstrap, in order to prevent system performance degradation in terms of bandwidth consumption (due to signalling) and processing (due to state manipulation) to configure the resources.. Instead of per-flow, the AGTree mechanism creates the distribution trees in advance taking into consideration the number of possible associations which can be established between each root node and the available leaf nodes. In this context, the root nodes are the enter nodes of the sessions (i.e., ingress router) and the leaf nodes are the exit node of the network (i.e., egress router or the access point of the user, denoted as access router).. In unicast domains, each distribution tree (called un-ramified tree) is composed by a pair of root and leaf node and a set of core nodes between them, since no packet duplication is accomplished in the forwarding nodes. On the other hand, in multicast domains, the distribution trees may support several leaf nodes, where the packets are duplicated at routers called branching points (called ramified when supporting branching points).

[0047] The AGTree mechanism in one embodiment controls the session connectivity by dynamically allocating the session-flows into aggregation trees. All session-flows which share the same distribution path, independently of the class of service (CoS) in which they are assigned to, are aggregated into the same distribution tree which covers the entire path. Therefore, the AGTree mechanism creates aggregation trees in advance based on the number of ingress-to-egress(s) associations, instead of depending on large and dynamic parameters, such as the number of session/flow requests. The AGTree mechanism in one embodiment is able to allocate unicast and multicast traffic into the same aggregation tree, being perfectly matched when all its leaf nodes have members of the indicated session. In one embodiment, by default the AGTree mechanism allocates the incoming session-flows into the aggregation trees through a tunnel-based approach An embodiment of such an approach will be described in more detail later.

[0048] The AGTree mechanism in one embodiment avoids the distribution of session content to leaves of the tree without group members by controlling on demand the allocation of aggregation trees. Whenever the AGTree mechanism receives a request, it locally searches for another aggregation tree that covers the distribution path required by the indicated session and merges it into the matched tree, without affecting the session connectivity. When AGTree receives a request to an active session and realises that the destination is not covered by the current allocated tree, the session is switched to another aggregation tree with leaves for all active members, including the new one. According to one embodiment a similar approach is also implemented when a single user, attached to a leaf of the aggregation tree, leaves the session. This local dynamic switching scheme, combined with creation of aggregation trees in advance, allows fast recovery from failures (e.g., re-routing) in comparison to schemes that require signalling exchanges for rerouting from the source or from some point further away from the failed element.

[0049] The AGTree mechanism controls the admission of sessions by checking the local availability of perfectly matched distribution trees. By combining the creation of distribution trees in advance and admission control, the AGTree mechanism decreases the signalling overhead by avoiding on-demand message exchange for configuring the connectivity resources. Moreover, the network performance is increased through traffic aggregation and session connectivity control, where the impact of per-flow multicast state storage in the packet forwarding processing is reduced. According to one embodiment the mechanism comprises the following components or operations:

- Each receiver gets the available sessions from the sender(s) by any off-line or on-line means;
- Each receiver interested in joining a session, requests it to an agent in the network to which it is connected at the session set-up time (denoted access router);
- A protocol or mechanism in the network agent requests connectivity resource (i.e., the allocation of a distribution path) for each session-flow in each domain by indicating the IP address of the access router in which the interested user is attached to;
- An external mechanism is responsible to control the connectivity from the access router to the member users though notifying the multicast channel that must be joined/left. For example, the application in the member user triggers the IGMPv3 (see e.g. B. Cain *et al,* "Internet Group Management Protocol, Version 3", *IETF,* RFC 3376, Oct 2002) or MLDv2 (see e..g.. R. Vida and L. Costa, *"Multicast Listener Discovery Version 2 (MLDv2) for IPv6", IETF,* RFC 2710, Jun 2004). Moreover, this application may trigger the AGTree mechanism in order to inform when a session has no member users in an access router. For instance, the mechanism called SEACON is able to inform when no active member user are available. Such a mechanism called SEACON is described in European Patent application number 07102594.4 titled "Method and apparatus for session aware connectivity control" which was filed by the same applicant as the present one at the EPO on Feb. 16, 2007, and which is incorporated herein by reference. For further details of the SEACON mechanism reference is made to this application;
- A range of communication ports, Registered or Dynamic and/or Private (see e.g. E. Kohler, M. Handley and S. Floyd, "Datagram Congestion Control Protocol (DCCP) ", IETF, RFC4340, Mar 2006), should be available and filed in a pool of ports;
- Routers according to one embodiment recognise IP Routing Alert Option [see e.g. RFC2113].

[0050] The functionalities supported by the AGTree mechanism in one embodiment are implemented by agents located at the edge and core routers, respectively called AGTree Edge agent (AGTree-E) and AGTree Core agent (AGTree-C). The AGTree-E agents are statefull in order to control the admission of the session-flows and initiate the operations to control the connectivity resources in the entire network. An AGTree-E agent according to one embodiment implements internal and external interfaces, in order to, for instance, interact with other AGTree agents and to communicate with external protocols or mechanism. In contrast, AGTree-C agents according to one embodiment have reduced state, storing per-class ingress IP address and multicast channel (when applicable), supporting only internal interfaces to configure the connectivity resources upon signalled. Thus, the system scalability is achieved by pushing the complex operations to control the resources to the edge nodes, and leaving the core nodes simple..

[0051] In what concerns the multicast approach supported by an environment, AGTree may interact with central or distributed entities. Figure 1 illustrates a scenario composed by three network clusters, where domains A and B are multicast-aware, implementing PIM-SSM (see e.g. S. Bhattacharyya, "An Overview of Source-Specific Multicast (SSM)", IETF, RFC3569, Jul 2003) and PIM-SM (see e..g.. D. Estrin et al, "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification", IETF, RFC2362, Jun 1998) respectively, and C supports unicast. Th e AGTree mechanism in one embodiment operates in a distributed manner to control the connectivity resources, where the multicast routing protocol is triggered at the egress routers to create the multicast trees. On the other hand, domain C is unicast and the distribution trees are maintained by AGTree-E agents at the ingress nodes.

[0052] According to one embodiment, in order to implement the functionalities supported by AGTree, the following interfaces (which are schematically illustrated in Fig.. 2)are provided:

- Application Interface: Used by external mechanisms or protocols to trigger the AGTree mechanism with a request, which carries the session object. Moreover, the AGTree mechanism uses this interface to feedback the requester application with the result of the requested operation. For instance, before a session-flow enters the AGTree-aware cluster, an external application uses this interface to request resources (i.e., the distribution path) for the incoming flow. The AGTree mechanism also uses this interface to trigger the requester application in order to confirm the successful accomplishment of the previous request. In addition, the AGTree mechanism can receive indication to re-allocate connectivity resources of ongoing session-flows when new users are interested in the session or due to the movement of the user to another domain;

- Multi-user Control Interface: Used to manage the allocation of unicast and multicast IP addresses and communication ports for allocating the aggregation trees. This interface is also used to create the multicast aggregation trees by configuring the replicators for an assigned outgoing interface (acquired by the Communication Interface). The addition of entries in the MRIB is accomplished through this interface as well as the triggering of the multicast protocol in order to create/maintain multicast trees.. Finally, the AGTree mechanism uses this interface to query for control information, such as to check when a multicast tree is installed or to retrieve a path from the MRIB;

- Communication/routing Interface: Used as support to exchange messages, carrying control information, among AGTree agents as support for the accomplishment of operations. Additionally, the AGTree mechanism identifies the network interface for further processing via this interface, by interacting with the available unicast routing protocol (e.g., OSPF), where the information is retrieved from the Routing Information Base (RIB). The AGTree mechanism in one embodiment is able to detect re-routing events through this interface, for instance, when OSFP is informed by lower-level protocols that a network interface is down (see e..g.. J. Moy, "OSPF Version 2", IETF, RFC2328, Apr 1998).

[0053] Figure 2 shows the interfaces in accordance with the type of the AGTree agent. The exchange messages, which carry control information between AGTree agents, are implemented through the communication interface.

[0054] In the following messages of the AGTree mechanism according to one embodiment are explained.

