SINGLE SERVER ACCESS IN A MULTIPLE TCP/IP INSTANCE ENVIRONMENT

Inventors: George Eisenberger, White Plains, NY (US); Edgard H. McCulloch, Arlington, VA (US); Thomas L. Richards, Chicago, IL (US)

Correspondence Address:
DILLON & YUDELL LLP
8911 N. CAPITAL OF TEXAS HWY., SUITE 2110
AUSTIN, TX 78759 (US)

Publication Classification
Int. Cl. G06F 15/16 (2006.01)
U.S. Cl. 709/228

Abstract
Methods, systems and computer program products are provided that access a single server application executing on a processing system having multiple TCP/IP instances. The single server application may be accessed by establishing a registration repository containing information sufficient to identify to both the single server application and a client application seeking to access the single server application, one of the multiple TCP/IP instances through which the single server application may be accessed by client applications. In such a case, the registration repository is accessible by the multiple TCP/IP instances and the single server application.
FIG. 2

Start

105

Receive a request to publish a selected topic from a Subscriber

110

Assess whether the subscriber is authorized to receive data on the selected topic from participating Publishers

115

Provide a publication request to Publishers.

116

Authorize an approved subscription for an approved topic from a Publisher for a particular Subscriber

End
FIG. 3

Start

201
Receive notification of a request to publish for an identified Subscriber and selected topic

202
Determine whether to approve or deny the publication request based on the selected topic and/or the identified Subscriber

203
Electronically search and extract electronic clinical patient data records that match the selected topic for approved publication requests

204
Transmit the extracted Publisher clinical data to the message flow server

End

Remove patient personal identifiers from the extracted messages
FIG. 4

Processor 138

User Registry Module 120

Publisher/Subscriber Interface Module 124

Subscriber Accessible Topic Catalog 125

Application Programs 154

Operating System 152

Memory 136

User Profiles 126

Data 156

I/O Device Drivers 158

I/O Circuits 146
### Sample view of notifications on the HCN Portal

**FIG. 14**

#### Healthcare Collaboration Network

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Type</th>
<th>From</th>
<th>Subject</th>
</tr>
</thead>
<tbody>
<tr>
<td>03/25/YY</td>
<td>12:00</td>
<td>Email</td>
<td>Administrator</td>
<td>System Unavailable</td>
</tr>
<tr>
<td>03/25/YY</td>
<td>13:25</td>
<td>Information</td>
<td>Administrator</td>
<td>Publication added to topic Anthrax</td>
</tr>
<tr>
<td>03/25/YY</td>
<td>15:44</td>
<td>Information</td>
<td>&quot;A&quot; General Hospital</td>
<td>New Health Topic Diabetes has been added</td>
</tr>
<tr>
<td>03/24/YY</td>
<td>10:27</td>
<td>Email</td>
<td>&quot;A&quot; General Hospital</td>
<td>New Health Topic Diabetes has been added</td>
</tr>
<tr>
<td>03/23/YY</td>
<td>11:11</td>
<td>System</td>
<td>&quot;A&quot; General Hospital</td>
<td>New Health Topic Diabetes has been added</td>
</tr>
<tr>
<td>03/23/YY</td>
<td>16:03</td>
<td>System</td>
<td>&quot;A&quot; General Hospital</td>
<td>New Health Topic Diabetes has been added</td>
</tr>
</tbody>
</table>

---

#### Publication Requests

The following table lists subscriber requests for publications:

<table>
<thead>
<tr>
<th>Topic Name</th>
<th>Requesting Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accutane with no pregnancy test</td>
<td>FDA</td>
</tr>
<tr>
<td>ADR for Accutane</td>
<td>FDA</td>
</tr>
<tr>
<td>ADR for Accutane</td>
<td>Centers for Medicare and Medicaid</td>
</tr>
</tbody>
</table>

---

#### Health Topics

Display Topics:
- All topics
- Containing the word:

---

#### Participants

Display Participants:
- All participants
- Beginning with letters:

---
FIG. 16B

The following tables lists the publications your publishing. Click on a Publication name to view details.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>No. of Subscribers</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMI Diagnosis - KPI Demo</td>
<td>AMI patients who had Aspirin ordered within 24 hours of admission</td>
<td>1</td>
</tr>
</tbody>
</table>

home  contact us  legal  help  log out  → return to publications page
Alert messages are sent to all subscribers of the same "Health Topic" for this ADE.  

