Description

Background of the invention

The present invention relates to use of alias/logical names, in place of physical addresses, for establishing communications between entities in a data processing network (storage resources, human operators, etc.). More particularly, it relates to methods for enabling computing nodes in a network to operate as peers in respect to creating and using such names (i.e. communicate directly with each other), and thereby avoid reliance on a central or master communication node whose failure could prevent further name handling activities.

Existing distributed data processing networks have taken a centralized approach to support of alias name usage. This entails translations of messages directed to named entities into messages directed to network address locations at which respective entities are physically located.

This tends to increase the volume of communication traffic required to be handled by the network, as well as to increase the probability of error in any one communication dependent on such translation. Furthermore, if the translations from name-directed messages to location-directed messages are carried out by a "master" computing system at one node, failure of the latter system could disable the network.

Both European Patent Application EP—A—0074865 and IBM Technical Disclosure Bulletin, Vol. 23, No. 5, October 1980, Pages 1811—1812, disclose distributed data processing networks which do not take a centralized approach. However, in the first document, although there is provided an automatic generation of addresses for local processes, these addresses are not logical, as is the case in the invention disclosed and claimed below. In the second document, there is not provided any automatic generation of addresses: a storage array contains the addresses of those processes which are resident in the local processor.

The present invention seeks to avoid these difficulties by using a distributed approach to creation of logical names and the use of logical names for communication. It does not require a central directory, although it doe not preclude use of such.

Summary of the invention

This invention as claimed permits data processing systems linked to nodes of a communication network to create and use logical names on a distributive basis, and thereby to sustain data communications between resources known by various logical names, and distributed throughout the network, without dependence on a central or master directory.

The invention as claimed provides a method of exchanging controlling communications between nodes for establishing sessions between entities served by said nodes and known across said network only by logical names which are not related to their physical locations in the network, for a data communication network comprising multiple computing nodes linked by communication media and/or channels, comprising:

maintaining name association tables at said nodes indicating logical names associated with respectively served entities;

broadcasting a Call Name check communication over said network from any node seeking to establish a session between a pair of calling and called entities; said communication containing information including the physical address in said network of the node serving the calling entity and the names of the calling and called entities;

receiving said communication at other nodes in said network and searching respective name association tables for the name of the called entity;

transmitting an acknowledgment of said communication from any other node finding the name of said called entity in its table to the address of the node originating said communication;

examining acknowledgments received by said originating node in response to said communications; and conditional upon said acknowledgments, initiating an exchange of further communications between the Call Name check originating node and an acknowledging node for establishing a session relation between the calling and called entities.

These and other features, aspects and advantages of our invention will be more fully understood and appreciated by considering the following detailed description and claims.

Description of the drawings

Figure 1 is a schematic illustration of a network environment adaptable to the present invention;

Figure 2 illustrates exemplary frame formats for conducting communication between nodes in the environment network of Figure 1;

Figure 3 illustrates the form of a nodal name table in accordance with the present invention;

Figure 4 is an exemplary map of a portion of memory at a processing node in Figure 1, illustrating storage parameters of present name creation and session establishment processes and their relationships to other stored information;

Figure 5 illustrates the form of a Network Control Block (NCB) information array useful for name creation in accordance with the present invention;

Figure 6 illustrates the present process of name creation in a flow diagram form;

Figures 7a—7d illustrate various types of network communication frames used in the present name creation process;

Figure 8 illustrates an NCB for session establishment;

Figure 9 illustrates process sequences for session establishment;

Figure 10 illustrates a session table;

Figure 11 illustrates a Call Request frame for session establishment;

Figure 12 illustrates a Listen Acknowledgment frame for session establishment;

Figure 13 illustrates a Call Confirmation frame for session establishment;

Figure 14 illustrates a Listen Confirmation frame for session establishment;

Figure 15 illustrates an NCB for setting listen status;

Figure 16 shows the process flow sequence for setting listen status; and

Figure 17 shows a multi-network topology useful for explaining the need for "hop count" restrictions in systems allowing for passage of present name check and session establishment communications across networks.

Detailed description of the invention

1. Introduction

For obvious purposes, such as security, distributed data processing systems have been organized to require clearance of access to the system through a single checking facility, one usually located at an access node on a communication medium or channel shared by the processing nodes.

Typically, access to such systems has been controlled by a central facility via "Login" commands and passwords. Such systems have evolved to permit designation of authorized operators by password names and designation of resources by either addresses representing physical locations, or logical names having no direct relation to physical locations. Concommitantly, the central facility for access control has evolved to maintain a directory of names and resources available to each name, and to control access to such resources through network communications and references to the directory.

Control centers of this type are difficult to manage and have inherent limitations tending to adversely affect system performance and availability. They require supervisory software or the equivalent for keeping track of associations between names and physical entities and address locations, and if a center fails it incapacitates the whole network. Even systems adapted for automatic relocation of the central control function on failure are incapacitated for substantial intervals of time during relocation. Such control centers also generally require dedication of a discrete processing facility or unit to the central control function, which for obvious reasons is inefficient.

