XML BASED TRANSACTION DETAIL RECORDS

The present invention relates to telecommunications, and is more particularly related to recording transaction data in a telecommunications network.

There are many factors driving the move toward converged networks such as deregulation, new sources of competition, substantial growth of the Internet, and the growth and importance of data in customers' enterprise networks. The popularity and convenience of the Internet has resulted in the reinvention of traditional telephony services. IP(Internet Protocol) telephony, which is also referred to as Voice-over-IP (VoIP), involves the conversion of voice information into data packets that are subsequently transmitted over an IP network. IP telephony over the Internet is often offered at minimal, or no cost to the users. Thus, IP telephony has found significant success, particularly in the long distance market.

Users also have turned to IP telephony as a matter of convenience. Both voice and data services are often accessible through a single piece of equipment, e.g., the personal computer. Furthermore, traditional DTMF (Dual-Tone Multi-Frequency) phones can enjoy the benefits of VoIP technology through the use of network adapters. The continual integration of voice and data services further fuels this demand for IP telephony applications.

The primary incentives for customers to adopt a converged solution are cost and the promise of new and expanded capabilities. However, if IP telephony is to be fully accepted in the marketplace, VoIP must be interoperable with the Public Switched Telephone Network (PSTN) and have a comparable Quality of Service (QoS). Therefore, to ensure the highest success rate with the customers, the service providers need to build a network that provides call quality, service reliability, and security that is at minimum, on par with the PSTN. It is essential that IP Telephony solutions meet customer demands of high-quality, ease of use, superior customer service, and lower cost. Since the public Internet can only provide "best-efforts" service, managed IP networks are required to support VoIP traffic with the call quality, service reliability, and security that users are accustomed to.

One approach that is being considered in providing VoIP with the call quality, service reliability, and security that users are accustomed to, involves the Session Initiation Protocol (SIP). SIP is an application-layer signaling protocol that has been developed to create, modify, and terminate sessions with one or more users. These sessions include Internet telephone calls, multi-media conferences, and multi-media distribution. SIP functionality is typically resident on application servers. Sip servers are configured to provide telephony

services, and process call event information. Because vendors have developed their own custom SIP application programs, call events and telephony services are processed by each vendor's application server in a proprietary way. Unfortunately, when a network includes network elements provided by a multiplicity of vendors, it becomes necessary to accommodate a variety of proprietary interfaces that enable the devices to transmit and receive network transaction data. By way of example, transaction data may include call event information, billing information, monitoring information, error data, fraud prevention data, timeout data and any other data that must be tracked by the network.

What is needed is a platform independent method for capturing transaction data in a uniform manner. Preferably, the system and method will be extensible, providing embedded information that will enable a receiving computer to read the generic, uniformly formatted records without needing to accommodate any proprietary interface.

The present invention relates to a platform independent method for capturing transaction data and other information in a uniform manner. The method and system of the present invention is extensible, producing generic, uniformly formatted records that can be read by a receiving computer without needing a special proprietary interface.

One aspect of the present invention is a method for recording transactions in a telecommunications network. The method includes creating an XML transaction detail file. At least one transaction detail record is stored in the XML transaction detail file in response to a telecommunications transaction. The at least one transaction detail record includes transaction data corresponding to the telecommunications transaction.

In another aspect, the present invention includes a computer-readable medium having stored thereon a data structure for recording transactions in a telecommunications network. The data structure includes: an XML declaration field, the XML declaration field defining the data structure as an XML file; a server identification field, the server identification field including an IP address of a server generating the XML file; and a transaction detail section including at least one transaction detail record, the at least one transaction detail record being stored in the data structure in response to a telecommunications transaction, the at least one transaction detail record including transaction data corresponding to the telecommunications transaction.

In another aspect, the present invention includes a telecommunications network. The network includes at least one telecommunications apparatus configured to perform a telecommunications transaction. At least one SIP-server is coupled to the at least one telecommunications apparatus. The at least one SIP-server is configured to create an XML transaction detail file, process the telecommunications transaction, and store at least one transaction detail record in the XML transaction detail file. The at least one transaction detail record includes transaction data corresponding to the telecommunications transaction.

In another aspect, the present invention includes a computer-readable medium having stored thereon computer-executable instructions for performing a method for recording transactions in a telecommunications network. The method includes creating an XML transaction detail file. The XML transaction detail file is active for a predetermined period of time. At least one transaction detail record is stored in the XML transaction detail file in response to a telecommunications transaction. The at least one transaction detail record includes transaction data corresponding to the telecommunications transaction.

Additional features and advantages of the invention will be set forth in the detailed description which follows, and in part will be readily apparent to those skilled in the art from that description or recognized by practicing the invention as described herein, including the detailed description which follows, the claims, as well as the appended drawings.

It is to be understood that both the foregoing general description and the following detailed description are merely exemplary of the invention, and are intended to provide an overview or framework for understanding the nature and character of the invention as it is claimed. The accompanying drawings are included to provide a further understanding of the invention, and are incorporated in and constitute a part of this specification. The drawings illustrate various embodiments of the invention, and together with the description serve to explain the principles and operation of the invention.

Figure 1 is a block diagram of an IP telecommunications network in accordance with one embodiment of the present invention; and

Figure 2 is a chart showing a method for managing telecommunications transactions in accordance with another embodiment of the present invention.

Reference will now be made in detail to the present exemplary embodiments of the invention, examples of which are illustrated in the accompanying drawings. Wherever possible, the same reference numbers will be used throughout the drawings to refer to the same or like parts. An exemplary embodiment of the network of the present invention is shown in Figure 1, and is designated generally throughout by reference numeral 10.

In accordance with the invention, the present invention includes a method for recording transactions in a telecommunications network. The method includes creating an XML transaction detail file. At least one transaction detail record is stored in the XML transaction detail file in response to a telecommunications transaction. The at least one transaction detail record includes transaction data corresponding to the telecommunications transaction. The present invention provides a platform independent method for capturing transaction data and other information in a uniform manner. The system and method of the present invention is extensible, providing embedded information in generic, uniformly formatted transaction detail records that can be read by a receiving computer without needing a special proprietary interface.

As embodied herein, and depicted in Figure 1, a block diagram of an IP telecommunications network 10 in accordance with one embodiment of the present invention is disclosed. Telecommunications network 10 includes IP network 100, coupled to the Public Switched Telephone Network (PSTN) 20, Internet 30, a customer PBX 40, SIP phones 50, and SIP-clients 52. IP network 100 includes IP network backbone 120. Backbone 120 is coupled to a number of SIP elements that support voice services, including SIP proxy server 102, redirect server (RS) 104, SIP conferencing server 106, voice mail server 108, Operational Support Systems (OSS) 110, Dedicated Access Line (DAL) gateway 112, network gateway 114, INCP gateway 116, and enterprise gateway 118. Network backbone 120 is also directly coupled to Internet 30, SIP phones 50, and SIP Clients 52. Although the present invention is discussed with respect to the Session Initiation Protocol (SIP) and an Internet Protocol (IP)-based network, those of ordinary skill in the art will recognize that the present invention is equally applicable to other telecommunication networks and protocols.

IP network 100 may be of any suitable type, but there is shown by way of example a network having a layered architecture. The layered nature of the architecture provides

protocol separation and independence, whereby one protocol can be exchanged or modified without affecting the other higher layer or lower layer protocols. It is advantageous that the development of these protocols can occur concurrently and independently. The foundation of the layered architecture is the Internet Protocol (IP) layer. The IP layer provides a connectionless data delivery service that operates on a "best effort" basis; that is, no guarantees of packet delivery are made. A TCP (Transmission Control Protocol) layer is disposed above the IP layer. The TCP layer provides a connection-oriented protocol that ensures reliable delivery of the IP packets, in part, by performing sequencing functions. This sequencing function reorders any IP packets that arrive out of sequence. In another embodiment, the UDP (User Datagram Protocol) is employed instead of TCP. The User Datagram Protocol provides a connectionless service that utilizes the IP protocol to send a data unit, known as a datagram. Unlike TCP, UDP does not provide sequencing of packets. It relies on the higher layer protocols to sort the arriving packets. UDP is preferable over TCP when the data units are small, which saves processing time because of the minimal reassembly time. One of ordinary skill in the art will recognize that embodiments of the present invention can be practiced using either TCP or UDP, as well as other equivalent protocols.