AGTree Messages:

[0055]

- TREE: This message is used by the AGTree-E to discover the available distribution paths and according to one embodiment to request the creation of the aggregation trees in multicast-aware networks.. For the latter, the message according to one embodiment is filled with the multicast channels as supports for their configuration. Message specific flags may control the behaviour of the AGTree agents, indicating for instance whether an operation must not be performed;

- RESPONSE: This message is used to feedback an AGTree-E agent about the requested operation. The message may contain specific control information, such as the set of nodes which compose a distribution path. Whenever a node is informed by the routing protocol that an interface is down, according ton one embodiment this message is sent to all ingress nodes locally stored, with a specific flag set indicating that a re-routing event was detected.

[0056] According to one embodiment, by default, the AGTree mechanism operates end-to-end to accomplish its functionalities.. However, combined with an external session controller, according to one embodiment the AGTree mechanism may work edge-to-edge, limiting its operation to each network. For instance, a session controller which has an inter-network end-to-end operation may trigger AGTree at the edges of each network (ingress and egress routers) allowing tree aggregation to be processed for that request inside the network an in the inter-network link.

[0057] Embodiments of the AGTree mechanism provide the following advantages:

- Decreases the system complexity and reduces its operational costs by using a unique solution for creating distribution trees in advance, controlling the aggregations and the session connectivity;
- Reduces the session set-up time;
- Reduces significantly the signalling and state overhead.

[0058] In the following embodiments of the AGTree mechanism will be described in somewhat more detail.

[0059] Each AGTree-E agent stores lists of active sessions, flows and the distribution paths. Each active session entry contains the session identifier and a list with the associated flows, and each flow entry keeps its flow identifier, CoS, bit rate, source address, destination address, egress address, access address and the distribution path. Each entry of the list of distribution paths is filled with the IP address of the core nodes which compose the entire path. In multicast-aware networks, AGTree stores multicast state on each outgoing interface, being filled with their associated multicast channels. The AGTree-C agents only store the multicast states when applicable.

[0060] In the following there will be described the functionalities implemented by the AGTree mechanism in order to over-provision connectivity resources.

[0061] In the following the in-advance creation of the distribution trees will be explained. "In advance" here means during an initialisation phase such as system bootstrap, or any initialisation phase which takes place before actually distributing a session.

[0062] All distribution trees (in one embodiment all possible distribution trees from an ingress node to an egress node or any combination of egress nodes or for any combination of ingress and egress nodes) are created by the AGTree mechanism during an initialisation phase, e.g. at the system bootstrap, provisioning thus the environment along its entire lifetime for all future requests without requiring extra signalling for resource configuration. The term "all possible" distribution trees should be understood such that it not necessarily encompasses all theoretically possible distribution trees (although this may be possible as well) but such that from this reservoir of "all theoretically possible distribution trees" only those which may from a practical point be chosen are included. This may e.g. exclude some distribution trees for efficiency reasons, e..g. trees where an ingress node or an edge node would form an intermediate node of the tree. One may therefore say instead of "all possible distribution trees" "all distribution trees from which one may be chosen for the multicast distribution of a session" based on a given set of ingress and egress nodes participating in that session. The intention is to provide a reservoir of distribution paths such that for a (or for any) given combination of ingress and egress nodes participating in the session there is already existing a corresponding distribution path which may be used without any signal overhead being necessary for its creation. In other words, if a session requires a certain distribution path from an ingress node or a source node to one or more egress nodes, the corresponding distribution path has already been created during said initialisation phase. Therefore; according to one embodiment for any given set of receivers (corresponding to a certain set of egress nodes to which they are connected) and a corresponding ingress node (or more of them, if many sessions are running in parallel) there will be a suitable distribution path which may just be chosen for the distribution, and if the set of receivers (or egress nodes) changes, there may be performed a switching to a different distribution path matching with the new set of receivers (or egress nodes). In the same manner a switching may be performed if the ingress node (or the set of ingress nodes) changes..

[0063] Each distribution tree which has been created is identified by a multicast tuple composed by the source IP address (S) and a destination IP address (G). The former is the ingress router which is the enter point of the session-flows, and the later is an IP multicast address which according to one embodiment may be assigned via the Multi-user Control Interface.

[0064] According to one embodiment the initialisation of the distribution trees is accomplished in a single phase in unicast networks and in three phases in multicast networks. The first phase, common to both types of connectivity support, consists in discovering the available distribution paths within the network and creating the un-ramified aggregation trees..

[0065] The distribution paths are discovered by flooding operations initialised by the ingress routers, which consists in flooding the network with a "flooding message" referred to hereafter as TREE message with a specific flag set as *on*, indicating initialisation. Upon receiving the TREE(*I*) message, the router propagates the message on each outgoing interface except the one in which the message was received. Each message carries the local IP address, filled in the distribution path list, of the outgoing interface used for propagation. After receiving the TREE(*I*) message, an egress router composes a RESPONSE message corresponding to the TREE(*I*) message received and sends it to the ingress router which is source of the equivalent TREE(*I*) message. The RESPONSE messages carry the distribution path list of the equivalent TREE(*I*) message, the local outgoing interfaces and a confirmation code (*OK*).

[0066] The flooding operation according to one embodiment is based on the unicast routing protocol, and this way it provides all available distribution paths. Since the benefits derived from using distribution paths (e.g., avoidance of packet duplication) are dependent upon their shape, AGTree according to one embodiment filters the distribution paths and chooses only the best paths. The "filtering" may comprise the selection of the best paths or it may comprise the discarding of inefficient paths, or it may comprise both. The performance of a multicast tree broadly depends upon its topology, where the amount of comprised links and where branching occurs within the tree must be considered. Once long trees results in heavy transmission costs, the AGTree mechanism in one embodiment discards paths in which ingress routers are within the forwarding nodes. Moreover, the distribution trees demands more network resources (e.g., bandwidth) as the branch gets close to their root, increasing thus the amount of packet duplications. Since the worst case of multicast is when the root of a tree is a branching point, AGTree discards the paths with branching point in the ingress routers.

[0067] Figure 3 shows the operations of the first phase for initialising the resources, where the AGTree-E agent at each ingress router (*I1* and *I2*) floods the network with a TREE(*I*). All core nodes reached by the TREE(*I*) message adds in the distribution path list the local IP address of the outgoing interface in which the packet will be sent to and forward it into all interfaces, except the one in which the message was received. After receiving the TREE(*I*) message, each egress router (*E1*, E2 and E3) composes one RESPONSE message, corresponding to each TREE(*I*) message received, adds its local IP address in the distribution path list and sends it to the ingress router which is source of the equivalent TREE(I) message.. The RESPONSE messages carry the distribution path list of the equivalent TREE(*I*) message and a confirmation code (*OK*). When *I*2 receives the TREE(I) message sent by *I1,* no RESPONSE message is sent to *I1* in order to avoid path entries comprised with *I2*.

[0068] Based on the information provided by each RESPONSE(*OK*) message received, the AGTree-E at *I1* and *I2* updates the local distribution path information. AGTree creates the un-ramified aggregation trees by assigning a multicast tuple (S, G) to each distribution path entry (not show in Figure 3 for simplification). The source (root node) is the IP address of the ingress router which received the RESPONSE(*OK*) message, and the destination is a multicast IP address assigned via the Multiuser Control Interface.

[0069] In multicast-aware networks, the initialisation of the distribution trees requires two more phases, where the second phase aims to configure the un-ramified multicast aggregation trees in the nodes along their paths, which is not addressed to unicast networks The AGTree mechanism may control the multicast trees in two ways, by direct configuring the multicast address in the outgoing interface during the end-to-end-signalling or controlling the MRIB entries (in the end-to-end signalling) and triggering the available multicast routing protocol at the egress router so as to create the multicast tree. For the illustration provided by Figure 4, the AGTree mechanism operates in the later method.. Therefore, the AGTree-E agent at each ingress router firstly allocates a multicast channel (destination address and port) to each distribution path entry..