FIG. 18  
System for Monitoring and Alerting of Adverse Drug Events in Near-Realtime
FIG. 19
System for Monitoring and Alerting of Disease/Exposures in Near-realtime Using ME

Alert messages are sent to all Subscribers of the same "Health Topic" for this ADE.
FIG. 21A

Administrative Server with Privacy Level Register: 1100R

Message Flow Server 100

Privacy Level A 300₁

Privacy Level B 300₂

Privacy Level C 300₃

Topic Catalog
Topic 1, PL A
Topic 2, PL B
Topic 3, PL C 1101

Privacy Level B 200₁

Privacy Level B 200₂

Privacy Level B 200₃
FIG. 21B

Administrative Server with Privacy Level Register: 1100R

1100

Topic Catalog
Topic 1, PL A
Topic 2, PL B
Topic 3, PL C

1101

A Topic List
B Topic List
C Topic List

Privacy Level A
3001

Privacy Level B
3002

Privacy Level C
3003

Privacy Level A
2001

Privacy Level B
2002

Privacy Level A
2003
FIG. 22

<table>
<thead>
<tr>
<th>Subscriber</th>
<th>Priv Level</th>
<th>Topic</th>
<th>Publisher</th>
<th>Subscription?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sub₁</td>
<td>A</td>
<td>1</td>
<td>Pub 1</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Pub 2</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Pub 3</td>
<td>N</td>
</tr>
<tr>
<td>Sub₂</td>
<td>B</td>
<td>2</td>
<td>Pub 1</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Pub 2</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Pub 3</td>
<td>Y</td>
</tr>
</tbody>
</table>

MESSAGE FLOW SERVER 100
SINGLE SERVER ACCESS IN A MULTIPLE TCP/IP INSTANCE ENVIRONMENT

PRIORITY CLAIM


FIELD OF THE INVENTION

[0002] The present invention relates to computer systems in general and more particularly to computer systems employing the Transmission Control Protocol/Internet Protocol (TCP/IP) to access server applications.

BACKGROUND OF THE INVENTION

[0003] In recent years there has been an increase in interest in the Internet and Internet applications. The Internet is a client-server paradigm where clients access server applications to obtain information. This proliferation of Internet users has led to many networks employing the same client-server paradigm in an internal network known as an intranet. These intranets are often built on the same communications protocols as the Internet. Thus a user or client may communicate with a server on an internal network or on the Internet in a seamless fashion such that the location of the server and access to that server may be transparent to a user.

[0004] The communication protocol on which the Internet is based is known as the Transmission Control Protocol/Internet Protocol (TCP/IP). The popularity of the Internet and of intranets has led to the proliferation of TCP/IP throughout the computing community. This proliferation has led to TCP/IP being utilized on local area networks (LANs), wide area networks (WANS) and even on mainframe computers such as the International Business Machines Corporation’s OS/390 systems.

[0005] To communicate using TCP/IP, a computer system typically has a TCP/IP protocol stack executing on the computer. A TCP/IP protocol stack or TCP/IP instance governs the communications between the client and the server. If the computer systems capabilities allow, a computer system may have multiple TCP/IP instances communicating with multiple clients and servers. Thus, as is illustrated in FIG. 1, a computer system 22 may have multiple TCP/IP instances (illustrated by TCP/IP A 12 and TCP/IP B 16) and multiple servers (illustrated by Server A 10 and Server B 14) communicating with multiple clients (illustrated by Client A 18 and Client B 20).

[0006] One typical constraint of the client-server model illustrated in FIG. 1, is that applications such as Server A 10 and Server B 14 may not span multiple TCP/IP instances. Thus, where multiple TCP/IP instance exist on a single computer system a particular server is limited to the IP address domain of the TCP/IP instance to which it is connected. Thus, to support clients connecting to the computer system through more than one of the computer systems TCP/IP instances a duplicate of the server should be connected to each of the TCP/IP instances. Thus, to make a server application available to all IP address domains of a computer system the computer system is generally either limited to a single TCP/IP instance or to duplicate copies of a server application for each TCP/IP instance of the computer system.