A telephony application layer is disposed over the TCP layer. In one embodiment, the Session Initiation Protocol (SIP) is employed. SIP is an application-layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants. SIP is also a client-server protocol wherein servers respond to requests generated by clients. A detailed discussion of SIP and its call control services are described in IETF RFC 2543 and IETF Internet Draft "SIP Call Control Services", June 17, 1999. Those of ordinary skill in the art will recognize that application-layer protocols other than SIP may be employed, including the H.323 protocol.

Finally, the Session Description Protocol (SDP) is disposed above SIP in the layered architecture. SDP provides information about media streams in the multimedia sessions, permitting the recipients of the session description to participate in the session.

IP network backbone 120 may be of any suitable type, but there is shown by way of example a network that includes a nationwide high speed network that operates at 622MB/sec

(OC-12). Backbone 104 employs advanced packet switching technology commonly known as the Asynchronous Transfer Mode (ATM). Backbone 120 also utilizes a fiber-optic transmission technology referred to as the Synchronous Optical Network (SONET). The combination of ATM and SONET enables high speed, high capacity voice, data, and video signals to be combined and transmitted on demand. The high speed of backbone 120 is achieved by connecting Internet Protocol through the ATM switching matrix, and running this combination on the SONET network.

INCP 116 is an Intelligent Network Control Point that is accessed by RS 104 to obtain dial plan information for existing private network customers. INCP 116 is an additional database that may be queried by RS 104 to route specific private calls. INCP 116 may also be accessed by SPS 102.

PSTN 160 is, by way of example, a circuit switched network employing Signaling System No. 7 (SS7). Plain-Old-Telephone-Service (POTS) telephone 22 may be any suitable telephone set currently in use or on the market.

Enterprise gateway 118 may be of any suitable type. In the example depicted in Figure 1, enterprise gateway 118 is coupled to PBX 40. Those of ordinary skill in the art will recognize that enterprise gateway 118 may also couple IP network 100 to other enterprise networks, such as LANs and/or WANs. Referring back to Figure 1, PBX 40 includes trunks or lines for PBX phones 42. Enterprise gateway 118 provides the interface between packet switched IP network 100 and the signaling employed by PBX 40. For example, enterprise gateway 118 may use Integrated Digital Services Network (ISDN), Circuit Associated Signaling (CAS), or other PBX interfaces (e.g., European Telecommunications Standards Institute (ETSI) PRI, R2) to interface with PBX 40.

DAL gateway 112 is a system configured to support private traffic between IP locations and non-IP locations. DAL gateway 112 may be optionally employed in network 100.

Network gateway 114 is an SS7 (Signaling System 7)/C7-to-SIP Gateway. This provides users with the ability to place calls between locations within IP network 100 and locations within PSTN 20. For example, network gateway 114 is configured to provide

access to a voice switch (not shown), such as a Class 3 switch for domestic call processing, or a Class 5 switch for long-haul and/or international connections.

SIP phones 50, and SIP-client devices 52, may be of any suitable type provided that they conform to the standards promulgated in IETF 2543. SIP phones 50 have a 10-key dial pad similar to traditional phones. SIP URLs, which include PSTN 20 numbers, can be entered using the keypad or retrieved from a speed dial directory. To place a call, digits are entered using the dial pad. The entered digits are collected by the phone. When the "Dial" button is pressed, the call is initiated. SIP phones 50 ring similar to traditional phones when receiving an incoming call. SIP phones 50 may take the form of standalone devices – e.g., a SIP phone may be configured to function and appear like a Plain Old Telephone Service (POTS) telephone set. On the other hand, SIP client 52 is a software client that runs, for example, on a personal computer (PC) or laptop. From a signaling perspective, SIP-devices 50 and 52 are very similar, if not identical, in some cases.

At this point, the various SIP-servers disposed in network 100 will be described in more detail. Each type of server in network 100 has a critical role in recording and managing the various transactions supported by network 100.

Referring back to Figure 1, SIP-proxy server (SPS) conforms with SIP standards detailed per IETF RFC 2543. SPS 102 functions as both a server and a client for the purpose of making requests on behalf of other clients. SPS 102 may service a request directly or pass it on to another server. SPS 102 may also rewrite a message before forwarding it. SPS 102 is configured to create a transaction detail file in XML format to record transaction data processed by SPS 102. The transaction detail file is populated with transaction detail records(TDR). Each TDR records a transaction such as a SIP call-event (INVITE, ACK, BYE, CANCEL, OPTIONS, REGISTER, etc.), or any of the other events described above, such as errors, or timeouts. SPS 102 includes an XML processor module that is called by SPS 102 application software to create the XML transaction detail file. The XML processor module may also be called upon to read XML documents, e.g., to provide access to the content and structure of an XML file. The format of the XML transaction detail file is shown in detail in Table I.

Table 1 SIP Service Transaction Detail File Structure-Network Server

Fields	Syntax/Description	Comments
XML Declaration	**<?xml version="1.0"?>** Used to define the file as an XML document. This XML declaration is located on the first line of the file and is used to identify the file as an XML file.	Only one per file
XML Document Type	**<tdr>** Used to define the beginning of the TDR document.	Only one per file. Always after the XML declaration.
Network Server ID	**<nsid>**AA.BB.CC.DD:port**<nsid>** (e.g. 166.23.44.157) The Network Server IP address, where AA, BB, CC, DD can be digits 0 through 255 and port can be a number 1024 through 65535.	One per TDR file
Time	**<time>**Wed Mar 1 10:55:21 2000**</time>** A time field contains the time that this TDR field was opened in the following form: <time> Abbreviated Day <space+> Abbreviated month <space+> Day of month <space+> Hour <:> Minute <:> Second <space+> Year <time> Where: Abbreviated Day = Mon/Tue/Wed/Thu/Fri/Sat/Sun Abbreviated Month= Jan/Feb/Mar/Apr/May/Jun/Jul/Aug/Sep/Oct/Nov/Dec Day of month = 1 thru 31 Hour = 00 thru 23 Minute = 00 thru 59 Second = 00 thru 59 Year = 4 digit year	One time per TDR file
Tdr Record	**<tdr_record>** Contains that fields that are to be recorded for an event (message or other event).	Multiple Tdr Records may be in one TDR file (or none if time period expires with no events processed)

Correlation ID	**<corr_id>**text**</corr_id>** The correlation ID is used to uniquely identify all events (messages, features, etc.) that occur for a NS transaction (in the case of REGISTERs) or Session (in the case of INVITEs), starting with the reception of a SIP Request at the NS, and ending when the transaction (in the case of REGISTERs) or Session (in the case of INVITEs). This includes all messaging and feature processing that is done at the RS and the DAP for an NS transaction. The field will contain the value of the CallID that was present in the original request received at the NS.	One Correlation ID per Tdr Record
Time	**<time>**Wed Mar 1 10:55:21 2000**</time>** A time field contains the time that the event (message or other event occurred). The format is specified above.	One time per TDR record
Request	**<request>** A request contains the information about any SIP Request that is either sent or received by the NS. NOTE: Requests sent to the RS and AUS are not recorded.	Optional field - if present on per Tdr Record
Received-from	**<received-from>**AA.BB.CC.DD : port**</received-from>** (e.g. 166.23.44.157 : 5060) The IP Address that the Request message was received from if the Request was received at the NS, where AA, BB, CC, DD can be digits 0 through 255 and port can be a number 1024 through 65535.	One Received-from element per Request
Sent-to	**<sent-to>**AA.BB.CC.DD : port**</sent-to>** (e.g. 166.23.44.157 : 5060) The IP Address that the Request message was sent to if the Request was sent by the NS, where AA, BB, CC, DD can be digits 0 through 255 and port can be a number 1024 through 65535.	One Sent-to element per Request
Message	**<msg>**text**</msg>** This is a text field that will have all the fields associated with the particular request including, but not limited to: Request line, To, From, Call ID, Cseq, Via, Record-Route, Route, Expires, Max-Forwards, Proxy-Authorization, Proxy-Require, Correlation ID, and Contact. The NS will not record headers in the message that it does not use in processing, nor record the message body, for performance reasons.	One Message per Request