[0070] Figure 4 shows the operations to create the unramified multicast aggregation trees where, for simplification, the signalling events are not extensively depicted and it is considered only the state at *I1*. The table of paths from I1 is composed by the key index (*Index*) of each distribution path, the nodes (Nodes) that compose each path and the IP multicast destination (G) allocated to each path (Gx). The source address is not filled in the table since it is the local IP address (e.g., the IP address of the ingress router). After allocating a multicast channel to a distribution path, AGTree sends a TREE message along the distribution path.. Each AGTree-C adds in the MRIB the IP address of the previous router and propagates the message downstream.. After receiving the TREE message, the AGTree-E at the egress node also updates the MRIB with the IP address of the previous router, triggers the multicast routing protocol for creating the multicast tree and sends a RESPONSE(OK) message to the requester ingress node, which then implements the third phase.

[0071] The third phase addresses to compute and create the ramified multicast aggregation trees that cover more than one egress node. AGTree computes the ramified trees by combining all un-ramified trees taking into consideration the ingress router and the available egress routers.. After that, AGTree discards all trees with branching point at the root and signals each one with a TREE message in order to configure the multicast trees.

[0072] Figure 5 shows the list of the ramified multicast aggregation trees, only at *I1* for simplification. To each computed ramified trees, AGTree allocates a multicast channel and sends a TREE message in order to configure the multicast tree.. The TREE message is sent towards each egress router of the distribution paths comprising the ramified tree. AGTree configures the multicast channel indicated by the received message in the outgoing interface towards the destination of the message (i.e., the egress router). When an agent realises that the indicated multicast channel is already configured in the retrieved interface, the operation is skipped. After receiving the TREE message, the egress router sends a RESPONSE(OK) message towards the ingress.

[0073] After all aggregation trees are created, the AGTree mechanism according to one embodiment controls the session connectivity in order to propagate the session-flows into the aggregation trees and such as to avoid waste of resources, e.g. due to the replication of packets on leaves of a tree with no active user members.

[0074] In the following the session connectivity control according to embodiments of the invention will be described in somewhat more detail..

[0075] The session connectivity control functionality is used by the AGTree mechanism as the support to ensure the session continuity along the end-to-end session path during its entire lifetime. The path may be heterogeneous in terms of connectivity schemes, being for instance, composed by unicast and multicast networks.

[0076] The end-to-end connectivity control over heterogeneous networks can be accomplished by using tunnel-based (see e.g. B. Zhang et al, "Host Multicast: A Framework for Delivering Multicast to End Users", Annual Joint Conference of the IEEE Computer and Communications Societies, June 2002) or translation-based (see e..g.. E.. Pearce *et al,* "System and Method for Enabling Multicast Telecommunications", *Application Patent,* US7079495 B1, July 2006) approaches. According to one embodiment, by default, AGTree uses the tunnel-based approach, where the aggregation node adds an extra header in the incoming packets of each session-flow, being filled with their original IP header five tuple. On the other hand, the IP header is updated with the five tuple information of the matched aggregation tree. At the deaggregator node (egress), AGTree updates the IP header with the five tuple carried in the extra header and removes it from the packet. Alternatively, AGTree may work combined with an external end-to-end connectivity control solutions (e.g., translation-based). This way, AGTree provides to the application the forwarding information of the selected aggregation channel.

[0077] Moreover, AGTree controls the switching of aggregation trees in order to avoid packet replication to undesired destinations. The state diagram shown in Figure 6 outlines the operations of AGTree to control the session connectivity on-demand in terms of switching of aggregation trees.. Whenever triggered with a session request (R*i*) carrying the session object (Sobj) of a session (Si) and the egress router IP (E), AGTree-E must locally retrieve a perfect matching aggregation tree (ATpm) for merging R*i*. However, when S*i* is locally activated on another egress router, AGTree retrieves another ATpm which supplies both egress routers (the existing one and the new one) and switches S*i* into the new AT*pm*. When a session has user members no longer activated at an access/egress router, such as due to a handoff or session leaving, AGTree must switch the session to another multicast tree in order to avoid waste of resources. Thus, an external mechanism able to implement this detection (such as described in E. Cerqueira et al. "Multi-user Session Control in the Next Generation Wireless System", 4th ACM International Workshop on mobility Management and Wireless Access, Malaga, Spain, Oct 2006), may report this situation to AGTree (L*i*), which then searches a new ATpm and switches S*i* accordingly..

[0078] One may say that according to one embodiment the switching of aggregation trees is performed such that for a given set of egress routers for a certain session there is chosen a "perfect match" or "best match" aggregation tree, and if the set of egress routers where there are receivers connected participating in the session changes (e.g. due to a new receiver joining or one receiver leaving) the switching is performed such that a new aggregation tree is chosen from the ones which have been generated during the initialisation phase. "Best match" or perfect match here according to one embodiment means that the selected aggregation tree has no empty leave nodes formed by an egress node not participating in the session, and further that there is a leave node in the selected aggregation tree for each participating in the session. In this manner the system can adapt to changing circumstances without a large signalling overhead.

[0079] Figure 7 illustrates the operations accomplished by AGTree in controlling on-demand session connectivity, in terms of switching aggregation trees, in a general scenario where the mobility support (e.g., Mobile IP) and session control (e.g., provided by the SEACON mechanism described in the patent application mentioned before) are considered. For simplification, the mobility and session control operations are not considered in the illustration. Consider the domain A as a network with PIM-SSM multicast support, where *N1* is a source server supplying session *S1*. The SEACON mechanism (mentioned already before) on *I1* triggers AGTree-E with the session request *R1* for *S1* due to the joint of *R1* in *E1*. AGTree merges *S1* into the aggregation tree 0, since it exclusively serves *E1*, and then reports the SEACON mechanism with the identification of the channel so that enable *R1* to receive the content of *S1*. tree 0 comprises I1-C1-E1. Next, SEACON triggers AGTree-E with session request R2 for *S1*, due to the joint of R2 in *E2*. This way, AGTree-E switches *S1* from aggregation tree 0 to aggregation tree 1 (which additionally comprises a branch at C1 and C1-C4-E2) for serving *E1* and *E2* simultaneously and reports SEACON with the identification of the new channel. In this case R1 may use IGMP or MLD to leave the old tree and join the new one. The same occurs due to the *S1* joint of R3 in E3, where the AGTree mechanism switches *S1* from aggregation tree 1 to aggregation tree 2 which now additionally comprises a branch at I1 and the path I1-C2-C3-E3. When R3 moves away from E3 to *E2* and no user members are activated on E3, the AGTree mechanism is triggered at *I1 e.g.* (by the SEACON mechanism) with a leave request from *E3*. After the attaching of R3 to E2, the AGTree mechanism is triggered with a request from R3 to join *S1* at E2. Since AGTree realises that *S1* is activated on *I1*, and supplies *E1* and E2, AGTree switches *S1* from aggregation tree 2 to aggregation tree 1.

[0080] In the case of a new handoff occurrence, AGTree works in the same way, after triggered at *I1*. Upon matched a new perfect aggregation tree, AGTree performs the switching of the old tree to the new one..

[0081] There have been described embodiments herein for illustrative purposes. According to one embodiment there is provided a configuration of a reservoir of local multicast trees during initialisation such as at bootstrap.. Then there may be performed a mapping of external multicast trees to the internal ones generated in advance, and there may be performed an adjustment of the mapping of external multicast trees based on the position of receivers such as to avoid leaky or incomplete matches between the receivers and the selected multicast tree..

[0082] One embodiment of the invention may be summarised as follows.

- a) Creation of aggregated multicast trees in advance (e.g. during bootstrap or initialization):

- a1) Discover of the available distribution paths..
- a2) Creation of un-branched trees.
- a3) Creation of branched distribution trees.
- a4) Choose the best paths by filtering:

- a4i)Discard paths with ingress routers as middle nodes.
- a4ii)Discard paths with branching points in ingress routers..

b) Mapping of multicast requests to aggregated trees:

- b1) Based on flow encapsulation..
- b2) Switches multicast trees to avoid delivering data

to leaf nodes with no users associated to them.