For these reasons we have devised the present distributed system for controlling name usage and allocation of resources. Our system does not preclude existence or usage of a central control, but it does not require such. In our system, resources and entities (processors, storage files, human operators, etc.) are associated by names with access nodes to network media through which they may communicate directly (i.e. without central translation of names to addresses).

In this environment, a most primitive form of which is suggested in Figure 1, collections of physically separated data handling resources and other entities 1—3 are communicatively linked via bus 4 into a distributed data processing network. Each collection interfaces to the bus at a respective access node 1a—3a. As typified at 1, the equipment at any access node may comprise a hierarchy of primary ("host") processing equipment 1b, secondary processing equipment 1c serving as a communication I/O controller, and transceiving elements 1d for interfacing directly to the medium. Equipment 1b, 1c may couple to not-shown peripheral devices (in addition to the network bus 4), and thereby provide for communication between these devices and the bus. Bus 4 may be either a continuous line (forming e.g. a carrier sense multiple access local area net), or several line segments forming a ring or star type network, or a free space radio link between microwave repeaters or between such repeaters and an orbiting earth satellite.

Depending on the processing resources available at a node, the processing functions required at that node for name and resource management may be variously apportioned between the primary and secondary processors. If the secondary processor is a specialized I/O controller, these functions could be concentrated entirely in the primary processor, whereas if the secondary processor includes a general purpose computer with sufficient capacity it could be programmed to handle many of these functions. A node with extensive secondary processing capacity would be more expensive to construct and maintain than a node having only a highly specialized I/O controller, but this expense would generally be offset by the performance advantages realizable through the freeing up of the primary processor for other processing functions.

In this environment, the processing facilities at the network nodes (1a, 2a, 3a, ...) exchange control information and data in discrete packets or frames having formats exemplified in Figure 2. Control frames (Figure 2a), consist of a fixed length header H, followed by a Cyclic Redundancy Check character CRC/H for checking the preceding information content. Data frames (Figure 2b), include a fixed length header portion H, followed by a variable length data portion D. The data portion concludes with a cyclic redundancy check character, CRC/D. The header position optionally concludes with an intermediate check character CRC/H. If the intermediate check character is provided, it and the CRC/D character are used for verifying respective portions H and D. If the intermediate character is not provided, the final character CRC/D is used for verifying all preceding information. Control frames are used for requesting controlling actions and acknowledging frame receptions. Data frames are used for transferring data information D.

The header (Figure 2c) generally includes a synchronizing flag byte F, origin information (O), denoting the address of the originating node,

destination information (D), denoting intended receiving node positions, information T representing the frame type, information S denoting sequence, and information L denoting the frame (byte) length.

Upon gaining access to medium 4 (Figure 1), a "ready to send" node sends a control or data frame out over that medium (the method for gaining access to the medium is not relevant to the present invention, and indeed it is contemplated that the present invention will operate through virtually any medium and/or access discipline). The destination byte D and type byte T in this frame's header may direct reception of its information by either one specific node or all nodes.

A frame directed to all nodes is termed a "broadcast" type frame. A node receiving a frame directed specifically to it acknowledges such reception; i.e. it returns a control frame, which contains sequence numbers associating it to the received frame and information indicating the status of its reception (error/no error). Except for certain frames associated with the present invention, broadcast frames are not acknowledged. If a frame sent to a specific destination is not acknowledged within a pre-arranged interval, the originating node assumes that its transmission (or at least the destination information therein) was not recognizable, and repeats the transmission (up to a preset "retry" limit).

The foregoing environment is representative but "primitive". As will be shown later, the concepts underlying the present invention may be applied in more sophisticated network environments wherein sub-networks associated with different channels and/or media are linked into a major network through bridges and/or gateway adapters linking the channels/media. In the foregoing representative environment, processing facilities at the nodes provide name management, session management and resource management functions—in a distributively controlled mode—which are the subject of the present invention and are described in detail below.

2. Name creation elements

In accordance with the present invention, processing facilities at the nodes maintain name tables in their memories having the form shown in Figure 3. Each table contains only names currently assigned to entities which interface to the network through the respective node.

Entries in these tables are used at respective nodes for responding to Name Check, session establishment and other communications received through the network. Name check communications are used to verify either that a named entity exists within the network or that an entity name chosen at a transmitting node is unique within all or a part of the network. Session establishment communications are used to establish sessions (i.e. logical connections) between calling and called entities identified by names. These communications are more fully described hereinafter. They permit nodal computing systems to share resources across the network by name; i.e. without requiring translations of such communications through a central directory, or requiring each system to have detailed information as to the physical structure of the network or the paths required for linking any pair of entities. Names assigned to entities for use in session establishment should be unique across the network.