Request-End Delimiter	**</request>** Used to define the end of the request fields withing a TDR Record.	Optional field - if present, one per Tdr Record
Authentication	**<auth>** This field records information on the authentication that was performed on a request. If no authentication was performed.	Optional field - if present, one per Tdr Record
Authentication Indicator	**<auth_ind>**text**</auth_ind>** This field records the reason that authentication was performed. The values for this field are as follows: "Proxy-Authorization Header Present" "Authorization Header Present" "Authentication for Method is active in Auth Entity Table" "Authentication for Method is active in Global Table" "Next Hop (Route Header) is a protected entity" "AuthEntity has disabled record for source address"	One per Authentication
Authentication Server Indicator	**<aus_query>**text**</aus_query>** This field is recorded when a query was sent to the Aus. If no query was sent, this field would not be present. The values for this field are as follows: "Pending Challenge where nonce matches" "Auth Cache exists, but response does not match" "Auth Cache exists, but expiration time needs update"	Optional field - if present, one per Authentication

Authentication Result	**<auth_res>**text**</auth_res>** This is a text field that indicates the results of authentication if it was performed. If authentication was not performed, this field will not be present. The possible values are as follows: "Pass-Trusted Entity" "Pass-Authenticated" "Pass-Non Trusted User" "Fail-Entity Disabled" "Fail-System Failure" "Challenge-User information not found" "Challenge-Authentication information mismatch" "Challenge-No local Pending Challenge or Auth Cache Record" "Challenge-Expired Pending Challenge" "Challenge-Expired Auth Cache"	One per Authentication
Authentication - End Delimiter	**</auth>** Used to define the end of the Authentication fields withing a Tdr Record.	Optional field - if present, one per Tdr Record
Response	**<response>** A response contains the information about any SIP Response that is either sent or received by the NS. NOTE: Responses received from the RS or AUS are not recorded.	Optional field - if present one per Tdr record
Received-from	**<received-from>**AA.BB.CC.DD : port**</received-from>** (e.g. 166.23.44.157 : 5060) The IP Address that the Response message was received from if the Response was received at the NS, where AA, BB, CC, DD can be digits 0 through 255 and port can be a number 1024 through 65535.	One Received-from element per Response

Sent-to	**<sent-to>**AA.BB.CC.DD : port**</sent-to>** (e.g. 166.23.44.157 : 5060) The IP Address that the Response message was sent to if the Response was sent by the NS, where AA, BB, CC, DD can be digits 0 through 255 and port can be a number 1024 through 65535.	One Sent-to element per Response
Message	**<msg>**text**</msg>** This is a text field that will have all the fields associated with the particular Response including, but not limited to: Status Line, To, From, Call ID, Cseq, Via, Record-Route, Proxy-Authenticate, and Contact. The NS will not record headers in the message that it does not use in processing, nor record the message body, for performance reasons.	One Message per Response
Response - End Delimiter	**</response>** Used to define the end of the response fields withing a Tdr Record.	Optional field - if present, one per Tdr Record

Event	**<event>**text**</event>** Used to record events other than messages that caused actions at the NS during a transaction. The possible values are as follows: "TCP Connection to OUA lost" "TCP Connection to TUA" AA.BB.CC.DD:EE "lost" - where AA.BB.CC.DD is the IP address and EE is the port number of the TUA "TCP Connection to TUA" AA.BB.CC.DD:EE "could not be established" - where AA.BB.CC.DD is the IP address and EE is the port number of the TUA "RS Timeout" for AA.BB.CC.DD:EE - where AA.BB.CC.DD is the IP Address and EE is the port number of the RS that timed out "UA Invite Timeout for TUA" AA.BB.CC.DD:EE - where AA.BB.CC.DD is the IP Address and EE is the port number of the TUA "UA Non-Invite Timeout for TUA" AA.BB.CC.DD:EE - where AA.BB.CC.DD is the IP Address and EE is the port number of the TUA "ACK Timeout" "Txn Clear Timeout" "Expires Timeout" "AuS Timeout" for AA.BB.CC.DD"EE - where AA.BB.CC.DD is the IP Address and EE is the port number of the AUS that timed out "Ring Timeout for TUA" AA.BB.CC.DD:EE - where AA.BB.CC.DD is the IP Address and EE is the port number of the TUA "487 Timeout for TUA" AA.BB.CC.DD:EE - where AA.BB.CC.DD is the IP Address and EE is the port number of the TUA	
TDR Data - End Delimiter	**</tdr_record>** Used to define the end of the TDR Record fields within a TDR File	Multiple Tdr Records may be in one TDR file (or none if timer period expires with no events processed)

XML Document Type end	</tdr> Used to define the end of the TDR Document.	Only one per file. Always at the end of the file.

Voice mail server (VMS) 108 is a SIP-server that provides voice mail services. Users of the IP network 100 are provided with the capability to integrate voicemail services based upon SIP. Calls are routed to the voice mail system 108 by SPS 102 and RS 104 for certain calls, such as those that indicate a Busy or Ring No Answer condition. Calls to voice mail can also occur as a Find-Me/Follow-Me termination option, or as an Unconditional Call Forward option selected by the user. Calls by the user to log in and retrieve messages are routed to VMS 108 as a SIP endpoint. A voice mail address can be entered for any destination address in RS 104. For instance, the Call Forwarding Unconditional address or Find-Me address, etc., can be the SIP URL of a voice mail account. SIP enabled VMS 108 supports all alphanumeric SIP URLs, Headers, Request, Methods and Status codes (e.g., per IETF RFC 2543). VMS 108 supports SUBSCRIBE, NOTIFY, and Message Waiting Indicator (MWI) messages.

VMS 108 may restrict access to the system through a variety of ways. Access may be secured through private access code. The access code may be supplied in the SIP INVITE message or through DTMF. VMS 108 may reject messages based on the IP address of the originating server. In other words, if the message is coming from a server that is not trusted, then VMS 108 may reject the message.

VMS 108 is also configured to create a transaction detail file in XML format to thereby record transaction data corresponding to all network transactions processed by VMS 108. Because the format of the VMS XML transaction detail file is very similar to the SPS 102 XML transaction detail file, it will not be repeated here.

SIP conferencing server (SCS) 106 is a centralized SIP-conference server configured to provide audio conferencing capabilities. SCS 106 support G.711 (RTP/AVT 0), as well as other codecs. SCS 106 may specify two modes of operation. Under a Reserved mode, the users are required to reserve a bridge ahead of time. An Instant Conferencing mode refers to the ability to set-up a conference as needed without any need for advance reservation, allowing ad-hoc set-up of conferences as well

permitting client based conferences to migrate to a bridge. Conference access is secured through an access code. Participants joining the bridge can send their access code via the SIP Invite message. POTS telephone users can enter through DTMF depending on the support for DTMF at the gateway. An audible tone may be played to announce each participant as they join the bridge. The system supports a coordinator /operator initiated conference, wherein the operator dials-out to each of the conference participants and brings them into the conference. The conference operator can enter and announce the name of the participants into the conference. The conference coordinator can notify the participants of the time and date for the call. The operators may be able to put parties On and Off Hold. Music On Hold is supported, whereby the parties on Hold are provided with music.