[0007] While in some situations multiple copies of a server application may be acceptable, in certain situations it is not possible to have multiple copies of a server on a single computer system. Limitations on the number of allowable server applications may be imposed architecturally by the server application or may result from hardware or software limitations for the system. For example, only a single instance of the OS/390 OpenEdition Open System Adapter Support Facility (OSA/SF) may run in a MVS partition although up to 8 TCP/IP instances are allowed. Thus, in order to be available to all clients in a computer system’s IP domain then only a single TCP/IP instance may be utilized on a computer system with such a server.

[0008] Even if a particular server application allows for multiple instances in a single computer system and the computer system’s hardware and software are capable of multiple copies of a server there may still be disadvantages to multiple copies of a server on a single system. For example, system resources are still occupied by multiple copies of a server application. Central Processing Unit (CPU) usage may be increased as well as storage and memory requirements. Furthermore, synchronization of multiple copies of a server application to provide consistent server responses across multiple TCP/IP instances may also present difficulties.

[0009] In view of the above discussion, there exists a need for improvement in computer systems having multiple TCP/IP instances to allow access to server applications across the IP domain’s of multiple TCP/IP instances.

SUMMARY OF THE INVENTION

[0010] In light of the limitations of existing client-server systems, it is an object of the present invention to provide for the access to server applications across multiple TCP/IP instances.

[0011] A further object of the present invention is to provide access to server applications across multiple TCP/IP instances without requiring the system resources of replicating the server application.

[0012] Another object of the present invention is to provide access across multiple TCP/IP instances to a server application for which multiple copies cannot be provided on a single computer system.

[0013] These and other objects of the present invention are provided by methods, systems and computer program products that access a single server application executing on a processing system having multiple TCP/IP instances. The single server application may be accessed by establishing a registration repository containing information sufficient to identify both the single server application and a client application seeking to access the single server application, one of the multiple TCP/IP instances through which the single server application may be accessed by client applications. In such a case, the registration repository is accessible by the multiple TCP/IP instances and the single server application.

[0014] By establishing a registration repository in which clients may determine which TCP/IP instance of multiple TCP/IP instances to access to reach a single server application, the client may connect through that TCP/IP instance to reach the single server application. Thus, only a single instance of the server application may be provided on the computer system and may be accessed by any client with access to the registration repository. Furthermore, because multiple copies of a server application are not required there may be fewer resources used to allow access from all of the IP
domains of the computer system. Also, because the multiple copies of the server application are not required to provide access to the server applications to clients of multiple TCP/IP instances, the present invention allows server application which cannot have multiple copies to be accessible across multiple TCP/IP instances.

In a particular embodiment of the present invention, the registration repository is established by registering a TCP/IP instance with the registration repository. If a single server access is requested by a client application utilizing the TCP/IP instance, in such a case, the registration may include storing an entry identifying an available TCP/IP instance. A time stamp associated with the entry identifying the available TCP/IP instance which identifies when the TCP/IP instance was registered with the registration repository may also be stored in the registration repository.

In a preferred embodiment, the registration repository includes the TCP/IP instance identifier, the function of the server application associated with the TCP/IP instance and protocol information sufficient to allow connection between the TCP/IP instance and the client.

In another embodiment of the present invention, entries for a TCP/IP instance are removed from the registration repository when a TCP/IP instance is no longer available for connection to the single server application or client access to the single server application. In such a case, the entry may be removed by invalidating the registration of the TCP/IP instance in the registration repository.

In another embodiment of the present invention, the single server application selects a TCP/IP instance from the registration repository to be used for connection to the single server application. In a particular aspect of the present invention, an entry identifying an available TCP/IP instance is stored in the registration repository. A time stamp associated with the entry identifying the available TCP/IP instance is also stored in the registration repository. Then, the selection of the TCP/IP instance comprises selecting the TCP/IP instance with the earliest time stamp.

In still another embodiment of the present invention wherein the client application accesses the registration repository to determine the TCP/IP instance through which the single server application may be accessed. The client application then connects to the single server application by connecting to the TCP/IP instance through which the single server application may be accessed. In such a case, the client application may determine the TCP/IP instance through which the single server application may be accessed utilizing the same procedure by which the single server application selects a TCP/IP instance. Furthermore, an entry identifying an available TCP/IP instance and a time stamp associated with the entry identifying the available TCP/IP instance in the registration repository may be stored in the registration repository. Then access of the registration repository may include accessing the registration repository to obtain the time stamp of entries in the registration repository and selecting the TCP/IP instance with the earliest time stamp as the TCP/IP instance through which the single server application may be accessed.