Processing facilities at each subject node are organized for processing in accordance with the software hierarchy suggested in the memory map of Figure 4. Application or utility software 31 controls the primary processor (e.g. 1b, Figure 1) to process data 32 under the direction of supervisory software 33. Supervisor 33 also schedules I/O operations through the primary processor's I/O channels, which attach to both the secondary processor (e.g. 1c, Figure 1) and other I/O devices. Interface software 34 sustains information exchanges between the primary and secondary processors. Network Control Programs 35 direct communications between the secondary processor and the network medium or channel (e.g. bus 4, Figure 1). The name table referred to above is shown in a block form at 36 and a session table indicated at 37 is used to hold parameters of established session connections. Send buffers 38 hold information frames for outgoing transmission to the network medium, and receive buffers 39 temporarily store incoming frames directed to the respective node (there may be plural send buffers and plural receive buffers).

To create a name association for an entity, the primary processing facility at a node serving that entity assembles a Network Control Block (NCB) information array (representing a command), having the form shown in Figure 5 and schedules its interrogation by another processing entity. This other entity may be another software module running on either the primary or secondary processor, depending on the capabilities of the latter processor (if the secondary processor and primary processor have access to common memory facilities, the NCB transfer may be accomplished by a simple exchange of control signals). This NCB is used for name creation or session establishment.

The fields in the NCB array are defined in Figure 5. The command field CMD defines the operation to be performed (NCB's can be used for many operations). If the function to be performed is name creation, the CMD term will be the operation code (30H) for an Add Name command. If the function is session establishment, the CMD code will be 10H or 11H (see Figures 10 and 17, and corresponding text). The NCB command evolves one of the operations described below associated with name creation and session establishment.

The field RC is used to hold a "return code" for returning to the program which was running before the NCB was fetched. Fields NUM and LSN are provided for respectively storing information

relating to name and session establishment functions. When the NCB is used for name creation, the NUM field stores a number which represents the name in a shortened form. When the NCB is being used for session establishment, the LSN field stores a (local session) number which corresponds to the session and serves as a shortened notation for the identity of the session.

If the NCB is for name creation, the name to be created and information identifying its associated entity are stored in the NCB NAME field. If the NCB is for session establishment, the names of the entities to be connected are stored in the CALL NAME field.

The POST field contains the address of a program routine for posting completion status to the program which issued the NCB (the execution of the command associated with an NCB is an interruptable function. i.e. one which can be conducted "piecemeal". When the operation is concluded, either successfully or unsuccessfully, the post address is used to access a program for transferring the concluding status information to the supervisory system software which issued the NCB).

2.1 Name creation operations

To adopt a name, the creating processor entity operates according to the flow diagram of Figure 6 to:

a. Check the local name table for indication of conflict in respect to use of the name specified in the NCB (operation 61, Figure 6).

b. Terminate the command interpretation operation if such conflict is detected, and pass a corresponding status indication to the originating program (operation 62). This may occur via conventional interruption or whatever other method is being used in the system for passing status information between program (or other) entities.

c. If a local conflict indication is not detected, the node creates a "Name Check" packet (frame), having the form illustrated in Figure 7a, and transmits that frame to the network (operation 63, Figure 6). The indication that this frame has a broadcast destination (i.e. that it is to be received and interpreted at all nodes) may be provided either explicitly, by value B in the destination field, as shown in Figure 7a, or it may be implied in the value (01) contained in the "type" field of the frame.

d. In accordance with the present invention, nodes receiving the Name Check frame are required to check respective name tables for local use of the name specified in the frame (operation 64), and return an acknowledgment fraxe (operation 65) having the form illustrated in Figure 7b if and only if a match (local duplication) is found. In the present system, and other contemporary communication systems, broadcast frames are normally not acknowledged. However, the above acknowledgment of "match found" is treated presently as an "exception" to this "general rule".

e. The node originating the Name Check frame waits a predetermined timeout interval for acknowledgments (operation 66, Figure 6), and then decrements a retry count function RTC which is initially set to a predetermined value (operation 68, Figure 6). If an acknowledgment is received, a reply count function RPC initially set to 0 value, is incremented (operation 67, Figure 6). If the value of RTC after decrementing is not 0 (decision 69), the Name Check transmission is repeated (operation path 70). If no acknowledgments are received after a predetermined number of repetitions, "good completion" status is posted (operation 71) via the NCB POST above, and the name table is updated to include an entry indicating the specified name, the entity to which it associates, its physical location at the node (I/O channel port) and status information defined later.

f. If one and only one acknowledgment is received (RPC=1), a final "name in use" status is posted (operation 72), indicating name duplication and requiring selection of another name.

g. If more than one acknowledgment is received ("no" path at decision 73), "name in use" status is posted and the node creates and broadcasts a "Duplicate Use" frame (operation 74) having the form shown in Figure 7c. This frame does not require acknowledgment. Nodes receiving it check their tables for the specified name (operation 75) and if a match is found, the table entry is marked as in conflict (such entries are not used until the conflict is resolved).

Upon completion of any of the foregoing operations, a final status indication is posted, as explained previously, indicating the state of the operation.

When a name has been assigned to an entity at a node via the foregoing procedure, the association between the name and entity may be communicated to other nodes by means not presently relevant (e.g., telephone calls between operators) so that the other nodes may share access to the entity via sessions (to be described).