SCS 106 also permits private conferencing (i.e., sub-conferencing), wherein designated conference callers may confer privately within a conference call and then be returned to the main call. Calls from PSTN 20 may be forwarded to SCS 106 by network gateway 114. From the perspective of SCS 106, a SIP originated call is not processed differently than a non-SIP call because network gateway 114 is able to translate the called number to the conference URL. However, SCS 106 is able to validate the caller by prompting for passwords and validating the password entered as DTMF digits. As an alternate to password collection through DTMF, SCS 106 may support authentication using SIP. In this scenario, the SIP INVITE message carries additional user parameters, such as username/password combination that may be used by SCS 106 to validate the user. Further, conferencing system 106 supports web based provisioning by the users. SCS 106 interfaces with the OSS 110 for provisioning, alarming and reporting. The provisioning and reporting interface of the OSS 110 assists with a number of conferencing functionalities, such as the capability to Setup, Modify and Delete conferences. The administrator or moderator of the conference is able to specify the number of attendees to a conference, as well as specify duration of the conference, date and time-by-time zone, and name of reserved conference.

SCS 106 is configured to create a transaction detail file in XML format to thereby record transaction data corresponding to all the above described transactions processed by conferencing server 106. Because the format of the SCS 106 XML

transaction detail file is similar to the SPS 102 XML transaction detail file, it will not be repeated.

RS 104 is a SIP redirect server that conforms with SIP standards detailed per IETF RFC 2543. RS 104 accepts SIP messages, maps the address into one or more new addresses, and returns these addresses to the client, which could be SPS 102. RS 104 does not initiate its own SIP requests, and RS 104 does not accept calls. RS 104 is essentially, a location server wherein information about possible terminating locations can be obtained. RS 104 also serves as a repository for end user information enabling address validation, feature status, and real-time subscriber feature configuration. RS 104 may also be used to store configuration information.

RS 104 is also configured to create a transaction detail file in XML format to thereby record transaction data corresponding to all SIP transactions, timeouts and errors processed by RS 104. The transaction detail file includes transaction detail records used to record network transactions processed by RS 104. RS 104 includes an XML processor module that is called by RS 104 application software module to create the XML transaction detail file. The XML processor module may also be called to read an XML file. Because RS 104 has a different function in the management of network 100, its XML transaction detail file is substantially different than the SPS XML transaction detail file. The format of the RS XML transaction detail file is shown in detail in Table II.

Table II SIP Services Transaction Detail File Structure – Redirect Server

Fields	Syntax/Description	Comments
XML Declaration	**<?xml version="1.0"?>** Used to define the file as an XML document. This XML declaration is located on the first line of the file and is sued to identify the file as an XML file.	Only one per file
XML Document Type	**<tdr>** Used to define the beginning of the TDR document	Only one per file. Always after the XML declaration.

Redirect Server ID	<**rsid**>AA.BB.CC.DD:port</**rsid**> (e.g. 166.23.44.157:5060) The Redirect Server IP address where AA, BB, CC, DD can be digits 0 through 255 and port can be a number 1024 through 65535.	Only one per file
Time	<**time**>Wed Mar 1 10:55:21 2000</**time**> The time field contains the time that this TDR file was opened in the following form: <time>AbbreviatedDay<space+>Abbreviated Month <space+>Day of Month<space+>Hour<:>Minute <:>Second<space+>Year</time> Where: Abbreviated Day=Mon/Tues/Wed/Thu/Fri/Sat/Sun Abbreviated Month+Jan/Feb/Mar/Apr/May/Jun/Jul/Aug/Sep/Oct/Nov/Dec Day of Month= 1 thru 31 Hour = 00 thru 23 Minute = 00 thru 59 Second = 00 thru 59 Year = 4 digit year	One time per TDR file
Tdr Record	<**tdr_record**> Contains the fields that are to be recorded for a transaction within the RS.	Multiple Tdr Records may be in one TDR file (or none if time period expires with no transaction processed)
Correlation ID	<**corr_id**>text</**corr_id**> The correlation ID is used to uniquely identify all events (messages, features, etc.) that occur for a NS transaction (in the case of REGISTERS) or Session (in the case of INVITE), starting with the reception of a SIP Request at the NS, and ending when the transaction(in the case of REGSTER) or session (in the case of INVITE). This includes all messaging and feature processing that is done at the RS and the DAP for an NS transaction. The filed will contain the value of the Call ID that was present in the original request received at the NS.	One Correlation ID per Tdr Record
Request	<**request**> A request contains the information about the initial SIP Request message that is received at the RS from the NS. This request is what triggers the transaction processing at the RS.	One Request per Tdr Record

Received-from	**<received-from>**AA.BB.CC.DD:port**</received-from>** (e.g. 166.23.44.157 : 5060) The IP address and port that the request was received from - where AA, BB, CC, DD can be digits 0 through 255 and port can be a number 1024 through 65535.	One Received-from element per Request
Time	**<time>**Wed Mar 1 10:55:21 2000**</time>** The time field contains the time that the request was received. Same format as described above	One time per Request
Message	**<msg>**text**</msg>** The contents of the SIP message that was received from the NS. This will have all the fields associated with the particular request including, but not limited to: Request URI, To, From, Call ID, Cseq, Via, Proxy-Authorization, Expires and Contact.	One Message per Request
Request-End Delimiter	**</request>** Used to define the end to the request fields within a TDR Record	One Request per Tdr Record
Feature	**<feature>** This field is used to record the data concerning the features that were executed at the RS during the transaction. If no features were executed, this field will not be included in the Tdr Record.	Optional field-if present, one per Tdr Record
Recursive Call Routing	**<rec_call_routing>** This field is sued to record when the recursive call routing feature is invoked. It is invoked when a non-final address that was passed back to the NS results in a new query to the RS with previous context information appended to the Request URI of the new query. This will only be present when the "final-no" or "final=egwy" URL parameter is received on the RequestURI.	Optional field-if present, one per feature.

Terminating Nature of Address	**<term_noa>**text**</term_noa>** This field is used to record the terminating Nature of Address which appears as a url parameter in the RequestURI The possible values for this are the following: "Private" "Local" "E.164" "IP Address"	One per Recursive Call Routing
Originating Dial Plan ID	**<orig_dpid>**text**</orig_dpid>** This field is used to record the originating dial plan Id which appears as a url parameter in the RequestURl	One per Recursive Call Routing
Location Name	**<locname>**text**<locname>** This field is used to record the location name which appears as a url parameter in the RequestURI	One per Recursive Call Routing
Recursive Call Routing - End Delimiter	**</rec_call_routing>** Used to define the end of the Recursive Call Routing fields within a TDR Record	Optional field-if present, one per Feature
Originating Call Validation	**<orig_val>** This field is used to record the data gathered during the validation of the originator of the call. This data includes the nature of address, the dial plan ID, the prefix plan ID, the location name, and an indicator if the subscriber was not authorized. If there is no record for the originator of the call, portions of this data (dial plan ID, prefix plan ID, location name) will be retrieved for the gateway (2nd most Via header in the request). If there is no record for the gateway, portions of this data (dial plan ID, prefix plan ID, location name) will be retrieved from the global information table.	Optional field - if present, one per Feature