In another embodiment of the present invention, the registration repository is stored in common storage. Accessing the registration repository then includes accessing the common storage location of the registration repository and obtaining from the common storage location registration information to determine the TCP/IP instance through which the single server application may be accessed. Furthermore, the single server application may then be connected to by connecting to the determined TCP/IP instance through which the single server application may be accessed. Also, the common storage may be directly accessible by the client.

In yet another embodiment of the present invention, the common storage is accessed by accessing a first TCP/IP instance to access the common storage location of the registration repository. Registration information is then obtained from the common storage location through the first TCP/IP instance and the single server application connected to through a second TCP/IP instance.

As will further be appreciated by those of skill in the art, the present invention may be embodied as a method, apparatus or computer program product.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram of a conventional multiple client/multiple server system:

Fig. 2 is a block diagram of a client-server system according to the present invention:

Fig. 3 is a flow chart illustrating the creation of an entry in a registration repository according to the present invention;

Fig. 4 is a flow chart illustrating the operation of the present invention; and

Fig. 5 is a block diagram of an example system utilizing the present invention.

DETAILED DESCRIPTION OF THE INVENTION

The present invention now will be described more fully hereinafter with reference to the accompanying drawings, in which preferred embodiments of the invention are shown. This invention may, however, be embodied in many different forms and should not be construed as limited to the embodiments set forth herein; rather, these embodiments are provided so that this disclosure will be thorough and complete, and will fully convey the scope of the invention to those skilled in the art. Like numbers refer to like elements throughout. As will be appreciated by one of skill in the art, the present invention may be embodied as methods or devices. Accordingly, the present invention may take the form of an entirely hardware embodiment, an entirely software embodiment or an embodiment combining software and hardware aspects.

Fig. 2 illustrates a computer system utilizing the present invention. As seen in Fig. 2, a computer system 35 includes a single server application (Server A 30) and multiple TCP/IP instances (TCP/IP A 32 and TCP/IP B 34). Connected to the TCP/IP instances 32 and 34 are client applications (Client A 38 and Client B 39) to the server application 30. The computer system 35 may be any computer system capable of supporting multiple TCP/IP protocol stacks, however, the present invention is particularly well suited to mainframe computers such as International Business Machines Corporation's OS/390 computer systems utilizing the Multiple Virtual System (MVS) operating system. Thus, as will be appreciated by those of skill in the art, computer system 35 will typically include a processing unit, random access memory, non-volatile memory, mass storage devices, input devices, output devices and communication devices as well as
software applications and an operating system to control the various devices and application associated with computer system 35.

[0030] As is further illustrated in FIG. 2, a registration repository 36 is provided in computer system 35. The registration repository is accessible by the TCP/IP instances 32 and 34 as well as the server 30 and client applications 38 and 39. The registration repository is preferably directly accessible by the clients 38 and 39 as is illustrated in FIG. 2, but may also be accessible indirectly through, for example, the TCP/IP instances 32 and 34. The registration repository 36 is preferably implemented in common storage of computer system 35. Thus, for example, in an MVS system where the registration repository is implemented in common storage, a client may directly address the common storage associated with the registration repository to retrieve entries from the registration repository. Similarly, the server application could directly access the common storage to retrieve entry information.

[0031] In a further aspect of the present invention where the computer system 35 is a logical unit in a Sysplex the registration repository could be implemented in the coupling facility of the Sysplex. Thus, a single server application could be shared across logical units of the Sysplex and provide single server access to the IP address domain of all of the logical units with access to the coupling facility.

[0032] In operation, the TCP/IP instances 32 and 34 create entries in the registration repository uniquely identifying the TCP/IP instance. When a server application is started it selects a TCP/IP instance from the entries in the registration repository. When a client seeks to access a single server application 30, the client 39 accesses the registration repository 36 and retrieves the information indicating which TCP/IP instance allows access to the single server application 30. This selection process may be accomplished by the clients using the same selection mechanism as the server applications in selecting from the TCP/IP instances registered with the registration repository. The client 39 then connects to the TCP/IP instance 32 with access to the single server application 30.