3. Session establishment

In the present system, point-to-point communications between pairs of entities served by processing facilities at either a single node or two different nodes are conducted via "sessions". A session is a logical connection between a pair of entities (e.g. an operator at a keyboard served by one node and a storage file maintained at another node). A session may span multiple network frames. Sessions are initiated when entities such as application programs or keyboard operators require reference to other entities such as storage files.

Upon receipt of a suitable interruption request from a "calling entity" (an entity requiring reference to another entity, herein termed "called entity"), operating system software serving the calling entity sets up an NCB for creating a session with the called entity. If the calling entity is an application program, it may be free to continue other useful work provided that its continued operation is not dependent on the reference to the called entity.

This session initiating NCB (Figure 10), specifying the calling and called entities by name, is passed to another processing entity for interpretation (either to another virtual processor or software module in the primary processor or to the secondary processor). The NCB contains a Call command evoking the operations illustrated in the two parts of Figure 9, Figures 9A and 9B.

The interpreting processor first checks its local Name Table (operation 200, Figure 9A). If both named entities are attached locally, and the called entity is in "listening status" as defined later, a local (or intra nodal) session is established (operation 202, Figure 9A) by entry of information in the Session Table (Figure 10). This entry includes the names of the calling and called entities, their physical locations and identities, a bit flagging the session as local, and a session number (a pair of entities may have more than one session in process concurrently, and this number allows the local system to associate communications between the pair with the correct session). Thereafter, communications in either direction between these entities are initiated by name (or name number), passed to the processor responsible for maintenance of the Session Table and routed by that processor to the appropriate application and entity via reference to the session number.

If the called entity is not local, the processor originating the NCB causes a Call Name Check frame (Figure 7d) to be broadcast to the network (Operation 204, Figure 9A) for checking the existence of the name of the called entity, and examines acknowledgments (Figure 7b). Call Name Check differs from Name Check (Figure 7a) in the type field value (08 for Call Name Check; 01 for Name Check) and in the "context" of the name field (which represents a "called" entity name rather than a name proposed for adoption). This permits the optional Name Management facility of paragraph 2.4 above to ignore/allow unrestricted Call Name Check communications. This is true also for the acknowledgments.

This broadcast is repeated a predetermined number of times, via retry path 206, Figure 9A, to ensure accurate transfer of the request and interpretation of responses. If no acknowledgment or more than one acknowledgment is received, the NCB processor sets a Call Name Not Found status indication (208, Figure 9A) which terminates the operation on an unsuccessful basis. Also, if more than one acknowledgment returns (for a single repetition of the Name Check frame), the NCB processor creates and broadcasts a Duplicate Use frame (210, Figure 9B) as described previously for name creation. If only one acknowledgment is returned, the calling node broadcasts a Call Request packet (212, Figure 9B), having the form shown in Figure 11, which is acknowledged by a Listen Acknowledgment packet (214, Figure 9B) having the form shown in Figure 12, if correctly broadcast and received, and if the called entity is in Listening status as defined below.

The Call Request and Listen Acknowledgment frames contain session numbers, and establishment of the session is conditional upon these numbers matching. If these numbers match, the calling node sends a specifically addressed Call Confirmation frame (216, Figure 9B) having the form shown in Figure 13 to the node serving the called entity. The latter frame, if properly received, is acknowledged by a "Listen Confirmation" frame (218, Figure 9B) with form of Figure 14. Then, the processors serving the calling and called nodes make appropriate entries in their session tables establishing the session (identifying the paired entities, their session number and their physical locations in the network). Thereafter, communications in either direction between these entities are initiated by name and translated (via the Session Table) into point-to-point frames indicating the session number and unit identities.

Considering now the "listening" state mentioned above, supervisory software modules at the nodes use "LISTEN" NCB's of the form shown in Figure 15 to establish such states. These NCB's, containing Listen commands directed to named "caller" entities, are passed to respective processing entities which refer to respective Name Tables and pass back to the NCB source processor the physical identity and operating status of the caller entity. If that entity is available for Session Establishment, relative to the listenee/called entity named in the NCB, the NCB source processor signals the NCB evaluating processor to mark listening status in the Name Table, to associate that status with the listenee entity and relate it to the caller entity.

Consequently, if a node receives a Call Request and has suitable listen status pre-marked for the called entity, it will return a Listen Acknowledgment. This procedure thereby allows each node to govern its session traffic.

It should be appreciated that the foregoing procedures for Name Creation and Session Establishment allow a resource at any node to be shared across the network by name. These processes operate discretely separate from application processes running at the nodes, and permit the application processes and associated operating system software to call for a resource by name without knowing the physical location of that resource in the network.