[0083] Operation a1) is schematically illustrated in Fig. 3, in operation a2) in Fig.. 4 and operation a4) in Fig. 5. The mapping is schematically illustrated in Fig. 7

[0084] It is apparent that modifications to this and other embodiments can be made by the skilled person, particularly by combining features of the foregoing description un such a way as is has not been explicitly disclose. It is, however, to be understood that the foregoing disclosure should also encompass such combinations of features for which it is clear to the skilled person that a feature which has been mentioned in connection with one embodiment may also be used in connection with another embodiment where it has not been explicitly mentioned. For example, in the above list of features only a4ii) without a4i) may be combined with the other features, even if such a combination is not explicitly disclosed it would be understood by the skilled person that such combinations are to be considered as at least implicitly disclosed.

[0085] It will be understood by the skilled person that the embodiments described hereinbefore may be implemented by hardware, by software, or by a combination of software and hardware. The modules and functions described in connection with embodiments of the invention may be as a whole or in part implemented by microprocessors or computers which are suitably programmed such as to act in accordance with the methods explained in connection with embodiments of the invention. An apparatus implementing an embodiment of the invention may e.g comprise a node or element (such as a router) in a network which is suitably programmed such that it is able to carry out a session-aware connectivity control as described in the embodiments of the invention..

[0086] According to an embodiment of the invention there is provided a computer program, either stored in a data carrier or in some other way embodied by some physical means such as a recording medium or a transmission link which when being executed on a computer enables the computer to operate in accordance with the embodiments of the invention described hereinbefore..

[0087] Embodiments of the invention may be implemented e..g.. by nodes in a network or any entities in a network which are programmed to operate in accordance with the multicast tree allocation mechanisms as described before.

Claims

1. A method for multicast tree allocation in a network, said method comprising:

   creating during an initialisation phase a plurality of possible multicast distribution trees, wherein said plurality of possible multicast distribution trees is created such that for each of a plurality of combinations of egress nodes and a given ingress node there has been created a corresponding distribution tree;
   switching of distribution trees by switching to another distribution tree selected from the distribution trees which have been generated during said initialisation phase, wherein said switching is performed in response to an egress node joining or leaving a session.

2. The method of claim 1, wherein

   said multicast trees comprise multicast aggregation trees in which multiple multicast groups are sharing one distribution tree.

3. The method of one of the preceding claims, further comprising:

   controlling the session connectivity by selecting a suitable aggregation tree such that the selected aggregation tree has no empty leave nodes formed by an egress node not participating in the session, and further such that there is a leave node in the selected aggregation tree for each node participating in the session.

4. The method of one of the preceding claims, wherein said generation of distribution trees further comprises the filtering of distribution paths, wherein said filtering comprises the selection of the best paths or the discarding of inefficient paths.

5. The method of claim 4, wherein said filtering comprises:

   discarding paths in which ingress routers are within the forwarding nodes, and/or
   discarding the paths with a branching point in the ingress routers.

6. The method of one of the preceding claims, said method comprising:

   creating during said initialisation phase all possible distribution trees such that if a session requires a certain distribution path from an ingress node or a source node to one or more egress nodes, the corresponding distribution path has already been created during said initialisation phase.

7. The method of claim 6, wherein each distribution tree which has been created is identified by a multicast tuple composed by the source IP address being the address of the ingress router which is the enter point of the session-flows and a destination IP address, being an IP multicast address.

8. The method of one of the preceding claims, further comprising:

discovering the available distribution paths within the network and creating the un-ramified aggregation trees.

9. . The method of claim 8, further comprising:

flooding the network during initialisation by the ingress nodes of said network with a TREE message, and upon receiving the TREE message, a router which has received said TREE message propagates the message on each outgoing interface except the one in which the message was received, wherein each message carries the local IP address, filled in the distribution path list, of the outgoing interface used for propagation, said method further comprising:
after receiving the TREE message, an egress node composes a RESPONSE message corresponding to the TREE message received and sends it to the ingress router which is source of the equivalent TREE message, said RESPONSE message carrying the distribution path list of the equivalent TREE message and the local outgoing interfaces.

10. The method of one of the preceding claims, said initialisation phase further comprising:

creating the un-ramified multicast aggregation trees, said creation comprising:
after having allocated a multicast channel to a distribution path, sending a TREE message along the distribution path, wherein each core node adds in the MRIB the IP address of the previous node and propagates the message downstream, said creation further comprising:
after receiving the TREE message, updating by the egress node the MRIB with the IP address of the previous router and triggering the multicast routing protocol for creating the multicast tree.

11. The method of one of the preceding claims, said initialisation phase further comprising:

computing and creating the ramified multicast aggregation trees that cover more than one egress node, wherein the ramified trees are computed by combining all un-ramified trees taking into consideration the ingress router and the available egress routers,
allocating a multicast channel to each of the ramified multicast aggregation trees.

12. The method of claim 11, further comprising:

discarding all trees with branching point at the root, and
sending a signalling message to all trees in order to configure the multicast trees..

13. The method of one of the preceding claims, further comprising:

controlling the session connectivity in order to propagate the one or more session-flows into the generated distribution trees such as to avoid a waste of resources.

14. An apparatus for multicast tree allocation in a network, said apparatus comprising:

a module for creating during an initialisation phase all possible distribution trees such that if a session requires a certain distribution path from an ingress node or a source node to one or more egress nodes, the corresponding distribution path has already been created during said initialisation phase.

15. The apparatus of claim 14, further comprising:

a module for performing a method according to one of claims 2 to 13.

16. A computer program comprising computer program code which when being executed on a computer enables said computer to carry out a method according to one of claims 1 to 13.

Amended claims in accordance with Rule 137(2) EPC.

1. A method for multicast tree allocation in a network, said method comprising:

creating during an initialisation phase a plurality of possible multicast distribution trees, wherein said plurality of possible multicast distribution trees is created such that for each of a plurality of combinations of egress nodes and a given ingress node of said network there has been created a corresponding distribution tree;

characterized by
switching of distribution trees by switching to another distribution tree selected from the distribution trees which have been generated during said initialisation phase, wherein said switching is performed in response to an egress node joining or leaving a session.

2. The method of claim 1, wherein

said multicast trees comprise multicast aggregation trees in which multiple multicast groups are sharing one distribution tree.

3. The method of claim 2, further comprising:

controlling the session connectivity of a session by selecting a suitable aggregation tree such that the selected aggregation tree has no empty leaf nodes formed by an egress node of said network not participating in the session, and further such that there is a leave node in the selected aggregation tree for each node participating in the session.

4. The method of one of the preceding claims, wherein said generation of distribution trees further comprises the filtering of distribution paths, wherein said filtering comprises the selection of the best paths or the discarding of inefficient paths.

5. The method of claim 4, wherein said filtering comprises:

discarding paths in which ingress routers of said network are within the forwarding nodes of a path, and/or
discarding the paths with a branching point in an ingress routers of said network.

6. The method of one of the preceding claims, said method comprising:

creating during said initialisation phase all possible distribution trees such that if a session requires a certain distribution path from an ingress node or a source node to one or more egress nodes of said network, the corresponding distribution path has already been created during said initialisation phase.

7. The method of claim 6, wherein each distribution tree which has been created is identified by a multicast tuple composed by the source IP address being the address of an ingress router of said network which is the point where session-flows enter the network and a destination IP address, being an IP multicast address.

8. The method of one of the preceding claims, further comprising:

discovering the available distribution paths within the network and creating un-ramified aggregation trees.

9. The method of claim 8, further comprising:

flooding the network during initialisation by the ingress nodes of said network with a TREE message, and upon receiving the TREE message, a router which has received said TREE message propagates the message on each outgoing interface except the one in which the message was received, wherein each message carries the local IP address, filled in the distribution path list, of the outgoing interface used for propagation, said method further comprising:

after receiving the TREE message, an egress node of said network composes a RESPONSE message corresponding to the TREE message received and sends it to the ingress router which is source of the equivalent TREE message, said RESPONSE message carrying the distribution path list of the equivalent TREE message and the local outgoing interfaces.