[0033] Alternatively, a server application could access the registration repository to determine which TCP/IP instances allow single server access and then select one of the TCP/IP instances. The server could then update the registration repository to indicate that the server application was associated with the selected TCP/IP instance. Clients would then search the entries in the registration repository to find an entry identifying the desired server application and then utilize the associated TCP/IP instance to access the server application. However, such an implementation is essentially a special case of the first alternative. Accordingly, the detailed operation described below focuses primarily on the first alternative briefly described above. Those of skill in the art will understand that the special case may be readily implemented once the principles of the general case are understood.

[0034] The present invention will now be described with respect to FIG. 3 and FIG. 4 which are flowcharts illustrating the operation of a routing device and a master routing device utilizing the present invention. It will be understood that each block of the flowchart illustrations, and combinations of blocks in the flowchart illustrations, can be implemented by computer program instructions. These program instructions may be provided to a processor to produce a machine, such that the instructions which execute on the processor create means for implementing the functions specified in the flowchart block or blocks. The computer program instructions may be executed by a processor to cause a series of operational steps to be performed by the processor to produce a computer implemented process such that the instructions which execute on the processor provide steps for implementing the functions specified in the flowchart block or blocks.

[0035] Accordingly, blocks of the flowchart illustration support combinations of means for performing the specified functions, combinations of steps for performing the specified functions and program instruction means for performing the specified functions. It will also be understood that each block of the flowchart illustration, and combinations of blocks in the flowchart illustration, can be implemented by special purpose hardware-based systems which perform the specified functions or steps, or combinations of special purpose hardware and computer instructions.

[0036] FIG. 3 illustrates the creation of an entry in the registration repository 36 when a TCP/IP instance is started. When a TCP/IP instance is configured to require access to a single server application, either initially or dynamically after startup of the TCP/IP instance, an entry is added to the registration repository. As seen in FIG. 3, a TCP/IP instance is established (block 40) and determines in a client requires access to a single server application (block 42). If single server access is not required then the TCP/IP instance enters a wait loop (block 44) until a client request single server access.

[0037] If a client initially requests single server access or subsequently request single server access, then the TCP/IP instance selects a time stamp for incorporation in the registration repository entry (block 46). The time stamp acts as a unique identifier of the TCP/IP instance and prioritizes the TCP/IP instances for selection by clients and server applications. The time stamp may be the time the TCP/IP instance was started or when the registration request was made. Whatever manner of selecting a time stamp is utilized it needs to be the same for all TCP/IP instances.

[0038] Alternatively, other methods of providing unique priority for the TCP/IP instances could be utilized. For example, priority tokens could be selected by TCP/IP instances or a priority count could be established and incremented each time a TCP/IP instance is initiated. What is required is a manner of identifying TCP/IP instances such that the client and the server may utilize the same selection procedure to reach the same selection result. Thus, whatever manner of prioritizing TCP/IP instances is utilized it should be utilized in the same manner for all TCP/IP instances.

[0039] Returning to FIG. 3, after selection of a time stamp, the TCP/IP instance adds an entry to the registration repository 36 for the TCP/IP instance (block 48). The entry in the repository preferably includes the TCP/IP instance identifier and protocol information sufficient to allow connection between the TCP/IP instance and the client. As briefly discussed above, the entry may also include the function of the server application associated with the TCP/IP instance.

[0040] By creating an entry in the registration repository 36 if a client requests access to a single server application, the present invention assures that at least one TCP/IP instance will be available in the registration repository 36 for selection by the server application. For example, in FIG. 2, if client 38 is the first client associated computer system 35 that request access to a single server application, then upon receiving the request TCP/IP A 32 will create an entry in registration
repository 36 to identify that it has received a request for single server access and to become available for selection by server 30 for use. Thus, even if a request for single server access is the first request received a TCP/IP instance will be present in the registration repository for use by the server application and the client. Additionally, by registering with the registration repository on a per TCP/IP instance basis, client access to a server application may be allowed or disallowed on a per TCP/IP instance basis.

