CONTEXT-AWARE VOIP CONGESTION CONTROL SERVICE

Gordon Agutu*, Karim Djouani*, Elmarie Biermann†, Guillaume Noel*
*F'SATI, Tshwane University of Technology, Pretoria
†F'SATI, Cape Peninsula University of Technology, Cape Town

ABSTRACT: IP networks can have difficulty coping with delay-sensitive VoIP traffics during emergency situations caused by fires and related disasters. During emergencies there is a huge increase in voice and video traffic, causing a huge strain on the network. The strain on the network is as a result of both essential and non-essential traffic. In such crisis situations, calls originating from or destined for rescue personnel, such as doctors and police, are considered essential. Any other calls from eyewitnesses and the public are considered non-essential, since they degrade the quality of service for the emergency response teams by consuming the scarce network resources. Providing the rescue team with the quality of service that they require necessitates network access restriction for non-essential traffic. In this paper, the authors present a voice and video service that uses Context-Awareness and Semantic Web technologies to restrict network access to privileged users during crisis situations. The service monitors the network for crisis conditions, enables the network to respond appropriately when a crisis occurs, detects the end of the crisis and reverts to its default state.

KEYWORDS

Call admission control, context-awareness, ontology, semantic web, web ontology language.

INTRODUCTION

Technological advancements, liberalisation, privatisation and progress in telecommunications regulation have ensured enormous growth in telecommunications markets in most African countries. For instance, the growth of Voice over Internet Protocol (VoIP) is very significant. VoIP, a technology that enables transmission of voice and data over packet switched IP-based networks, has recently become a popular means of transmission of voice and video traffic over IP networks. This is due to the cheaper option that it provides for phone calls, the flexibility of running a large number of "virtual users" through each network socket, the efficiency of combining phone calls with business data, the aspect of easily managing and maintaining the system since it is software, not hardware, based, and its support on both wireless and wired networks. The original design of VoIP was to support only voice traffic but recent developments have seen other real-time traffics such as video supported over the same platform. As the traffic is in real time, the IP platform on which VoIP is used should take care of the delay sensitive nature of both voice and video traffic. The work presented in this paper uses Semantic Web and Context-Awareness technologies to control congestion in VoIP networks caused by crisis/ emergency situations.

Emergency situations such as earthquakes, floods and volcanoes occur frequently on the African continent. During such emergencies there is a huge increase in voice, video or data traffic causing enormous strain on the network. The strain on the network is as a result of both essential and non-essential voice, video, and data traffic. In such crisis situations, calls originating from doctors, hospitals, rescue workers and emergency experts/professionals are considered essential to the rescue mission. Any other calls from the public and eye-witnesses are considered non-essential, since they degrade the quality of service for the emergency response teams by consuming the scarce network resources. The scarce network resources can be allocated fairly to the rescue team by restricting network access for non-essential traffic, thereby ensuring good quality of service for the rescue teams.

Semantic Web enhances the current web in a manner to enable computers to process the information presented on the World Wide Web (WWW), interpret the information, aggregate and combine the information from the web in a way that assists humans in finding the required knowledge (Ankolekar, Krötzsch, Tran & Vrandecic, 2008; Battle & Benson, 2008). This technology is intended to form a vast distributed knowledge-based system in the same way in which the WWW forms an immense distributed hypertext system. It is seen to dramatically change Information Technology (IT) applications such as knowledge discovery, enterprise-scale data integration, knowledge management and service-oriented architectures.

A Context-Aware system uses context to provide services or information to the user. It also takes into account the network nodes' context when carrying out network and application related computations and resource allocations (Jean & Galis, 2005). In a Context-Aware system, the context of the user is determined by the user's identity, location information, status, goal, behaviour and specific wishes. The context of computing resources is determined by network connection status, context of supported services, availability and demand of resources, and computing capabilities of various resources. Context-Awareness and Semantic Web technologies can be applied to develop a service that takes over call admission control from the Session Initiation Protocol (SIP) server during crisis situations on VoIP networks.

This paper is structured as follows: a discussion on related work on call admission control, context-awareness, and Semantic Web; presentation of the design of the proposed Context-Aware VoIP Congestion Control Service; presentation of laboratory test and simulations results; and conclusions and highlighting of future research areas.