Originating Nature of Address	**<orig_noa>**text**</orig_noa>** This field is used to record the nature of address for the originating subscriber based on the user portion of the From Header in the Request message. The possible values for this are the following: "Private" "E.164" "IP Address"	One per Originating Call Validation
Originating Dial Plan ID	**<orig_dpid>**text**</orig_dpid>** This field is used to record the dial plan ID of the originating subscriber. This is retrieved from the subscriberID or authprof record(depending on the present of Proxy-Authorization Header) if present. If not present, it is retrieved from the rsgwyinfo record if present. If not present, it is retrieved from the rsglobal record.	One per Originating Call Validation
Prefix Plan ID	**<ppid>**text**</ppid>** This field is used to record the prefix plan ID for the originating subscriber. This is retrieved from the subscriber record if present. If not present, it is retrieved from the rsgwyinfo record if present. If not present, it is retrieved from the served terminating hosts record.	One per Originating Call Validation
Location Name	**<locname>**text**</locname>** This field is used to record the location name for the originating subscriber. This is retrieved from the subscriber record if present. If not present, it is retrieved from the rsgwyinfo record if present. If not present, it is retrieved from the rsglobal record.	One per Originating Call Validation
Not Authorized Indicator	**<not_auth>**text**</not_auth>** This field is present if the originating subscriber if found to be unauthorized to make any phone calls. Its value if present is "Not Authorized."	Optional field-if present, one per Originating Call Validation
Not Trusted Indicator	**<not_trusted>**text**</not_trusted>** This field is present if the originating subscriber is not trusted. An originating subscriber is considered not trusted when it is a non-authenticated user that does not come through a trusted entity. The fields value if present is "Not Trusted."	Optional field-if present, one per Originating Call Validation

Originating Call Validation-End Delimiter	**</orig_val>** Used to define the end of the originating call validation fields within a TDR Record	Optional field-if present, one per Feature
Not Trusted Terminating Call	**<non_trusted_term>**text**</non_trusted_term>** This field is present if a call is originated from a non-trusted user. It is used to record whether or not a non-trusted call was allowed, and if so, whether the terminating subscriber's profile allowed the call, or whether the served hosts dial plan allowed the call. The possible values for this field are the following: "Non-Trusted User - Call Not Allowed" "Non-Trusted User - Call Allowed by Profile" "Non-Trusted User - Call Allowed by dial plan"	Optional field - if present, one per Feature
Terminating Call Validation	**<term_val>** This field is used to record the data gathered during the validation of the subscriber that is being called. This data includes the nature of address, the dial plan ID, and an indicator if the subscriber was not authorized. If there is no record for the subscriber that is being called, there will be no value for the dial plan ID or the not authorized indicator.	One per Feature
Terminating Nature of Address	**<term_noa>**text**</term_noa>** This field is sued to record the nature of address for the terminating subscriber based on the user portion of the Request URI. The possible values at this point in processing are: "IP Address" "E.164" "Other." If the values is "Other," it implies that the Flexible Dialing Plan Feature will need to be involked to determine the noa for further feature processing.	One per Terminating Call Validation

Terminating Dial Plan ID	**<term_dpid>**text**</term_dpid>** This field is used to record the dial plan ID of the terminating subscriber. This is retrieved from the subscriberID record if present. If not present this field will have the same value as the originating dpid.	One per Terminating Call Validation
Not Authorized Indicator	**<not_auth>**text**</not_auth>** This field is present if the terminating subscriber if found to be unauthorized to receive phone calls. Its value if present is "not Authorized."	Optional Field-if present, one per Terminating Call
Profile Found Indicator	**<profile_found>**text**</profile_found>** This field is used to record whether or not a profile record was found for the terminating subscriber. The possible values for this field are the following: "Profile found" "Profile not found"	One per Terminating Call Validation
Terminating Call Validation-End Delimiter	**</term_val>** Used to define the end of the terminating call validation fields within a TDR Record	Optional field -if present, one per Feature
Originating Call Screening	**<ocs>** This field is sued to record information when the originating call screening (aka Call Blocking) feature is executed at the RS. If the feature if not executed, this field will not be present.	Optional field-if present, one per Feature
Originating Call Screening List Number	**<list>**text**</list>** This field records the list number that was used to determine whether the call should be blocked or allowed. It was retrieved from the subscriber record if present. If no subscriber record was present, it was retrieved from the rsgwyinfo table. If no rsgwyinfo record was present, it was retrieved from the rsglobal table.	One per Originating Call Screening
Originating Call Screening Result	**<result>**text**</result>** This field records the result of the Originating Call Screening Feature. The possible values are "Blocked" and "Allowed."	One per Originating Call Screening
Originating Call Screening-End Delimiter	**</ocs>** Used to define the end of the Originating Call Screening fields within a TDR Record	Optional field-if present, one per Feature

Terminating Call Screening	**<tcs>** This field is used to record information when the terminating call screening feature is executed at the RS. IF the feature is not executed, this field will not be present.	Optional field-if present, one per Feature
Terminating Call Screening List Number	**<list>text</list>** This field records the list number that was used to determine whether the call should be screened or allowed. It was retrieved from the subscriber record, which must be present for this feature to be executed.	One per Terminating Call Screening
Terminating Call Screening Result	**<result>**text**</result>** This field records the result of the Terminating Call Screening Feature. The possible values are "Screened" and "Allowed."	One per Terminating Call Screening
Terminating Call Screening - End Delimiter	**</tcs>** Used to define the end of the Terminating Call Screening fields within a TDR Record	Optional field-if present, one per Feature
Call Forwarding Unconditional	**<cfu>** This field is used to record information when the Call Forwarding Unconditional feature is executed at the RS. If the feature is not executed, this field will not be present.	Optional field-if present, one per Feature
Call Forwarding Unconditional Address	**<cfu_addr>**text**</cfu_addr>** This field record the forwarding address that will be used to redirect the call.	Optional field-if present, one per Feature
Call Forwarding Unconditional Nature of Address	**<cfu_noa>**text**</cfu_noa>** This field records the nature of address that corresponds to the forwarding address that is used to redirect the call. The possible values for this field are the following: "Private" "E.164" "Local" "IP Address"	One per Call Forwarding Unconditional

Call Forwarding Unconditional-End Delimiter	**</cfu>** Used to define the end of the Call Forwarding Unconditional fields within a TDR Record	Optional field-if present, one per Feature
Find Me	**<findme>** This field is used to record information when the Find Me Feature is executed at the RS. If the feature is not executed, this field will not be present.	Optional field - if present, one per Feature
Find Me Error	**<error>**text**</error>** This field is used to record an error in the find me provisioning. The possible values for this field are: "No Find Me Record found for Subscriber,: "No Find Me Record Found for This Time," "No find Me Device Sequence List record Found," "No Active Devices were Found in the Find Me List"	Optional Field - if present, one per Find Me
Find Me Category	**<fm_cat>**text**</fm_cat>** This field records the category that applies to the originating subscriber. The possible values for this field are 0-3, where 0 is the default category.	Optional Field-if present one per Find Me
Find Me Device Number	**<fm_dev_num>**text**</fm_dev_num>** This fields records the device number a device that was found in the applicable find me record.	Optional field-there can be one or more instances of this field within the Find Me field
Fine Me Address	**<fm_addr>**text**<fm_addr>** This field records the address of the find me device that will be used to redirect the call.	Optional field-there can be one or more instances of this field within the Find Me field.
Find me Nature of Address	**<fm_noa>**text**</fm_noa>** This field records the nature of address that corresponds to the find me address that will be used to redirect the call. The possible values for this field are the following: "Private" "E.164" "Local" "IP Address"	Optional field- there can be one or more instances of this field within the Find Me field

Find Me- End Delimiter	**</findme>** Used to define the end of the Find Me fields within a TDR Record	Optional field - if present, one per Feature
Registered Address List	**<reglist>** This field is used to record information when the call is redirected to the list of Registered Addresses (not via the Find ME Feature). If the call is not redirected to the registered addresses, this field will not be present.	Optional field- if present, one per Feature
Registered Address	**<reg_addr>**text**</reg_addr>** This field records a registered address that will be used to redirect the call.	One or more instances of this field within the Registered Address List
Registered Nature of Address	**<reg_noa>**text**</reg_noa>** This field records the nature of address that corresponds to the registered address that will be used to redirect the call. The possible values for this field are the following: "Private" "E.164" "IP Address"	One or more instances of this field within the Registered Address List
Registered Address List-End Delimiter	**</reglist>** Used to define the end of the Registered Address List fields within a TDR Record	Optional field-if present, one per Feature
Default Number	**<defnum>** This field is used to record information when the call is redirected to the default address in the subscriber record. If the call is not redirected to the default address, this field will not be present.	Optional field - if present, one per Feature
Default Address	**<def_addr>**text**</def_addr>** This field records the default address that will be used to redirect the call.	One per Default Number