10. The method of one of the preceding claims, said initialisation phase further comprising:

creating the un-ramified multicast aggregation trees, said creation comprising:

after having allocated a multicast channel to a distribution path, sending a TREE message along the distribution path, wherein each core node adds in the multicast routing information base (MRIB) the IP address of the previous node and propagates the message downstream, said creation further comprising:

after receiving the TREE message, updating by an egress node of said network the MRIB with the IP address of the previous router and triggering the multicast routing protocol for creating the multicast tree.

11. The method of one of the preceding claims, said initialisation phase further comprising:

computing and creating ramified multicast aggregation trees that cover more than one egress node of said network, wherein the ramified trees are computed by combining all un-ramified trees taking into consideration the ingress router of said network and the available egress routers of said network,
allocating a multicast channel to each of the ramified multicast aggregation trees.

12. The method of claim 11, further comprising:

discarding all trees with branching point at the root of the tree, and
sending a signalling message to all trees in order to configure the multicast trees.

13. The method of one of the preceding claims, further comprising:

controlling the session connectivity of a session in order to propagate the one or more session-flows of a session into the generated distribution trees such as to avoid a waste of resources.

14. An apparatus for multicast tree allocation in a network, said apparatus comprising:

a module for creating during an initialisation phase all possible distribution trees such that if a session requires a certain distribution path from an ingress node of said network or a source node to one or more egress nodes of said network, the corresponding distribution path has already been created during said initialisation phase; **characterized by**
a module for switching of distribution trees by switching to another distribution tree selected from the distribution trees which have been generated during said initialisation phase wherein said switching is performed in response to an egress node of said network joining or leaving a session.

15. The apparatus of claim 14, further comprising:

a module for performing all the steps of a method according to one of claims 2 to 13.

16. A computer program comprising computer program code which when being executed on a computer enables said computer to carry out all the steps of a method according to one of claims 1 to 13.

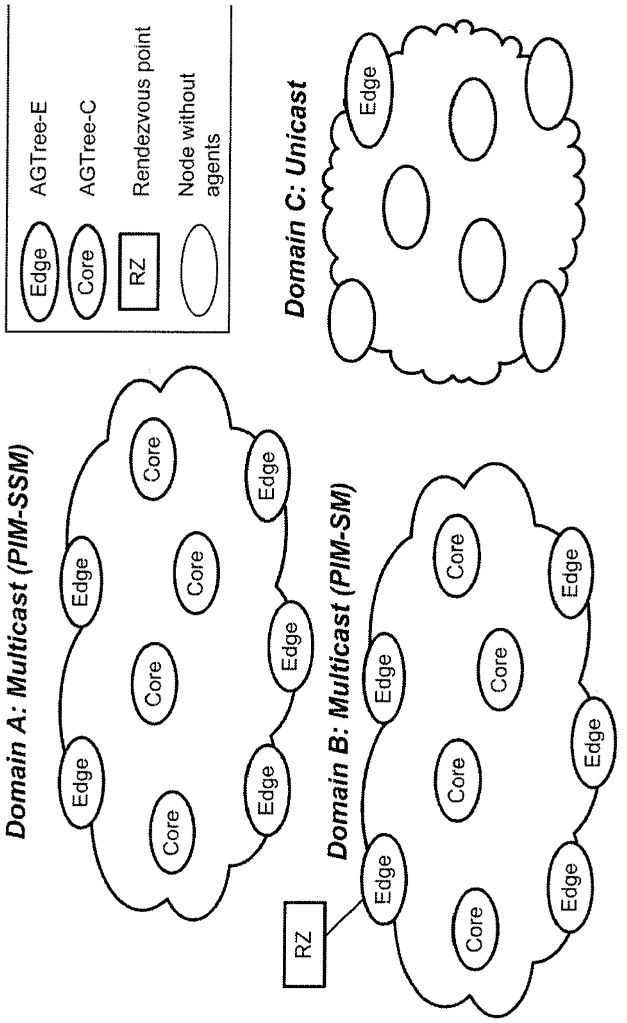

Domain A: Multicast (PIM-SSM)
Domain B: Multicast (PIM-SM)
Domain C: Unicast
Edge
Core
RZ
Edge AGTree-E
Core AGTree-C
RZ Rendezvous point
Node without agents

Fig. 1

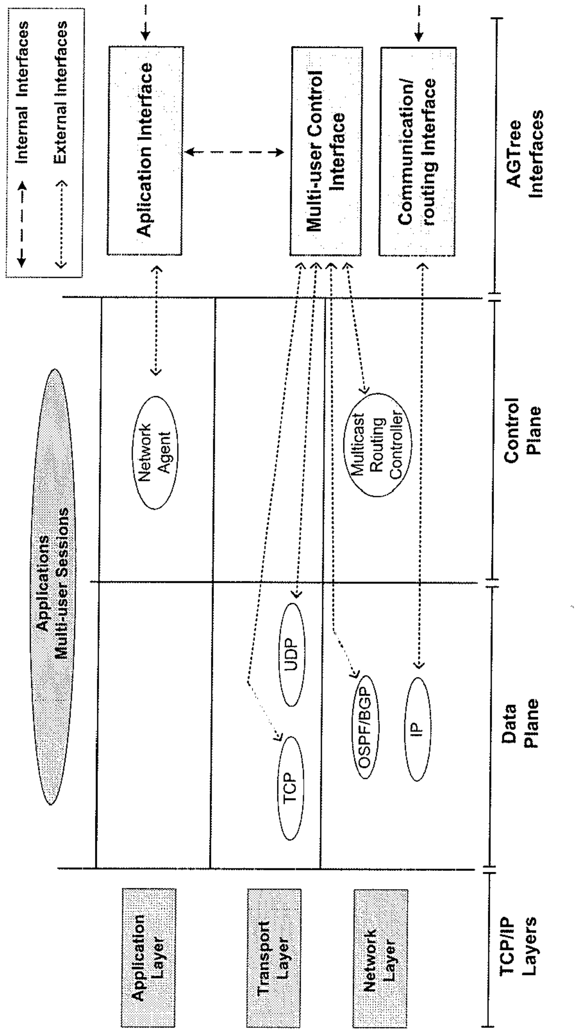
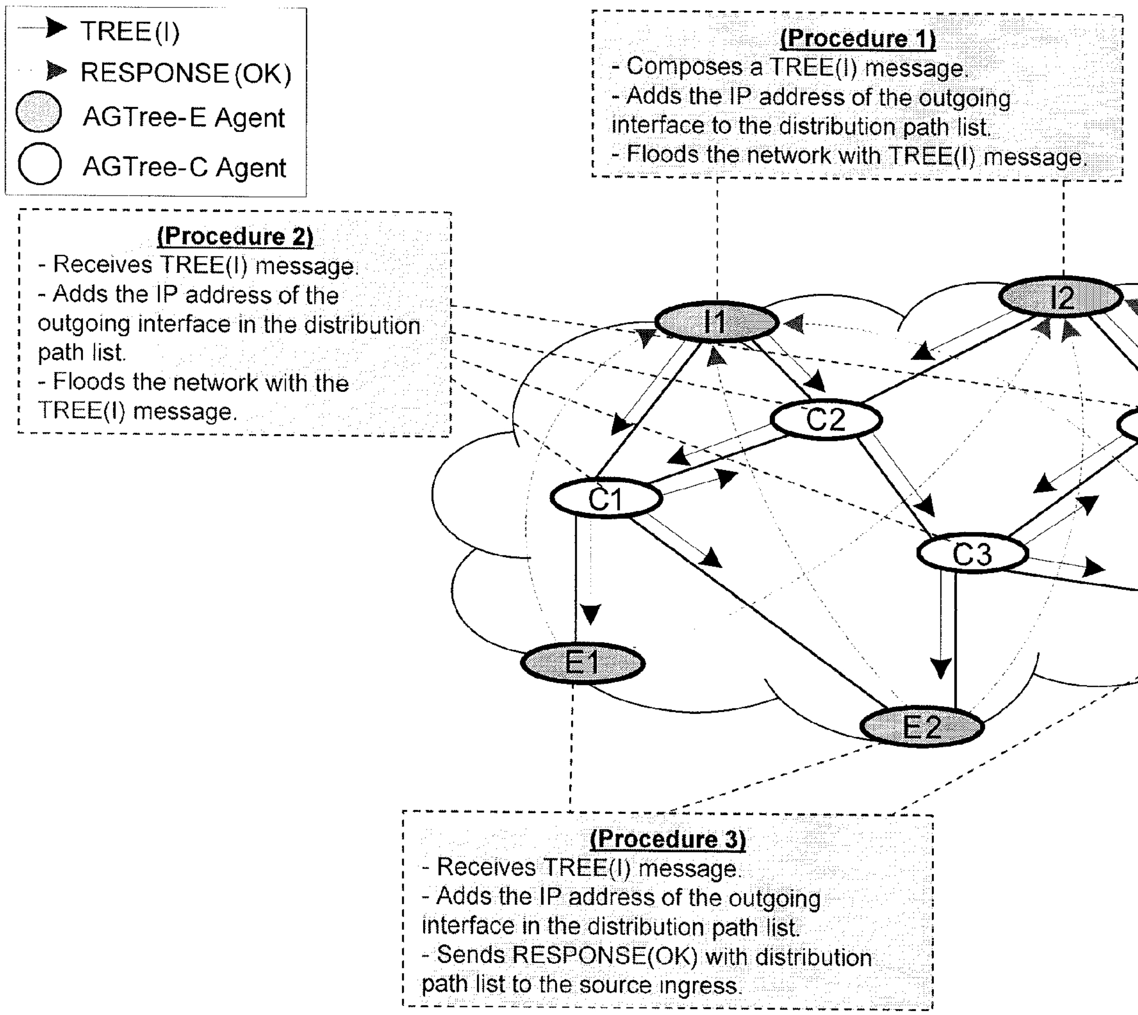