4. Extension of name communications through nodal bridges and gateways

The foregoing process for Name Creation can be extended across networks operating through different media or on different channels of a comman medium. Such networks may be linked physically through nodal adapters providing bridging or gateway access as explained previously. The extensions are accommodated as follows:

4.1 Extended name creation

a. Consider the simple configuration of Figure 17, in which three networks—A, B and C—are linked for intercommunication via gateways 300,

302 and 304; gateway 300 linking node 300a in A with node 300b in B, gateway 302 linking node 302a in A with node 302c in C, and gateway 304 linking node 304b in B with node 304c in C.

b. Subject to hop count restrictions defined later, assume that Name Check and Call Name Check requests originating at a node in one network are forwarded unconditionally to the other networks through these gateways; whereby requests in A are forwarded to B and C via 300 and 302, requests in B are forwarded to A and C via 300 and 304, and requests in C are forwarded to A and B via 302 and 304. Assume also that acknowledgments and other communications associated with these requests (e.g. Duplicate Use frames) are similarly forwardable.

c. It should be apparent, from earlier discussions, that these requests would be transmitted with the intent to establish name/entity associations which would potentially extend across media or channel boundaries.

d. Note that without topological restrictions, such requests could circulate endlessly. For instance, a communication originating in A could recirculate to A; either through 300, B, 304, C and 302; or through 302, C, 304, B and 300.

e. To prevent this undesirable/unproductive multiplication of network traffic, it is required presently that any node originating a broadcast communication pertaining to Name Creation or Session Establishment provide a "hop count" number with that communication to restrict its transfer across network boundaries. This number (see Figures 7 and 11—14) indicates the number of network boundaries which the communication may cross.

f. Such communications may be transferred across boundaries either immediately, or held at the gateway and forwarded only if fewer than two acknowledgments are returned within the forwarding network (the latter procedure is slightly more complex and costly, but would reduce internetwork traffic by eliminating unnecessary forwarding communications after conditions indicative of Duplicate Name Use have been established).

g. In the foregoing example, the count would be set at 2, and decremented by 1 as the message containing it crossed any gateway. Thus, a Name Check originated in network A could transfer to B and C, through 300a and 302a respectively, and its hop count would be decremented to a value of 1 at each of these junctures. Corresponding messages could be cross-forwarded between B and C, through 304, with second decrements to their residual hop counts setting their values to 0. The 0 hop count value would prevent further forwarding so that the messages crossing between B and C could not re-enter A. This hop count function could be determined either by the nodes individually or by having a common node issue messages to all other nodes setting hop count restrictions for their name associated communications. The specific method is not considered relevant to the present invention. However, it should be noted that this determination must be based at least on knowledge of the linkage topology between the network media and/or channels.

h. An alternative to the use of the above hop count parameter would be to have each forwarding node insert an identifying tag in its forwarded messages. Then, if a gateway serving a node sees the tag of that node in an incoming message directed toward that node, it would not re-forward the message.

An additional requirement in the forgoing message forwarding system is that the timeout functions employed at the nodes for timing responses to Name Check and session creation communications must be tailored to the network system topology; i.e. such timeouts must be functionally related to the maximum communication delay in the system. If linked networks in a system have different delays, these timeouts may be related to the sum of the delays through all networks.

5. Conclusions

We have therefore described a method for enabling processors at individual nodes within a distributed processing network to create name associations for entities at respective nodes without having to communicate specifically with a central or controlling node, and a method for enabling such processors to use such names for establishing sessions between entities, whereby the processors and application programs running on the processors can initiate sessions without specific knowledge of the network topology or the physical locations of the entities in the network.

Claims

1. A method for establishing a communication session between a pair of entities having associated logical names, using only the associated logical names and not information regarding the physical locations of the entities, in a data communication network comprising multiple computing nodes, each node serving one or more entities which may be shared by other nodes, each node having a physical address in said network and being capable of having one or more logical names associated with either the node or its respectively served entities, comprising:

maintaining name association tables at said nodes for storing indications of logical names associated with respectively served entities;

at a node seeking to establish a session between a respectively served first entity having a first logical name and a second entity having a second logical name, comparing said second logical name to the logical names stored in the name association table maintained at the respective node;

establishing a session between said first and second entities, by means of processing operations performed entirely within the respective node—i.e. without any communications over said

302 and 304; gateway 300 linking node 300a in A with node 300b in B, gateway 302 linking node 302a in A with node 302c in C, and gateway 304 linking node 304b in B with node 304c in C.

b. Subject to hop count restrictions defined later, assume that Name Check and Call Name Check requests originating at a node in one network are forwarded unconditionally to the other networks through these gateways; whereby requests in A are forwarded to B and C via 300 and 302, requests in B are forwarded to A and C via 300 and 304, and requests in C are forwarded to A and B via 302 and 304. Assume also that acknowledgments and other communications associated with these requests (e.g. Duplicate Use frames) are similarly forwardable.

c. It should be apparent, from earlier discussions, that these requests would be transmitted with the intent to establish name/entity associations which would potentially extend across media or channel boundaries.

d. Note that without topological restrictions, such requests could circulate endlessly. For instance, a communication originating in A could recirculate to A; either through 300, B, 304, C and 302; or through 302, C, 304, B and 300.