[0041] With respect to the server application, a server application accesses the registration repository at start-up to select a TCP/IP instance. If no TCP/IP instances are identified in the registration repository, the server enters a wait loop until one is entered. The server then selects the TCP/IP instance and connects to that instance. Optionally, the server may update the registration repository to indicate that the server is connected to the TCP/IP instance. The server preferably selects the TCP/IP instance based upon the time stamp of the TCP/IP instance entry in the registration repository. However, other methods of selection may be utilized as long as the clients also use the same selection procedure. If the TCP/IP instance to which the server is connected terminates, the server re-accesses the registration repository to select a new TCP/IP instance for connection.

[0042] FIG. 4 illustrates the operation of a client communication utilizing the present invention. When a client seeks to access a single server application, the client first accesses the registration repository 36 to retrieve the entry with the earliest time stamp (block 50). The client will utilize this entry to attempt to connect to the server. The server also accesses the repository to establish connections and uses the same selection process as clients. Therefore, even if the TCP/IP instance is one which is in the process of terminating, both the client and the server should eventually select the same TCP/IP instance if the same selection process is utilized.

[0043] After retrieving the entry from the registration repository, the client determines the TCP/IP instance through which a connection may be established to the server (block 52). As discussed above, this may be the TCP/IP instance corresponding to the earliest time stamp entry in the registration repository. However, other criteria for selecting the TCP/IP instance may be utilized as long as the client and the server use the same criteria. The client then connects to the server using the determined TCP/IP instance (block 54). Such a connection is possible because, as discussed above, the registration entry contains sufficient information to allow for a connection between the TCP/IP instance, the client and the server.

[0044] In the event of an error or termination of the TCP/IP instance, the TCP/IP instance notifies the clients and servers utilizing the TCP/IP instance of the error. The terminating TCP/IP instance is removed from the registration repository. The clients and servers then would re-access the registration repository to select a different TCP/IP instance for re-establishing connections or wait until an eligible TCP/IP instance is available (i.e. registered in the repository). Again, because the clients and servers would utilize the same process for selecting a TCP/IP instance a connection should be capable of being established.

[0045] The terms “single server access” and “single server application” have been used herein to describe the present invention. As used herein, those terms refer to instance of a server application which span multiple TCP/IP instances in that clients from multiple IP domains may access a single instance of a server application.

[0046] The present invention may be further understood by way of example. FIG. 5 illustrates one computer system in which the present invention may be utilized. As seen in FIG. 5, a computer system utilizes OSA/SF in an MVS partition. Multiple TCP/IP instances also include SNMP support.

[0047] In the example depicted by FIG. 5, the two TCP/IP instances, TCP/IP A 64 and TCP/IP B 66, were started with both requiring client access to the single server application represented by OSA/SF 62. OSA/SF (Open System Adapter/Support Facility) is an application that manages OSA/Adapters on a OS/390 system. Only a single instance of OSA/SF can be started per MVS system due to architecture constraints.

[0048] TCP/IP A 64 is identified in the Repository 68 as the first eligible TCP/IP instance for single server application access to OSA/SF 62. The single server application, OSA/SF 62 connects to TCP/IP A 64 after accessing the Repository 68 to determine that TCP/IP A 64 corresponds to the entry in the repository with the earliest time stamp.

[0049] At some point, two OSA/SF Client Applications, SNMP Agent 70 and SNMP Agent 72 are also started. Each of the client applications, SNMP Agent 70 and SNMP Agent 72, connects into their respective TCP/IP instance in order to receive requests from the associating TCP/IP Networks, 74 and 76.

[0050] Each TCP/IP instance requires separate SNMP Agents in order to provide network management support for each of the respective IP Address domains for TCP/IP Network 74 and TCP/IP Network 76. SNMP (Simple Network Management Protocol) is a standard management protocol implemented by many TCP/IP Hosts. The SNMP Agents handle all of the related SNMP protocol functions but require access to OSA/SF 62 in order to retrieve and set configuration variables specific to the adapters managed by OSA/SF.