RELATED WORK

IP networks are traditionally designed to support a best-effort service, with no guarantees on the reliable and timely delivery of packets (Houck & Meempat, 2002; Gao, Cai & King, 2005). This traditional design supports non-real time applications such as email and file transfer, which are characterised by bursty traffic, high bandwidth demands and insensitivity to delay, and delay variations. VoIP requires timely packet delivery with low delay, jitter and packet loss values. In this section, we look at related work on VoIP call admission control, application of Context-Awareness and Semantic Web.

A. VOIP CALL ADMISSION CONTROL

A significant amount of research has been done on call admission control in VoIP networks. Algorithms, frameworks, and schemes have been proposed as a result. Surveys have also been conducted on existing algorithms to determine their performance in call admission control. In Estepa & Estepa (2008), an algorithm is proposed that can be applied in admission control to determine the bandwidth reservation required to guarantee delay and loss in packet-switched multiplexer node for VoIP traffic.

The algorithm demonstrates a significant improvement in accuracy as compared to current on–off-based approaches such as fluid model and Markov Modified Poisson Process (MMPP)-based solutions.

Other channel-aware scheduling algorithms such as maximal rate (MAX) and proportionally fair (PF) are considered in Chung & Chiu (2002) with the objective of meeting the loss and delay requirements in VoIP packets. The PF algorithm is modified to support VoIP service in the CDMA2000 system. The designed frame structure shows that the PF scheduling with the proposed frame structure works well for contending VoIP packets. Further, a dynamic admission control scheme for scheduling services defined in the 802.16 specification is looked at in Wang, He & Agrawal (2007). The scheme provides the highest priority for Unsolicited Grant Service (UGS) connections. Bandwidth borrowing and degradation are employed by the scheme to maximise bandwidth utilisation. An evaluation of the scheme shows that it can guarantee the blocking probability of each class of services.

A measurement-driven framework for admission control in wireless networks is presented in Sheriff, Aravinda, Acharya & Belding (2008). Simulation results show that the proposed scheme works well to provide Quality of Service (QoS) requirements to real-time applications such as VoIP but does not address the challenges of mixed traffic network or make control admission robust enough to handle all network and traffic scenarios.

Other research works, such as Chung & Chiu (2002), show how to support QoS for VoIP by integrating SIP and 802.11e. Some adjustments and enhancements are also suggested to 802.11e to facilitate VoIP traffic over the wireless LANs. The adjustment helps in improving the network performance but does not address the handoff problem where QoS is a concern. Other proposals include Wei, Kim, Kashyap, & Ganguly (2006), where the notion of an interference capability model that can be used to design a Call Admission Control (CAC) algorithm to address the call admission control decision problem for VoIP calls on a mesh network is defined. Simulation results show that the CAC algorithm provides fewer than 20% incorrect decisions for different sizes of a multi-hop linear topology.

Early research carried out on call admission for various types of networks generally shows that a CAC algorithm satisfying QoS requirements for most types of networks has not been availed (Gao, Cai & King, 2005). In Houck and Meempat (2002), a methodology for call admission control and load balancing for VoIP services is presented. The framework is only applicable for packetised VoIP networks that support Diffserv. The framework does not address cases where multiple service provider networks are connected together. It only assumes that the entire network is a single Diffserv domain owned and operated by a single service provider. It also assumes that QoS within this network is the key item to manage.

A VoIP-based Voice over DSL (VoDSL) is proposed in Ram, DaSilva & Varadarajan (2003), with an architecture that provides QoS guarantees comparable to QoS offered by Asynchronous Transmission Mode (ATM) in the DSL access network. Both regular and premium service categories for voice traffic are supported by this architecture. The architecture also supports a best-effort service category for data traffic by using a weighted fair queuing algorithm to schedule voice and data packets for transmission over the link. It also employs a control mechanism, admission control by implicit signalling, which takes advantage of the application layer signalling by mapping it to the IP header. An evaluation of this architecture shows that it can provide QoS comparable to that provided by the Voice over ATM (VoATM) architecture.

B. APPLICATION OF CONTEXT-AWARENESS TECHNOLOGY

Context-Awareness has received attention in recent years and has been employed in developing location-based services such as enquiry and information services, community services, traffic telematics, fleet management and logistics, mobile marketing and mobile gaming. These services require constant tracking of their users to compute context and deliver relevant information to the users. Also, some search engines and systems use Context-Awareness to deliver relevant information to their users. For example, a system called Jimminy searches for information based on a person's physical environment, including location, people nearby, time of day and subject of conversation (Jones, 2005).