Internal Interfaces
External Interfaces
Aplication Interface
Multi-user Control Interface
Communication/ routing Interface
AGTree Interfaces
Applications Multi-user Sessions
Network Agent
Multicast Routing Controller
Control Plane
UDP
TCP
OSPF/BGP
IP
Data Plane
Application Layer
Transport Layer
Network Layer
TCP/IP Layers

Fig. 2



(Procedure 1)
- Composes a TREE(I) message.
- Adds the IP address of the outgoing interface to the distribution path list.
- Floods the network with TREE(I) message.

(Procedure 2)
- Receives TREE(I) message.
- Adds the IP address of the outgoing interface in the distribution path list.
- Floods the network with the TREE(I) message.

(Procedure 3)
- Receives TREE(I) message.
- Adds the IP address of the outgoing interface in the distribution path list.
- Sends RESPONSE(OK) with distribution path list to the source ingress.

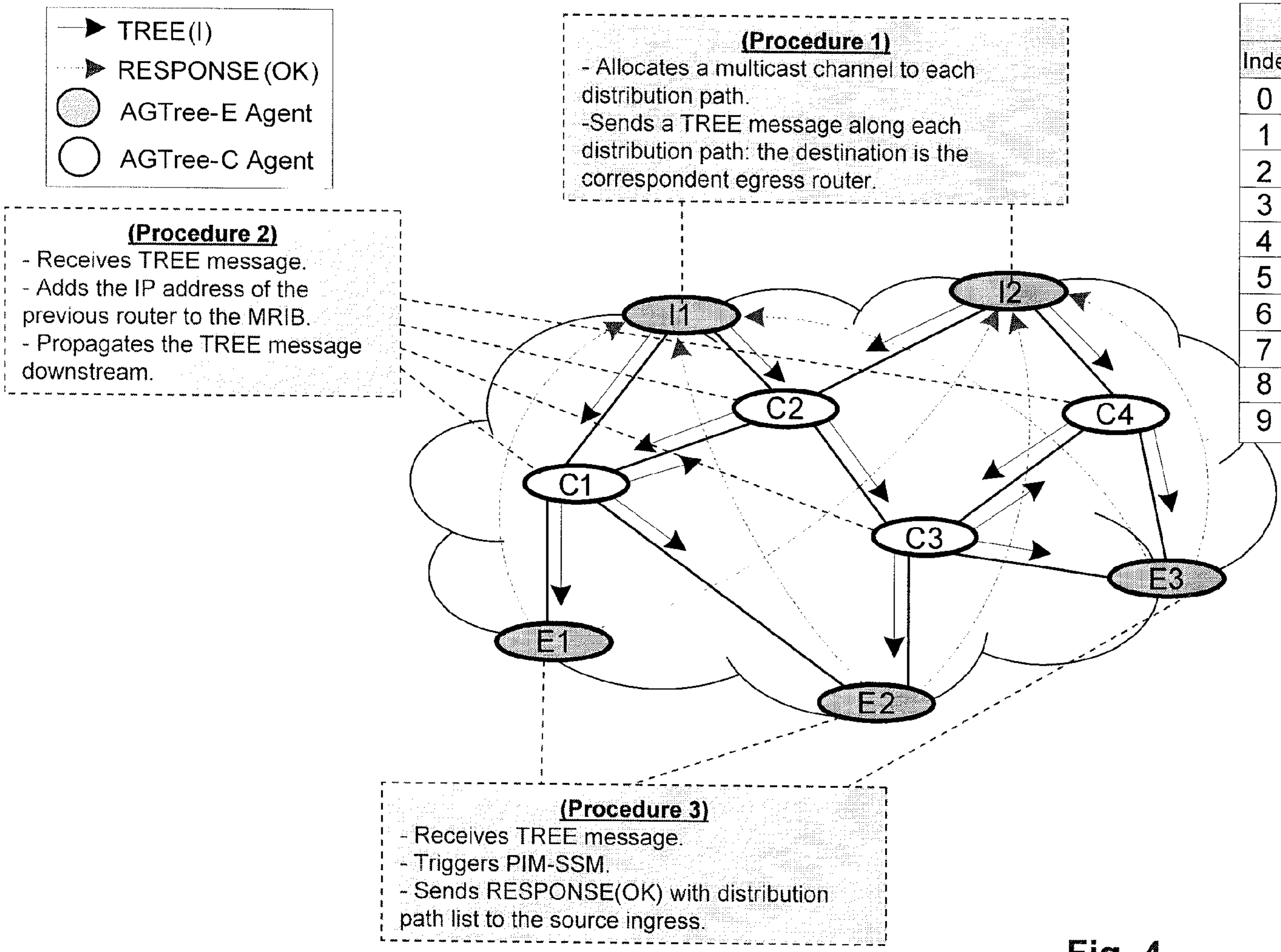
Paths from I1	
Index	Nodes
0	C1, E1
1	C1, E2
2	C2, C1, E1
3	C2, C1, E2
4	C2, C3, E2
5	C2, C3, E3
6	C2, C3, C4, E3
7	C1, C2, C3, E3
8	C1, C2, C3, E2
9	C1, C2, C3, C4, E3

Paths from I2	
Index	Nodes
0	C4, E3
1	C2, C1, E1
2	C2, C1, E2
3	C2, C3, E2
4	C2, C3, E3
5	C4, C3, E2
6	C4, C3, E3
7	C2, C3, C4, E3
8	C4, C3, C2, C1, E2
9	C4, C3, C2, C1, E1

Discovering Available Distribution Paths

Fig. 3

Creating Branched Trees



(Procedure 1)
- Allocates a multicast channel to each distribution path.
-Sends a TREE message along each distribution path: the destination is the correspondent egress router.

(Procedure 2)
- Receives TREE message.
- Adds the IP address of the previous router to the MRIB.
- Propagates the TREE message downstream.