[0051] When the SNMP Agent 70 is started, it connects to the single server application OSA/SF 62 via TCP/IP A 64 after accessing the Repository 68 to determine that TCP/IP A 64 is the first eligible TCP/IP instance. The client application SNMP Agent 72 after accessing the Repository 68 also establishes a connection to the Single Server application OSA/SF 62 through TCP/IP A 64. Note that both client applications actually have two connections established. The first is to their respective TCP/IP instance in order to receive and respond to requests from their respective TCP/IP Networks 74 and 76. Both SNMP Agent 70 and 72 establish a second connection through TCP/IP A 64 in order to gain access to the single server application, OSA/SF 62. In FIG. 5, the SNMP Agents (clients) and OSA/SF (server) connections are depicted using dashed lines. Thus, a single server application such as OSA/SF may be accessed by multiple IP address domains through multiple TCP/IP instances where only a single TCP/IP instance is connected directly to the server application.

[0052] In the drawings and specification, there have been disclosed typical preferred embodiments of the invention and, although specific terms are employed, they are used in a generic and descriptive sense only and not for purposes of limitation, the scope of the invention being set forth in the following claims.

That which is claimed is:

1. A system for accessing a single server application executing on a processing system having multiple TCP/IP instances comprising:
a processor;
a registration repository containing information sufficient
to identify to both the single server application and a
client application seeking to access the single server
application, one of the multiple TCP/IP instances
through which the single server application may be
accessed by client applications and wherein the
registration repository is accessible by the multiple TCP/IP
instances and the single server application; and
a utility which executes on the processor and provides
means for accessing the registration repository to select
a TCP/IP instance for connection to the server application.

2. The system according to claim 1, further comprising:
means for establishing the registration repository; and
means for registering a TCP/IP instance with the regis-
tration repository if single server access is requested by a
client application utilizing the TCP/IP instance.

3. The system according to claim 2, wherein said means for
registering comprises the steps of:

storing an entry identifying an available TCP/IP instance;
and
storing a time stamp associated with the entry identifying
the available TCP/IP instance which identifies when the
TCP/IP instance was registered with the registration
repository.

4. The system according to claim 3, wherein the entry
identifying the TCP/IP instance includes a TCP/IP instance
identifier and protocol information sufficient to allow con-
nection between the TCP/IP instance and the client.

5. The system according to claim 2, further comprising
means for removing from the registration repository a TCP/IP
instance which is no longer available for connection to the
single server application or client access to the single server
application.

6. The system according to claim 5, wherein said means for
removing comprises means for invalidating the registration of
the TCP/IP instance in the registration repository.

7. The system according to claim 1, wherein the single
server application provides the means for selecting a TCP/IP
instance from the registration repository to be used for con-
nexion to the single server application.

8. The system according to claim 7, further comprising:
means for storing an entry identifying an available TCP/IP
instance in the registration repository;
means for storing a time stamp associated with the entry
identifying the available TCP/IP instance in the registra-
tion repository; and

wherein said means for selecting a TCP/IP instance com-
prises means for selecting the TCP/IP instance with the
earliest time stamp.

9. The system according to claim 7, wherein the client
application provides:
means for accessing the registration repository to de-
termine the TCP/IP instance through which the single
server application may be accessed; and
means for connecting to the single server application by
connecting to the TCP/IP instance through which the
single server application may be accessed.

10. The system according to claim 9, wherein the client
application includes means for determining the TCP/IP
instance through which the single server application may be
accessed utilizing the same procedure by which the single
server application selects a TCP/IP instance.

11. The system according to claim 9, further comprising:
means for storing an entry identifying an available TCP/IP
instance in the registration repository; and
means for storing a time stamp associated with the entry
identifying the available TCP/IP instance in the registra-
tion repository; and

wherein said means for accessing the registration reposi-
tory comprises:
means for accessing the registration repository to obtain
the time stamp of entries in the registration repository; and

means for selecting the TCP/IP instance with the earliest
time stamp as the TCP/IP instance through which the
single server application may be accessed.

12. The system according to claim 9, wherein the registra-
tion repository is stored in common storage and wherein said
means for accessing the registration repository comprises:
means for accessing the common storage location of the
registration repository; and
means for obtaining from the common storage location
registration information to determine the TCP/IP
instance through which the single server application
may be accessed; and

wherein said means for connecting to the single server
application comprises means for connecting to the deter-
mined TCP/IP instance through which the single server
application may be accessed.

13. The system according to claim 12, wherein said common
storage is directly accessible by the client and wherein
said means for accessing the common storage comprises
means for directly accessing the common storage.