Default Nature of Address	**<def_noa>**text**</def_noa>** This field records the nature of address that corresponds to the default address that will be used to redirect the call. The possible values for this field are the following: "Private" "E.164" "Local" "IP Address"	One per Default Number
Default Number-End Delimiter	**</defnum>** Used to define the end of the Default Number List fields within a TDR Record	Optional field - if present, one per Feature
Call Forwarding Busy	**<cfb>** This field is used to record information when the Call Forwarding Busy feature is executed at the RS. If the feature is not executed, this field will not be present.	Optional field-if present, one per Feature
Call Forwarding Busy Address	**<cfb_addr>**text**</cfb_addr>** This field record the forwarding address that will be used to redirect the call.	One per Call Forwarding Busy
Call Forwarding Busy Nature of Address	**<cfb_noa>**text**</cfb_noa>** This field records the nature of address that corresponds to the forwarding address that is used to redirect the call. The possible values for this field are the following: "Private" "E.164" "Local" "IP Address"	One per Call Forwarding Busy
Call Forwarding Busy-End Delimiter	**</cfb>** Used to define the end of the Call Forwarding Busy fields within a TDR Record	Optional field-if present, one per Feature

Call Forwarding No Answer	**<cfna>** This field is used to record information when the Call Forwarding No Answer feature is executed at the RS. If the feature is not executed, this field will not be present.	Optional field-if present, one per Feature
Call Forwarding No Answer Address	**<cfna_addr>**text**</cfna_addr>** This field records the forwarding address that will be used to redirect the call	One per Call Forwarding No Answer
Call Forwarding No Answer Nature of Address	**<cfna_noa>**text**</cfna_addr>** This field records the nature of address that corresponds to the forwarding address that is used to redirect the call. The possible values for this field are the following: "Private" "E.164" "Local" "IP Address"	One per Call Forwarding No Answer
Call Forwarding No Answer-End Delimiter	**</cfna>** Used to define the e4nd of the Call Forwarding No Answer fields with a TDR Record	Optional field-if present one per Feature
Feature-End Delimiter	**</feature>** Used to define the end of the Feature fields within a TDR Record	Optional field-if present, one per TDR Record
DAP Query	**<dap_request>** This field is used to record information about a query to the DAP concerning a particular routing address	Optional field-if present, one or more instances per TDR Record
DAP Request	**<dap_request>** This field is used to record the request message that was sent to the DAP.	One per DAP Query

Sent-to	**<sent-to>AA.BB.CC.DD. : port<sent-to>** (E.g. 166.23.44.157 : 5060) The IP address and port of the DAP that the request was sent to- where AA, BB, CC, DD can be digits 0 through 255 and port can be a number 1024 through 65535.	One per DAP Request
Time	**<time>**Wed Mar 1 10:55:21 2000**</time>** The time field contains the time that DAP Request was sent in the format specified above.	One time per DAP Request
Message Type	**<msg_type>**text**</mst_type>** This field records the type of message sent to the DAP. Valid values are as follows: "DAL Request"	One per DAP Request
NCS Transaction ID	**<ncs_trans_id>**text**</ncs_trans_id>** This field records the NCS transaction ID that was internally generated at the RS and sent to the DAP	One per DAP Request
Originating Switch ID	**<o_sid>**text**</o_sid>** This field records the Originating Switch ID from the rsoriglocinfo table that is sent from the RS to the DAP	One per DAP Request
Originating Trunk Group Number	**<o_tgn>**text**</o_tgn>** This field records the Originating Trunk Group Number from the rsoriglocinfo table that is sent from the RS to the DAP	One per DAP Request
Address Digits	**<addr_digs>**text**</addr_digs>** This field records the address Digits from the routing	One per Dap Request
DAP Request-End Delimiter	**</dap_request>** Used to define the end of the DAP Request fields within a TDR Record	One per DAP Request
Event	**<event>** This field is used to record the event that occurred that caused the RS to not receive a response from the RS.	Optional Field - if present, One per DAP Query

Time	**<time>**Wed Mar 1 10:55:21 2000**</time>** The time field contains the time that the event occurred in the format specified above.	One time per Event
Event Type	**<type>**text**<type>** This field is used to record the type of event. The possible values are: "Timeout" "TCP Connection Lost"	One per Event
Event-End Delimiter	**</event>** Used to define the end of the Event fields within a TDR Record	Optional Field -if present, one per DAP Query
DAP Response	**<dap_response>** This field is used to record the response message that was received from the DAP. If no response was received, this field will not be present. NOTE: Since the interface to the DAP is TCP, there is no reason to record the address and port number that the response is received from.	Optional Field -if present, one per DAP Query
Time	**<time>**Wed Mar 1 10:55:21 2000**</time>** The time field contains the time that the response was received in the format specified above.	One per DAP Response
Message Type	**<msg_type>**text**</msg_type>** This field records the type of message received from the DAP. Valid values are as follows: "Routing Response" "Failure Response"	One per Dap Response
NCS	**<ncs_trans_id>**text**</ncs_trans_id>**	One per DAP
Transaction ID	This field records the NCS transaction ID that was received in the DAP Response	Response
Terminating Switch ID	**<t_sid>**text**</t_sid>** This field records the Terminating Switch ID thar was received in the DAP Response.	Optional field -if present, One per DAP Response
Terminating Trunk Group Number	**<t_tgn>**text**<t_tgn>** This field records the Terminating Trunk Group Number received in the DAP Response.	Optional field -if present, One per DAP Response
Action Code	**<act_code>**text**</act_code>** This field records the action code that was received in the DAP Response.	One per DAP Response

Subsequent Address	**<sub_addr>**text**</sub_addr>** This field records the subsequent address that was received in the DAP Response.	Optional field -if present, One per DAP Response
Ported Number	**<ported_num>**text**</ported_num>** This field records the ported number that was received in the DAP Response.	Optional field -if present, One per DAP Response
DAP Response -End Delimiter	**</dap_response>** Used to define the end of the DAP Response fields within a TDR Record	Optional Field -if present, One per DAP Query
DAP Query - End Delimiter	**</dap_query>** Used to define the end of the DAP Query fields within a TDR Record	Optional field -if persent, one or more instances per Tdr Record
Response	**<response>** This field is used to record the final SIP Response that is returned to the NS from the RS. This field is present for all Tdr Records with the exception of the GWSTAT message. That particular message does not result in a response	Optional field -if present, one per Tdr Record
Sent-to	**<sent-to>**AA.BB.CC.DD :port**<sent-to>** (e.g. 166.23.44.157 : 5060) The IP address and port of the NS that the response was sent to - where AA, BB, CC, DD can be digits 0 through 255 and port can be a number 1024 through 65535.	One per Response
Time	**<time>**Wed Mar 1 10:55:21 2000**</time>** The time field contains the time that the Response was sent in the format specified above.	One time per Response
Message	**<msg>**text**</msg>** The contents of the SIP message that was sent to the NS. This will have all the fields associated with the particular response including, but not limited to: Status Line, To, From, Call ID, CSeq, Via, Expires, Feature, Contact, and ReqCtl.	One Message per Response
Response-End Delimiter	**</response>** Used to define the end of the Response fields within a TDR Record	Optional Field -if present, one per Tdr Record

Tdr Record-End Delimiter	**</tdr_record>** Used to define the end of the Tdr Record fields within a TDR File	Multiple Tdr Records may be in one TDR file (or none if time period expires with no events processed)
XML Document Type end	**</tdr>** Used to define the end of the TDR file.	One per TDR file. Always at the end of the file

Referring back to Figure 1, OSS 110 is also a critical system for managing network 100. OSS 110 supports the establishment, provisioning, data collection, and billing of the services of the system 100. OSS 110 is a distributed computing system that includes customer management, account management, billing, network facilities provisioning, and network data collection functionality. All of the XML transaction detail files generated by the above described servers SPS 102, RS 104, SCS 106, and VMS 108, are transmitted to OSS 110 using the XML transaction detail files described above. The XML transaction detail files are used by OSS 110 for a variety of network functions including, but not limited to, network management, billing, and record keeping. Thus, the present invention provides a platform independent method for capturing transaction data in a uniform manner. The present invention is extensible, providing embedded information that will enable any receiving computer to read the generic, uniformly formatted XML files without needing any proprietary interface.