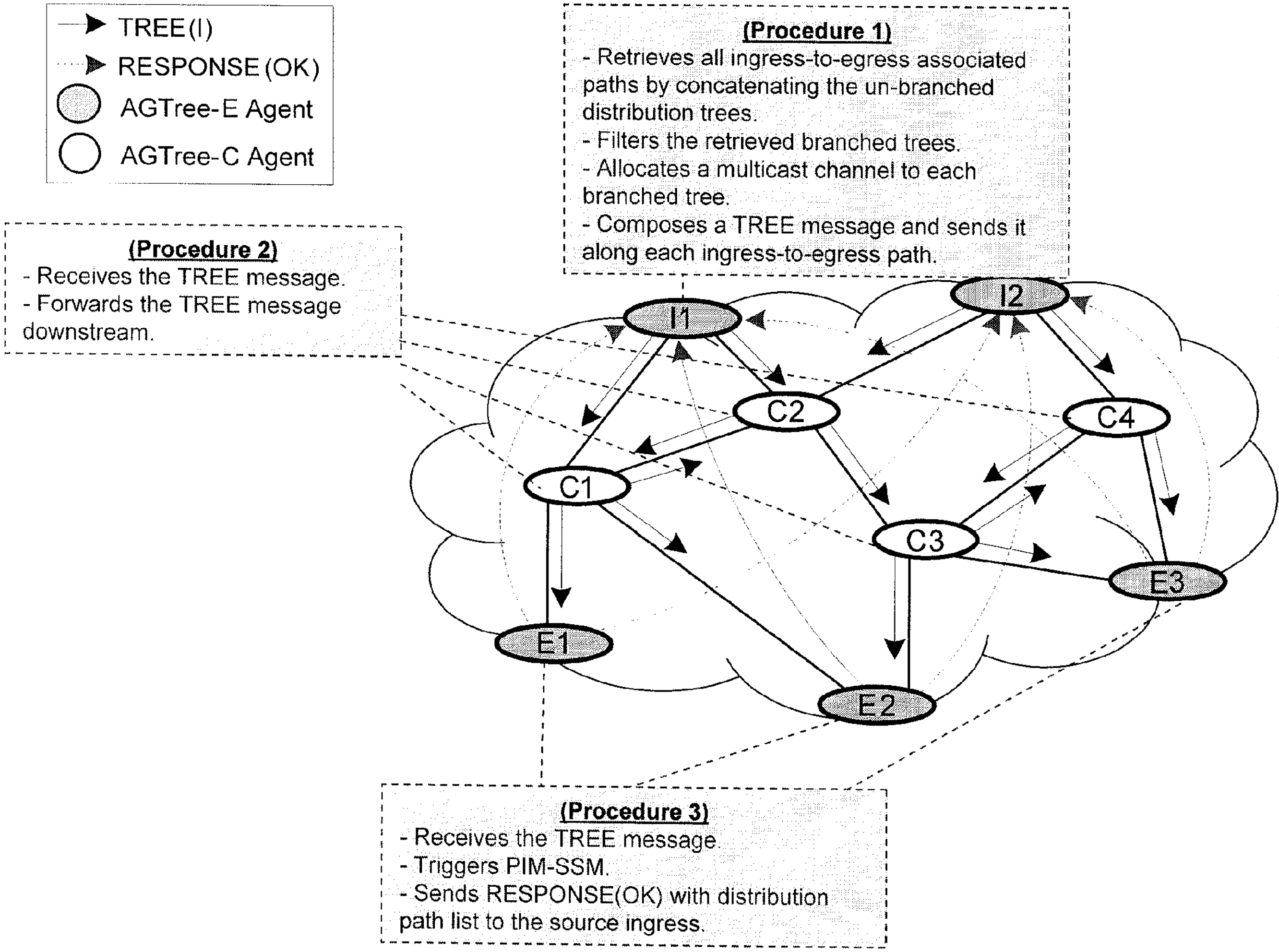
(Procedure 3)
- Receives TREE message.
- Triggers PIM-SSM.
- Sends RESPONSE(OK) with distribution path list to the source ingress.

Paths from I1

Index	Nodes	G
0	C1, E1	0
1	C1, E2	1
2	C2, C1, E1	2
3	C2, C1, E2	3
4	C2, C3, E2	4
5	C2, C3, E3	5
6	C2, C3, C4, E3	6
7	C1, C2, C3, E3	7
8	C1, C2, C3, E2	8
9	C1, C2, C3, C4, E3	9

Fig. 4

Creating Branched Trees



(Procedure 1)
- Retrieves all ingress-to-egress associated paths by concatenating the un-branched distribution trees.
- Filters the retrieved branched trees.
- Allocates a multicast channel to each branched tree.
- Composes a TREE message and sends it along each ingress-to-egress path.

(Procedure 2)
- Receives the TREE message.
- Forwards the TREE message downstream.

(Procedure 3)
- Receives the TREE message.
- Triggers PIM-SSM.
- Sends RESPONSE(OK) with distribution path list to the source ingress.

Paths from I 1		
Inde)	Nodes	G
0	C1, E1	0
1	C1, E2	1
2	C2, C1, E1	2
3	C2, C1, E2	3
4	C2, C3, E2	4
5	C2, C3, E3	5
6	C2, C3, C4, E3	6
7	C1, C2, C3, E3	7
8	C1, C2, C3, E2	8
9	C1, C2, C3, C4, E3	9

Branched trees from 1	
G	Paths
G10	8, 9
G11	6, 8
G12	4, 7
G13	3, 5
G14	2, 4
G15	1, 9
G16	1, 8
G17	1, 8, 9
G18	1, 7
G19	1, 6
G20	1, 6, 8
G21	1, 4
G22	1, 4, 7
G23	1, 2
G24	1, 2, 4
G25	0, 9
G26	0, 7
G27	0, 6
G28	0, 2
G29	0, 1
G30	0, 1, 9
G31	0, 1, 7
G32	0, 1, 6
G33	0, 1, 2

Fig. 5

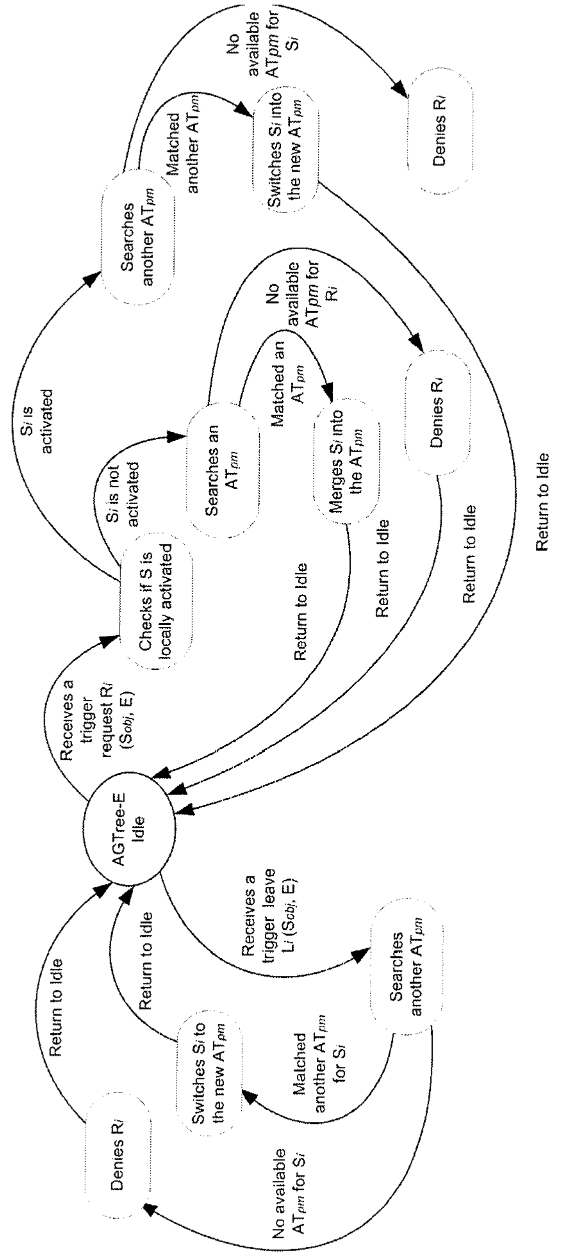

AGTree-E Idle
Receives a trigger request R_i (S_{obj}, E)
Checks if S is locally activated
S_i is activated
Searches another AT_{pm}
Matched another AT_{pm}
Switches S_i into the new AT_{pm}
No available $ATpm$ for S_i
Denies R_i
S_i is not activated
Searches an AT_{pm}
Matched an AT_{pm}
Merges S_i into the AT_{pm}
No available $ATpm$ for R_i
Denies R_i
Return to Idle
Return to Idle
Return to Idle
Return to Idle
Receives a trigger leave L_i (S_{obj}, E)
Searches another AT_{pm}
Matched another AT_{pm} for S_i
Switches S_i to the new AT_{pm}
No available AT_{pm} for S_i
Denies R_i
Return to Idle
Return to Idle

Fig. 6

Mapping into Multicast Aggregates

Implemented based on encapsulation:

- The aggregator node (ingress router) puts an extra header with information about the selected aggregation tree.
- The de-aggregator point (egress router) removes the extra header and forwarding the original packets.