14. A system according to claim 12, wherein:
said means for accessing the common storage comprises
means for accessing a first TCP/IP instance to access the
common storage location of the registration repository;
said means for obtaining registration information comprises
means for obtaining registration information from the
common storage location through the first TCP/IP
instance; and

said means for connecting to the single server application
comprises means for connecting to a second TCP/IP
instance through which the single server application
may be accessed.

15. A computer program product for accessing a single
server application executing on a processing system having
multiple TCP/IP instances comprising:
a computer-readable storage medium having computer-
readable program means embodied in said medium, said
computer-readable program means comprising:
program means for establishing a registration repository
containing information sufficient to identify to both
the single server application and a client application
seeking to access the single server application, one of
the multiple TCP/IP instances through which the
single server application may be accessed by client
applications and wherein the registration repository is
accessible by the multiple TCP/IP instances and the
single server application; and

program means for accessing the registration repository
to select a TCP/IP instance for connection to the
server application.

16. The computer program product according to claim 15,
wherein said program means for establishing a registration
repository comprises program means for registering a TCP/IP
instance with the registration repository if single server access is requested by a client application utilizing the TCP/IP instance.

17. The computer program product according to claim 16, wherein said program means for registering comprises program means for:
   storing an entry identifying an available TCP/IP instance;
   and
   storing a time stamp associated with the entry identifying the available TCP/IP instance which identifies when the TCP/IP instance was registered with the registration repository.

18. The computer program product according to claim 17, wherein the entry identifying the TCP/IP instance includes a TCP/IP instance identifier and protocol information sufficient to allow connection between the TCP/IP instance and the client.

19. The computer program product according to claim 16, further comprising program means for removing from the registration repository a TCP/IP instance which is no longer available for connection to the single server application or client access to the single server application.

20. The computer program product according to claim 19, wherein program means for removing comprises program means for invalidating the registration of the TCP/IP instance in the registration repository.

21. The computer program product according to claim 15, wherein the single server application provides the program means for connecting to the single server application.

22. The computer program product according to claim 21, further comprising:
   program means for storing an entry identifying an available TCP/IP instance in the registration repository and
   program means for storing a time stamp associated with the entry identifying the available TCP/IP instance in the registration repository; and
   wherein said program means for connecting to the single server application comprises program means for selecting the TCP/IP instance with the earliest time stamp.

23. The computer program product according to claim 21, wherein the client application provides:
   program means for accessing the registration repository to determine the TCP/IP instance through which the single server application may be accessed;
   and
   program means for connecting to the single server application by connecting to the TCP/IP instance through which the single server application may be accessed.

24. The computer program product according to claim 23, wherein the client application provides program means for determining the TCP/IP instance through which the single server application may be accessed utilizing the same procedure by which the single server application selects a TCP/IP instance.

25. The computer program product according to claim 23, further comprising:
   program means for storing an entry identifying an available TCP/IP instance in the registration repository; and
   program means for storing a time stamp associated with the entry identifying the available TCP/IP instance in the registration repository; and
   wherein said program means for accessing the registration repository comprises:
   program means for accessing the registration repository to obtain the time stamp of entries in the registration repository and
   program means for selecting the TCP/IP instance with the earliest time stamp as the TCP/IP instance through which the single server application may be accessed.

26. The computer program product according to claim 23, wherein the registration repository is stored in common storage and wherein said program means for accessing the registration repository comprises:
   program means for accessing the common storage location of the registration repository; and
   program means for obtaining from the common storage location registration information to determine the TCP/IP instance through which the single server application may be accessed; and
   wherein said program means for connecting to the single server application comprises program means for connecting to the determined TCP/IP instance through which the single server application may be accessed.

27. The computer program product according to claim 26, wherein said common storage is directly accessible by the client and wherein said program means for accessing the common storage comprises program means for directly accessing the common storage.

28. The computer program product according to claim 26, wherein said program means for accessing the common storage comprises program means for accessing a first TCP/IP instance to access the common storage location of the registration repository.
   wherein said program means for obtaining registration information comprises program means for obtaining registration information from the common storage location through the first TCP/IP instance; and
   wherein said program means for connecting to the single server application comprises program means for connecting to a second TCP/IP instance through which the single server application may be accessed.

* * * * *