e. To prevent this undesirable/unproductive multiplication of network traffic, it is required presently that any node originating a broadcast communication pertaining to Name Creation or Session Establishment provide a "hop count" number with that communication to restrict its transfer across network boundaries. This number (see Figures 7 and 11—14) indicates the number of network boundaries which the communication may cross.

f. Such communications may be transferred across boundaries either immediately, or held at the gateway and forwarded only if fewer than two acknowledgments are returned within the forwarding network (the latter procedure is slightly more complex and costly, but would reduce internetwork traffic by eliminating unnecessary forwarding communications after conditions indicative of Duplicate Name Use have been established).

g. In the foregoing example, the count would be set at 2, and decremented by 1 as the message containing it crossed any gateway. Thus, a Name Check originated in network A could transfer to B and C, through 300a and 302a respectively, and its hop count would be decremented to a value of 1 at each of these junctures. Corresponding messages could be cross-forwarded between B and C, through 304, with second decrements to their residual hop counts setting their values to 0. The 0 hop count value would prevent further forwarding so that the messages crossing between B and C could not re-enter A. This hop count function could be determined either by the nodes individually or by having a common node issue messages to all other nodes setting hop count restrictions for their name associated communications. The specific method is not considered relevant to the present invention. However, it should be noted that this determination must be based at least on knowledge of the linkage topology between the network media and/or channels.

h. An alternative to the use of the above hop count parameter would be to have each forwarding node insert an identifying tag in its forwarded messages. Then, if a gateway serving a node sees the tag of that node in an incoming message directed toward that node, it would not re-forward the message.

An additional requirement in the forgoing message forwarding system is that the timeout functions employed at the nodes for timing responses to Name Check and session creation communications must be tailored to the network system topology; i.e. such timeouts must be functionally related to the maximum communication delay in the system. If linked networks in a system have different delays, these timeouts may be related to the sum of the delays through all networks.

5. Conclusions

We have therefore described a method for enabling processors at individual nodes within a distributed processing network to create name associations for entities at respective nodes without having to communicate specifically with a central or controlling node, and a method for enabling such processors to use such names for establishing sessions between entities, whereby the processors and application programs running on the processors can initiate sessions without specific knowledge of the network topology or the physical locations of the entities in the network.

Claims

1. A method for establishing a communication session between a pair of entities having associated logical names, using only the associated logical names and not information regarding the physical locations of the entities, in a data communication network comprising multiple computing nodes, each node serving one or more entities which may be shared by other nodes, each node having a physical address in said network and being capable of having one or more logical names associated with either the node or its respectively served entities, comprising:

maintaining name association tables at said nodes for storing indications of logical names associated with respectively served entities;

at a node seeking to establish a session between a respectively served first entity having a first logical name and a second entity having a second logical name, comparing said second logical name to the logical names stored in the name association table maintained at the respective node;

establishing a session between said first and second entities, by means of processing operations performed entirely within the respective node—i.e. without any communications over said

network—if the second logical name matches a logical name stored in the respective name association table; and if the second logical name does not match any logical name in the respective name association table, successively broadcasting a Call Name check message over said network from the node seeking to establish said session, said messsge containing said second logical name and the address in said network of the node originating the broadcast;

timing out a predetermined time interval at said originating node immediately following said broadcast;

receiving said Call Name check message at other nodes in said network;

at each said other node, comparing the second logical name in said received message with logical names stored in the respective name association table and transmitting an acknowledgment message directed by address specifically to the node which broadcasts the Name Check message if and only if a match is detected between said second logical name and a logical name in the respective name association table;

at the node which originated the broadcasted Call Name check message, monitoring acknowledgments of said broadcast, received during said timeout interval; and

conditional upon said received acknowledgments, either initiating further communications with an acknowledging other node, for continuing the process of seeking to establish said session, or aborting said process.

2. The method for establishing a communication session of claim 1 wherein said further communications include:

sending a Call Request communication from said node seeking to continue said process to said second entity; and

returning an acknowledgment in response to said Call Request from the node serving said second entity to the node which sent said Call Request.

3. The method of claim 2 wherein said nodes are adapted for establishing Listening State associations between respectively served entities and other entities, for regulating communication traffic between entities, and said second acknowledgment is given only on condition that a Listening State association pre-exists between said second logical name and said first logical name at a node serving an entity associated with said second logical name and receiving a said Call Request.

4. The method of claim 1 including:

continuing said process if only one acknowledgment is received in response to said broadcasted Call Name check message and aborting said process if either no acknowledgment or more than one acknowledgment is received in response to said broadcasted Call Name check message; and

in response to receipt of more than the one acknowledgment of said broadcasted Call Name check message, broadcasting a Duplicate Use message over said network from said node which originated said broadcasted Call Name check message, said Duplicate Use message indicating existence of conflicting logical name usage and having the effect of informing the computing systems at the nodes which acknowledged the broadcasted Call Name check message that their uses of said second logical name are conflicting.