(Procedure 1)

- Receives a trigger due to the join of R1 in S1.
- Allocates S1 to Tree 0.

(Procedure 2)

- Receives a trigger due to the join of R2 in S1.
- Switches S1 from Tree 0 to Tree 1

(Procedure 3)

- Receives a trigger due to the join of R3 in S1.
- Switches S1 from Tree 1 to Tree 2.

(Procedure 4)

- Receives a trigger due to the leave of R2 in S1.
- Switches S1 from Tree 2 to Tree 1.

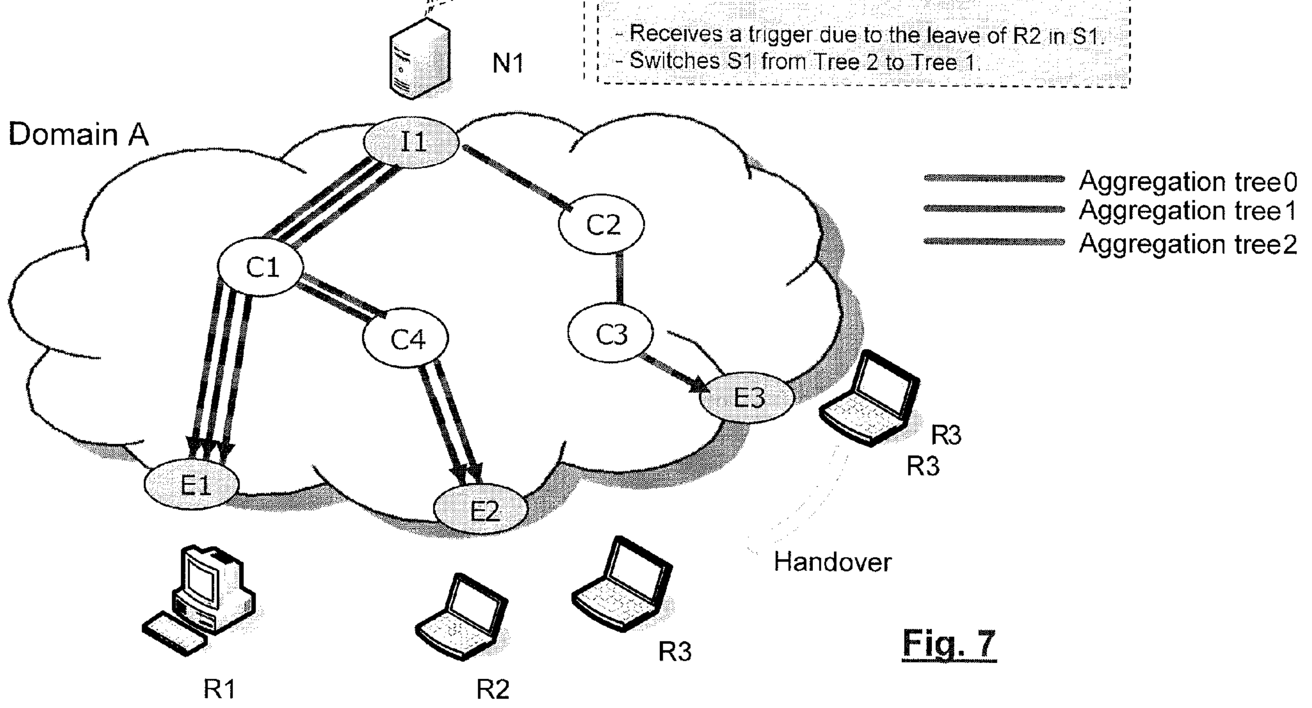


Fig. 7

European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 07 11 5344

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
X	LAO ET AL: "Tackling group-to-tree matching in large scale group communications" COMPUTER NETWORKS, ELSEVIER SCIENCE PUBLISHERS B.V., AMSTERDAM, NL, vol. 51, no. 11, 23 May 2007 (2007-05-23), pages 3069-3089, XP022094097 ISSN: 1389-1286	1-6,8, 11-16	INV. H04L12/18
Y	* page 3069, left-hand column, paragraph 1. - page 3071, left-hand column, paragraph 2.1.; figure 1 * * page 3072, left-hand column, paragraph 2.2.3. - page 3074, right-hand column, paragraph 3.2. * * page 3076, left-hand column, paragraph 4.1. - page 3077, right-hand column, paragraph 4.1; figure 2 * * page 3078, right-hand column, paragraph 4.5. * -----	7,9,10	
Y	SONG GUO ET AL: "A Distributed Minimum Energy Multicast Algorithm in MANETs" WORLD OF WIRELESS, MOBILE AND MULTIMEDIA NETWORKS, 2006. WOWMOM 2006. INTERNATIONAL SYMPOSIUM ON A BUFFALO-NIAGARA FALLS, NY, USA 26-29 JUNE 2006, PISCATAWAY, NJ, USA,IEEE, 26 June 2006 (2006-06-26), pages 134-142, XP010925858 ISBN: 0-7695-2593-8 * page 2, paragraph 3.2 * * page 3, paragraph 3.4 - paragraph 3.5 * -----	7,9,10	TECHNICAL FIELDS SEARCHED (IPC) H04L
A	EP 1 814 269 A (JUNIPER NETWORKS INC [US]) 1 August 2007 (2007-08-01) * page 9, column 16, paragraph 76 - page 11, column 20, paragraph 97; figure 11 * -----	1,9	

1

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
Munich	18 February 2008	Fischer, Erik

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons
....
& : member of the same patent family, corresponding document

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 07 11 5344

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report.
The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

18-02-2008

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 1814269 A	01-08-2007	JP 2007208953 A US 2007177594 A1	16-08-2007 02-08-2007

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- EP 07102594 A **[0049]**
- US 7079495 B1 **[0076]**

Non-patent literature cited in the description

- **T. BRAUN ; V. ARYA ; T. TURLETTI.** *A Backup Tree Algorithm for Multicast Overlay Networks,* May 2005 **[0007]**
- **M. KODIALEM ; T. LAKSHMAN.** Dynamic routing of bandwidth guaranteed multicasts with failure backup. *IEEE INFOCOM,* June 2006 **[0007]**
- **A. STRIEGEL ; G. MANIMARAN.** DSMCast: a scalable approach for DiffServ multicasting. *Computer Networks,* April 2004, vol. 44 **[0008]**
- **J. CUI et al.** Aggregated Multicast - A Comparative Study. *Cluster Computing,* January 2005, vol. 8 **[0009]**
- **E. KOHLER ; M. HANDLEY ; S. FLOYD.** Datagram Congestion Control Protocol (DCCP). *IETF, RFC4340,* March 2006 **[0049]**
- **S. BHATTACHARYYA.** An Overview of Source-Specific Multicast (SSM). *IETF, RFC3569,* July 2003 **[0051]**
- **D. ESTRIN et al.** Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification. *IETF, RFC2362,* June 1998 **[0051]**
- **J. MOY.** OSPF Version 2. *IETF, RFC2328,* April 1998 **[0052]**
- **B. ZHANG et al.** Host Multicast: A Framework for Delivering Multicast to End Users. *Annual Joint Conference of the IEEE Computer and Communications Societies,* June 2002 **[0076]**
- **E. CERQUEIRA et al.** Multi-user Session Control in the Next Generation Wireless System. *4th ACM International Workshop on mobility Management and Wireless Access,* October 2006 **[0077]**