In one embodiment, the OSS computing system is based on technology provided by SUN Microsystems, the databases employed by the computing system are based on technology provided by ORACLE. OSS 110 provides and controls access to customer accounts. Users may utilize a web page to monitor service, login to their account, and manage certain elements permitted by user profiles. The account management system allows network personnel to establish, maintain, or deactivate customer accounts. In one embodiment, customer information is viewed via a web interface. The billing system processes customer event records, the customer pricing plan data, adjustments, taxation and other data in the preparation of customer invoices. The network facilities provisioning system provides the information required by network engineers to ensure that the appropriate hardware and software is in place to provide service. This may involve the creation of a customer profile, and the reconfiguration of SPS 102, RS 104,

or other network elements. Network provisioning may also require the placement of hardware plug-in devices used in backbone 120.

A process management/work flow system serves as the core of OSS 110. The software is a Common Object Request Broker Architecture (CORBA) based publish-and-subscribe messaging middleware that provides graphical process automation, data transformation, event management and flexible connectors to transact with interfacing applications. This middleware architecture software fulfills the function of integrating all OSS 110 components and may provide hooks to non-OSS components using designated standard interfaces.

As embodied herein, and depicted in Figure 2, a chart showing a method for recording telecommunications transaction data in accordance with the present invention is disclosed. The method described herein is equally applicable to SPS 102, RS 104, SCS 106, and VMS 108. In step 200, the XML detail file is created. If the file is being created in RS 104, XML detail file will have the form depicted in Table II, otherwise, the XML detail file will be of the form, or similar to the form, shown in Table I. Each XML file is active for a predetermined period of time. Thus, once the XML file is created, the server initializes a timer to track elapsed time. For example, OSS 110 may direct each server to keep each XML detail file active for one day, or one hour, as the case may be. In step 204, if a transaction is detected, the server analyzes the transaction and performs an appropriate action. For example, SPS 102 (See Figure 1) may receive an INVITE message from SIP-phone 50, requesting a session with a user at POTS telephone 22. In processing the INVITE message, SPS 102 may perform and coordinate a plurality of transactions required to set up the session between SIP phone 50 and POTS telephone 22. In doing so, SPS 102 creates a transaction detail record (TDR) for each transaction in the call set-up process. In step 210, the TDRs are written into the XML file. On the other hand, if there are no transactions generated in the predetermined time period, no records are written into the file. In this case, only the header information in the XML file is transmitted to OSS 110.

Once the timer has elapsed, the server transmits the XML file to OSS 110. After the XML file is transmitted, a new file is created and the process repeats. If the timer has not elapsed, the server waits for additional transactions to process. In step

216, the server may suspend operations for any number of reasons. For example, if the server requires maintenance and is off-line, it is unnecessary to continue to monitor and record network transactions.

Those of ordinary skill in the art will recognize that the use of XML transaction detail files in accordance with the present invention can be employed for any events occurring within network 10. Calls placed between all or any combinations of SIP-phones, enterprise gateways, network gateways, DAL gateways, INCP gateways, SIP-voicemail servers, and SIP conferencing servers may employ the present invention. Those of ordinary skill in the art will also recognize that the present invention can be employed using any suitable type of transport network. Further, the present invention is applicable to any type of session that may be established including, but not limited to, telephony, video, audio, instant messaging, and etc. It is also contemplated that the present invention may be employed for billing, monitoring, management, or for any of a wide variety of services performed by the network.

It will be apparent to those skilled in the art that various modifications and variations can be made to the present invention without departing from the spirit and scope of the invention. Thus, it is intended that the present invention cover the modifications and variations of this invention provided they come within the scope of the appended claims and their equivalents.

What is claimed is:

1. A method for recording transactions in a telecommunications network, the method comprising:

creating an XML transaction detail file; and

storing at least one transaction detail record in the XML transaction detail file in response to a telecommunications transaction, the at least one transaction detail record including transaction data corresponding to the telecommunications transaction.

2. The method of claim 1, wherein the XML transaction detail file is active for a predetermined period of time.

3. The method of claim 2, further comprising:

closing the XML transaction detail file when the predetermined period of time expires;

creating a second XML transaction detail file, the second XML transaction detail file being active for a second predetermined period of time; and

repeating the steps of closing and creating for a number of times as directed by a network management system.

4. The method of claim 1, wherein the XML transaction detail file comprises:

an XML declaration field, the XML declaration field defining the data structure as an XML file;

a server identification field, the server identification field including an IP address of a server generating the XML file; and

a transaction detail section including at least one transaction detail record, the at least one transaction detail record being stored in the data structure in response to a telecommunications transaction, the at least one transaction detail record including transaction data corresponding to the telecommunications transaction.

5. The method of claim 1, wherein the telecommunications transaction is conducted between a telecommunications apparatus and a SIP-server.

6. The method of claim 5, wherein the telecommunications apparatus includes a telephone.

7. The method of claim 5, wherein the telecommunications apparatus includes a SIP-device.

8. The method of claim 7, wherein the SIP-device is a telephone.

9. The method of claim 7, wherein the SIP-device is a computer.

10. The method of claim 5, wherein the telecommunications apparatus includes a gateway.

11. The method of claim 5, wherein the SIP-server is a proxy server.

12. The method of claim 5, wherein the SIP-server is a conferencing server.

13. The method of claim 5, wherein the SIP-server is a voice mail server.

14. The method of claim 5, wherein the telecommunications apparatus is a proxy server and the SIP-server is a redirect server.

15. The method of claim 1, wherein the telecommunications transaction includes a SIP-INVITE message.

16. The method of claim 1, wherein the telecommunications transaction includes a SIP-ACK message.

17. The method of claim 1, wherein the telecommunications transaction includes a SIP-OPTIONS message.

18. The method of claim 1, wherein the telecommunications transaction includes a SIP-BYE message.

19. The method of claim 1, wherein the telecommunications transaction includes a SIP-CANCEL message.

20. The method of claim 1, wherein the telecommunications transaction includes a SIP-REGISTER message.

21. The method of claim 1, wherein the telecommunications transaction includes a billing transaction.

22. The method of claim 1, wherein the telecommunications transaction includes a monitoring transaction.

23. The method of claim 1, wherein the telecommunications transaction includes a performance measurement transaction.

24. A computer-readable medium having stored thereon a data structure for recording transactions in a telecommunications network, the data structure comprising:

an XML declaration field, the XML declaration field defining the data structure as an XML file;

a server identification field, the server identification field including an IP address of a server generating the XML file; and

a transaction detail section including at least one transaction detail record, the at least one transaction detail record being stored in the data structure in response to a telecommunications transaction, the at least one transaction detail record including transaction data corresponding to the telecommunications transaction.

25. The data structure of claim 24, further comprising a time/date field for indicating the time and date that the XML file was opened.

26. The data structure of claim 24, wherein the data structure is an XML transaction detail file created by an IP network server.

27. The data structure of claim 26, wherein the XML transaction detail file is active for a predetermined period of time.