Patentansprüche

1. Verfahren für die Erstellung einer Kommunikationssitzung zwischen einem Paar Einheiten mit assoziierten logischen Namen, unter Verwendung allein der assoziierten logischen Namen, ohne Information über die physischen Stellen der Einheiten, in einem Datenkommunikationsnetz, das mehrfache Rechnungsknoten enthält, wobei jeder Knoten eine oder mehrere Einheiten versorgt, die von anderen Knoten geteilt werden können, wobei jeder Knoten im besagten Netzwerk eine physische Adresse besitzt und fähig ist, einen oder mehrere logische Namen zu haben, die entweder dem Knoten oder seinen jeweils versorgten Einheiten entsprechen, enthaltend:

Tabellen für die Führung der Namensassoziationen an besagten Knoten für die Speicherung von Angaben logischer Namen in Verbindung mit den jeweils versorgten Einheiten;

an einem Knoten, der versucht, zwischen einer jeweils zuerst versorgten Einheit mit einem ersten logischen Namen und einer zweiten Einheit mit einem zweiten logischen Namen eine Sitzung aufzubauen, wobei der besagte zweite logische Name mit den logischen Namen verglichen wird, die in der Namensassoziationstabelle abgelegt sind, welche am jeweiligen Knoten geführt wird;

Erstellung einer Sitzung zwischen den besagten ersten und zweiten Einheiten durch Verarbeitungsvorgänge, die ganz innerhalb des jeweiligen Knotens abgewickelt werden, d.h. ohne irgendwelche Kommunikationen über das besagte netz, wenn der zweite logische Namen einem in der zugehörigen Namensassoziationstabelle abgelegten Namen entspricht; und wenn der zweite logische Name mit keinem logischen Namen in der zugehörigen Namensassoziationstabelle entspricht, nacheinander

Aussenden einer Namensaufruf-Kontrollmeldung per Funk über das besagte Netz vom Knoten, der versucht, die besagte Sitzung aufzubauen, wobei die besagte Meldung den besagten zweiten logischen Namen und die Adresse im besagten Netz des Knotens enthält, von dem aus der Funkruf ausgegangen ist;

Erstellung eines vorgegebenen Zeitintervalls am besagten Ausgangsknoten unmittelbar nach Ausgabe der Funkmeldung;

Empfang der besagten Namenskontrollmeldung an anderen Knoten im besagten Netz;

bei jedem besagten anderen Knoten Vergleich des zweiten logischen Namens in der besagten empfangenen Meldung mit logischen Namen, die in der zugehörigen Namensassoziierungstabelle gespeichert sind, und Übertragung einer Bestäti-

gungsmeldung, die durch die Adresse spezifisch an den Knoten gerichtet wird, der die Namenskontrollmeldung per Funk aussendet, wenn und nur wenn eine Übereinstimmung zwischen dem besagten zweiten logischen Namen und einem logischen Namen in der zugehörigen Namensassoziierungstabelle entdeckt wird;

an dem Knoten, von dem aus die Namensaufrufmeldung gefunkt wurde, Überwachung der Bestätigungen der besagten Funkmeldung, die während des besagten Zeitintervalles eingehen; und

je nach den besagten empfangenen Bestätigungen entweder Einleitung weiterer Kommunikationen mit Bestätigung eines anderen Knotens für die Fortsetzung des Vorgangs, in dem versucht wird, die besagte Sitzung aufzubauen, oder Abbruch des besagten Vorgangs.

2. Verfahren für den Aufbau einer Kommunikationssitzung gemäß Anspruch 1, in der die besagten Kommunikationen ferner enthalten:

Aussendung einer Rufanforderungsmeldung vom besagten Sucherknoten, um den Vorgang auf die besagte zweite Einheit auszudehnen; und

Rückgabe einer Bestätigung als Antwort auf den besagten Anforderungsruf von dem Knoten, der die besagte zweite Einheit versorgt, an den Knoten, der die besagte Rufaufforderung ausgegeben hat.

3. Verfahren gemäß Anspruch 2, in dem die besagten Knoten angepaßt werden, um Hörzustandsassoziationen zwischen den jeweils versorgten Einheiten und anderen Einheiten zu erstellen, für die Regulierung des Kommunikationsverkehrs zwischen Einheiten, und die besagte zweite Bestätigung nur erteilt wird, wenn eine Hörzustandsassoziation zwischen dem besagten zweiten logischen Namen und dem besagten ersten logischen Namen an einem Knoten vorhanden ist, der eine Einheit versorgt, die mit dem besagten zweiten logischen Namen assoziiert ist und eine besagte Rufanforderung empfängt.

4. Verfahren gemäß Anspruch 1, enthaltend:

Fortsetzung des besagten Vorgangs, wenn nur eine Bestätigung als Antwort auf die gefunkte Namensaufrufkontrollmeldung empfangen wird, und Abbruch des besagten Vorgangs, wenn entweder keine Bestätigung oder mehr als eine Bestätigung als Antwort auf die gefunkte Namensaufrufkontrollmeldung empfangen wird, und

in Anwort auf den Empfang von mehr als einer Bestätigung der besagten gefunkten Namensaufrufkontrollmeldung Funken einer Doppelnutzungsmeldung über das besagte Netz vom besagten Knoten aus, von dem die besagte gefunkte Namensaufrufkontrollmeldung ausgegangen ist, wobei die besagte Doppelmeldung das Vorhandensein eines konflikthaften Gebrauchs logischer Namen anzeigt und die Rechensysteme an den Knoten, die die Namensaufrufkontrollmeldungen bestätigt haben, darauf hinweist, daß ihre Benutzung des besagten zweiten logischen Namens zu Konflikten führt.