Building Context-Aware systems and services is a very challenging task, owing to the fact that contextual information is so dynamic in nature. The challenges related to gathering or sensing, modelling, storing, distributing and monitoring context justify the need for proper architectural support. Costa (2003), in her thesis, proposed a generic and configurable services platform architecture to support Context-Aware applications. The context platform enables dynamic deployment and management of a variety of Context-Aware services and applications. The platform also defines a subscription language used by applications and services to configure the platform to react to a given correlation of events involving contextual information. Web services are also used as a technology to enable the interactions of the platform with its environment.

Jean and Galis (2005) developed a voice service over IP that combines the capabilities of programmable networks and context awareness. The programmable Context-Aware voice (CAV) service enables VoIP services to cope with crisis situations. CAV involved simple privilege allocation and logic to deal with crisis. Also, the project was carried out only for VoIP. Recent migration of other real-time applications such as interactive video onto the IP platform requires an all-inclusive service that copes with VoIP crisis situations.

C. APPLICATION OF SEMANTIC WEB

Matheus, Kokar, Baclawski & Letkowski (2005) describe a Situation Aware Assistant (SAWA). SAWA is based on Semantic Web technologies and facilitates the development of user-defined domain knowledge in the form of formal ontologies and rule sets. In SAWA, Matheus, Kokar, Baclawski & Letkowski (2005) simply describe the functional building blocks of their system without elaborating on the technical software architecture of their system. It is not quite clear how the system addresses the challenges posed by the streaming nature of situation awareness and the cross-cutting role of ontologies. In another work, carried out by Chen, Finin & Joshi (2003), the Context Broker Architecture (CoBrA) does not discuss the aspects of scalability and reusability in their ontology-driven system architecture for pervasive context-aware environments.

In his thesis, Sotoodeh (2007) addresses the problem of interoperability between information technologies employed for emergency management. He addresses technical interoperability, organisational interoperability, and interoperability aspects of Semantic Web technologies employed. Technical interoperability is concerned with message formats, whereas semantic interoperability is concerned with the semantic technologies and definitions employed. Organisational interoperability is concerned with organisational practices and procedures. In his thesis, he describes a disaster ontology consisting of methodologies for development and evaluation of disaster ontologies and guidelines for utilisation of the disaster ontologies in practice.

Baumgartner, Retschitzegger & Schwinger (2008) address the challenges induced by the streaming nature of situation awareness and the cross-cutting role of ontologies as a technology for developing situation management systems. In their work, they develop domain-independent software architecture for situation awareness that tackles the problems of the cross-cutting nature of ontologies and the streaming nature of situation awareness, with the objective of leveraging scalability and reusability of the software components involved.

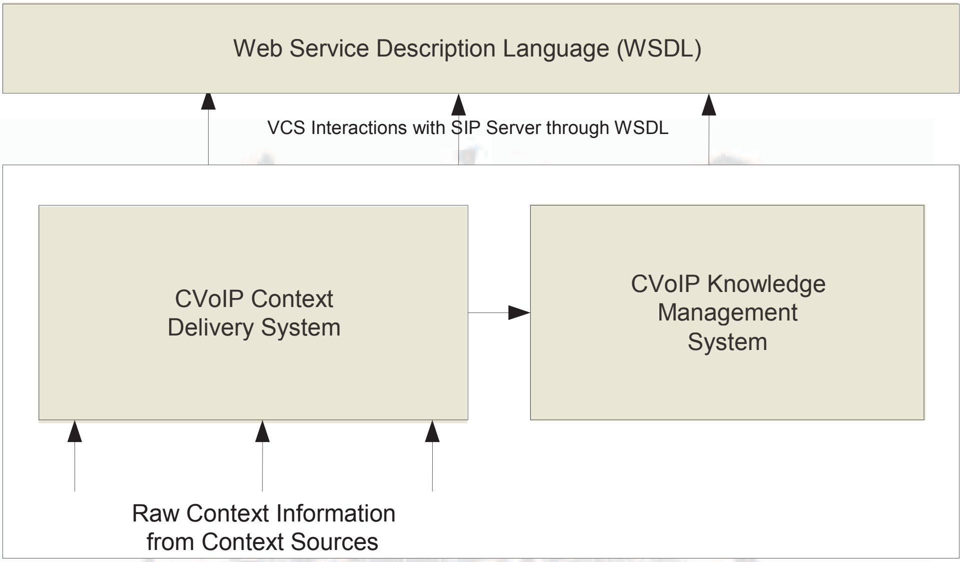
Oracle Database 11g (see www.oracle.com) is an example application that supports Resource Description Framework (RDF), RDF Schema (RDFS), and a Web Ontology Language (OWL) data management. This provides an integrated, secure, scalable, and efficient platform for semantic data management. Application developers can define a set of terms (ontologies) and relationships between data and metadata, hence adding meaning to them. The powerful technologies in Semantic Web can be used to develop applications to supplement the current admission control mechanisms.