28. The data structure of claim 26, wherein the server identification field includes the IP address of a proxy server.

29. The data structure of claim 26, wherein the server identification field includes the IP address of a conferencing server.

30. The data structure of claim 26, wherein the server identification field includes the IP address of a voice mail server.

31. The data structure of claim 26, wherein the at least one transaction detail record includes a plurality of transaction detail records.

32. The data structure of claim 31, wherein the at least one transaction detail record includes a correlation identification field for correlating each transaction detail record of the plurality of transaction detail records corresponding to a particular transaction.

33. The data structure of claim 26, wherein each transaction detail record includes a time/date field indicating when the transaction detail record was made.

34. The data structure of claim 26, wherein each transaction detail record includes a request section including information about any SIP request that is either sent or received by the IP network server.

35. The data structure of claim 34, wherein the request section includes a received-from field indicating the IP address of the sender of the SIP-request.

36. The data structure of claim 34, wherein the request section includes a sent-to field indicating the IP address of the addressee of the SIP-request.

37. The data structure of claim 34, wherein the request section includes a message field having all data associated with the request.

38. The data structure of claim 26, wherein each transaction detail record includes an authentication section, the authentication section indicates whether an authentication was performed, why the authentication was performed, and the result of the authentication.

39. The data structure of claim 26, wherein each transaction detail record includes a response section including information about any SIP response that is either sent or received by the IP network server.

40. The data structure of claim 39, wherein the response section includes a received-from field indicating the IP address of the sender of the SIP- response.

41. The data structure of claim 39, wherein the response section includes a sent-to field indicating the IP address of the addressee of the SIP- response.

42. The data structure of claim 39, wherein the response section includes a message field having all data associated with the response.

43. The data structure of claim 26, wherein each transaction detail record includes an event section for recording at least one event, other than messages, that occurred at the IP network server.

44. The data structure of claim 43, wherein the at least one event includes a timeout event.

45. The data structure of claim 43, wherein the at least one event includes an error event.

46. The data structure of claim 24, wherein the data structure is an XML transaction detail file created by a SIP redirect server.

47. The data structure of claim 46, wherein the XML transaction detail file is active for a predetermined period of time.

48. The data structure of claim 46, wherein the server identification field includes the IP address of a redirect server.

49. The data structure of claim 46, wherein the at least one transaction detail record includes a plurality of transaction detail records.

50. The data structure of claim 49, wherein the at least one transaction detail record includes a correlation identification field for correlating each transaction detail record of the plurality of transaction detail records corresponding to a particular transaction.

51. The data structure of claim 46, wherein each transaction detail record includes a time/date field indicating when the transaction detail record was made.

52. The data structure of claim 46, wherein each transaction detail record includes a request section including information about any SIP request that is received by the SIP redirect server from a SIP network server.

53. The data structure of claim 52, wherein the request section includes a received-from field indicating the IP address of an original sender of the SIP-request.

54. The data structure of claim 52, wherein the request section includes a message field having all data associated with the request.

55. The data structure of claim 46, wherein each transaction detail record includes a response section including information about any SIP response that is received by the SIP redirect server from a SIP network server.

56. The data structure of claim 55, wherein the response section includes a sent-to field indicating the IP address of the addressee of the SIP- response.

57. The data structure of claim 55, wherein the response section includes a message field having all data associated with the response.

58. The data structure of claim 46, wherein each transaction detail record includes a features section that records at least one feature executed by the SIP redirect server during the transaction.

59. The data structure of claim 58, wherein the features section includes at least one recursive routing field.

60. The data structure of claim 58, wherein the features section includes at least one originating call validation field, the at least one originating call validation field including data relating to the validation of a call originator, and data relating to the call originator.

61. The data structure of claim 58, wherein the features section includes a non trusted terminating call field for recording whether a non-trusted call was allowed.

62. The data structure of claim 58, wherein the features section includes at least one terminating call validation field, the at least one terminating call validation field including data relating to the validation of a subscriber being called, and data relating to the subscriber.

63. The data structure of claim 58, wherein the features section includes at least one originating call screening field for recording information when a call screening feature is executed.

64. The data structure of claim 58, wherein the features section includes at least one terminating call screening field for recording information when a terminating call screening feature is executed.

65. The data structure of claim 58, wherein the features section includes at least one call forwarding field for recording information when at least one call forwarding feature is executed.

66. The data structure of claim 58, wherein the features section includes at least one find-me field for recording information when a find-me feature is executed.

67. The data structure of claim 58, wherein the features section includes at least one registered address list field for recording information when a call is redirected to a list of registered addresses.

68. The data structure of claim 58, wherein the features section includes at least one default address field for recording information when the call is redirected to a default address in a subscriber's record.

69. The data structure of claim 46, wherein each transaction detail record includes at least one directory access protocol (DAP) field.

70. A telecommunications network, comprising:

at least one telecommunications apparatus configured to perform a telecommunications transaction; and

at least one SIP-server coupled to the at least one telecommunications apparatus, the at least one SIP-server being configured to create an XML transaction detail file, process the telecommunications transaction, and store at least one transaction detail record in the XML transaction detail file, the at least one transaction detail record including transaction data corresponding to the telecommunications transaction.

71. The network of claim 70, wherein the telecommunications apparatus includes a SIP device.

72. The network of claim 71, wherein the SIP device includes a telephone.

73. The network of claim 71, wherein the SIP device includes a computer.

74. The network of claim 71, wherein the telecommunications apparatus includes a SIP proxy server and the at least one SIP-server includes a SIP redirect server.

75. The network of claim 71, wherein the telecommunications apparatus includes a PBX.

76. The network of claim 71, wherein the telecommunications apparatus includes an enterprise gateway.

77. The network of claim 71, wherein the telecommunications apparatus includes a network gateway.

78. The network of claim 71, wherein the telecommunications apparatus includes a telephone.

79. The network of claim 71, wherein the at least one SIP-server includes a SIP proxy server.

80. The network of claim 71, wherein the at least one SIP-server includes a SIP conferencing server.

81. The network of claim 71, wherein the at least one SIP-server includes a SIP voice mail server.

82. A computer-readable medium having stored thereon computer-executable instructions for performing a method for recording transactions in a telecommunications network, the method comprising:

creating an XML transaction detail file, the XML transaction detail file being active for a predetermined period of time;

storing at least one transaction detail record in the XML transaction detail file in response to a telecommunications transaction, the at least one transaction detail record including transaction data corresponding to the telecommunications transaction.

83. The method of claim 82, further comprising:

closing the XML transaction detail file when the predetermined period of time expires;

creating a second XML transaction detail file, the second XML transaction detail file being active for a second predetermined period of time; and

repeating the steps of closing and creating for a number of times as directed by a network management system.

84. The method of claim 82, wherein the XML transaction detail file comprises:

an XML declaration field, the XML declaration field defining the data structure as an XML file;

a server identification field, the server identification field including an IP address of a server generating the XML file; and

a transaction detail section including at least one transaction detail record, the at least one transaction detail record being stored in the data structure in response to a telecommunications transaction, the at least one transaction detail record including transaction data corresponding to the telecommunications transaction.

85. The method of claim 82, wherein the telecommunications transaction includes a SIP-INVITE message.

86. The method of claim 82, wherein the telecommunications transaction includes a SIP-ACK message.

87. The method of claim 82, wherein the telecommunications transaction includes a SIP-OPTIONS message.

88. The method of claim 82, wherein the telecommunications transaction includes a SIP-BYE message.

89. The method of claim 82, wherein the telecommunications transaction includes a SIP-CANCEL message.

90. The method of claim 82, wherein the telecommunications transaction includes a SIP-REGISTER message.

91. The method of claim 82, wherein the telecommunications transaction includes a billing transaction.

92. The method of claim 82, wherein the telecommunications transaction includes a monitoring transaction.

93. The method of claim 82, wherein the telecommunications transaction includes a performance measurement transaction.