Revendications

1. Procédé pour établir une session de communication entre une paire d'entités ayant des noms logiques associés, en utilisant seulement les noms logiques associés et non des informations concernant les emplacements physiques de ces entités, dans un réseau de communicaton de données comportant des noeuds de calcul multiples dont chacun dessert une ou plusieurs entités qui peuvent être partagées avec d'autres entités, chaque noeud ayant une adresse physique dans ledit réseau et pouvant avoir un ou plusieurs noms logiques associés soit au noeud, soit aux entités respectives qu'il dessert, comprenant les étapes suivantes consistant à:

maintenir des tables d'association de noms auxdits noeuds pour stocker des indications des noms logiques associés aux entités respectivement desservies;

à un noeud cherchant à établir une session entre une première entité respectivement desservie et ayant un premier nom logique, et une seconde entité ayant un second nom logique, comparer ledit second nom logique avec les noms logiques contenus dans la table d'association de noms maintenue au noeud correspondant;

établir une session entre lesdites première et seconde entités au moyen d'opérations de traitement entièrement effectuées dans le noeud intéressé,—c'est-à-dire sans aucune communication effectuée dans ledit réseau—,si le second nom logique correspond à un nom logique contenu dans la table d'association de noms concernée; et si le second nom logique ne correspond à aucun nom logique dans la table d'association de noms concernée, effectuer successivement les étapes suivantes:

diffuser un message de vérification de Nom d'Appel dans ledit réseau depuis le noeud cherchant à établir ladite session, ledit message contenant ledit second nom logique et l'adresse, dans ledit réseau, du noeud qui est à l'origine de la diffusion;

décompter un intervalle de temps prédéterminé audit noeud qui est à l'origine de ladite diffusion immédiatement après cette dernière;

recevoir aux autres noeuds dudit réseau ledit message de vérification de Nom d'Appel;

à chacun desdits autres noeuds, comparer le second nom logique contenu dans ledit message reçu avec les noms logiques contenus dans la table d'association de noms concernée et transmettre un message constituant un accusé de réception à l'adresse spécifique du noeud qui diffuse le message de Vérification de Nom si, et seulement si, une correspondance est détectée entre ledit second nom logique et un nom logique contenu dans la table d'association de nom concernée;

au noeud qui est à l'origine de la diffusion du message de vérification de Nom d'Appel, superviser les accusés de réception de ladite diffusion reçus pendant ledit délai d'attente; et

sous réserve desdits accusés de réception reçus, soit déclencher des communications supplémentaires avec un autre noeud qui a transmis un accusé de réception, afin de poursuivre le processus de recherche d'établissement de ladite session, soit mettre fin audit processus.

2. Procédé pour établir une session de communication selon la revendication 1, dans lequel lesdites communications supplémentaires consistent notamment à:

transmettre une Demande d'Appel à ladite seconde entité, depuis ledit noeud cherchant à poursuivre ledit processus; et

transmettre un accusé de réception en réponse à ladite Demande d'Appel, du noeud desservant ladite seconde entité au noeud dont émane ladite Demande d'Appel.

3. Procédé selon la revendication 2, dans lequel lesdits noeuds sont adaptés de façon à établir des associations dites d'Etat d'Ecoute entre les entités respectivement desservies et d'autres entités, afin de réguler les communications entre entités, et ledit second accusé de réception n'est transmis qu'à la condition qu'une association d'Etat d'Ecoute pré-existe entre ledit second nom logique et ledit premier nom logique à un noeud desservant une entité associée audit second nom logique et recevant une desdites Demandes d'Appel.

4. Procédé selon la revendication 1, consistant notamment à:

poursuivre ledit processus si un seul accusé de réception est reçu en réponse audit message diffusé de vérification de Nom d'Appel et à mettre fin audit processus si aucun accusé de réception n'est reçu, ou si plusieurs accusés de réception sont reçus, en réponse audit message diffusé de vérification de Nom d'Appel, et

en réponse à la réception de plus d'un seul accusé de réception dudit message diffusé de vérification de Nom d'Appel, diffuser un message dit de Duplication d'Emploi dans ledit réseau depuis ledit noeud qui est à l'origine du message diffusé de vérification de Nom d'Appel, ledit message de Duplication d'Emploi indiquant l'existence d'un conflit relatif à l'emploi de noms logiques et ayant pour effet d'informer les systèmes de calcul des noeuds qui ont accusé réception du message diffusé de vérification de Nom d'Appel du fait qu'il existe un conflit en ce qui concerne leur utilisation dudit second nom logique.