DESIGN OF THE CONTEXT-AWARE VOIP CONGESTION CONTROL SERVICE

This section describes the architecture of the proposed Context-Aware VoIP (CVoIP) Congestion Control Service (VCS). We highlight the core components of the service, followed by detailed definitions of the architectural design of each major component. The technologies employed are also briefly explained. We further illustrate how the service detects and executes during crisis situations.

Figure 1 highlights major components of VCS. The service relies on context information collected from the VoIP service provider's network to detect a crisis and execute during crisis situations. The architecture of VCS therefore consists of a Context Delivery System that constantly collects, aggregates, and disseminates context information. Context information is provided by the service provider's equipment such as bandwidth manager, SIP server, application server, media server, and access gateway. Another major component of VCS is the Knowledge Management System, which acts as the nerve of the system during crisis situations. Also, a web interface is required to describe potential interactions with web services. The two major components of CVoIP, CVoIP Context Delivery System and CVoIP Knowledge Management System interact with the service provider's SIP servers through a web service interface defined by Web Service Description Language (WSDL).

FIGURE 1: MAIN COMPONENTS OF VOIP CONGESTION CONTROL SERVICE


Web Service Description Language (WSDL)
VCS Interactions with SIP Server through WSDL
CVoIP Context Delivery System
CVoIP Knowledge Management System
Raw Context Information from Context Sources

The internal architectures of the components highlighted in Figure 1 are constituted by sub-components such as context providers, context dissemination mechanism, context consumers and ontologies. The discussion therefore provides a detailed description of each component in the sub-sections that follow. It also highlights the importance of employing Policy-Based Service Management (PBSM) to govern how the various network and service components should act when certain conditions prevail within the network.

D. VCS CONTEXT DELIVERY SYSTEM ARCHITECTURE

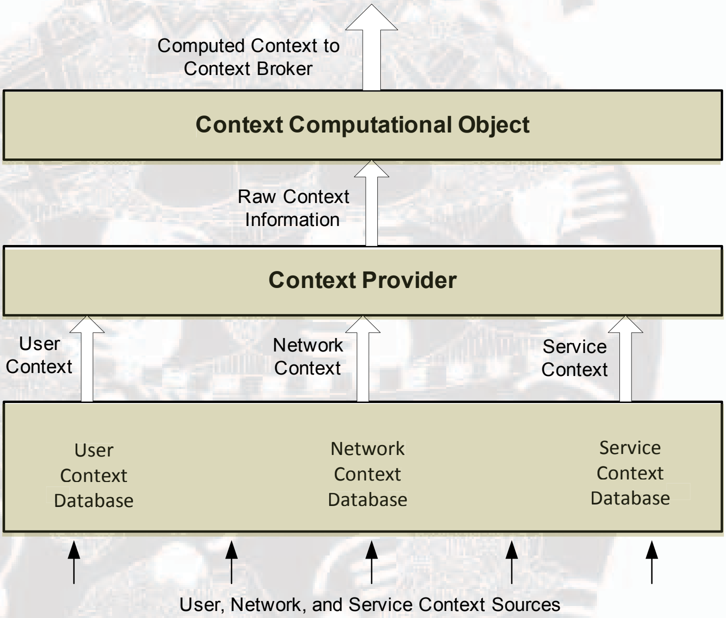
For context-aware services to adapt to their environment they need to constantly access information about it. This is achieved through context management, which deals with the collection, aggregation, and dissemination of aggregated context information. There are several raw data sources through which context information can be collected. The collected data is processed into the required meaningful context information. The processed information is then disseminated to different applications and devices situated in various locations of the network. All these processes are handled by the Context Information Dissemination System (CIDS). CIDS consists of context providers, context dissemination mechanisms and context consumers.

Context providers are the various entities located in diverse locations within the network that provide context information to the Context-Aware system. The provided information is processed and distributed through the dissemination mechanism that simply ensures that the processed context information reaches the context consumers. Context consumers are the applications and devices that use the disseminated context information.

For VCS, we consider three major categories of context information: user context, network context and service context, as highlighted in Figure 2. User, network, and service context databases act as temporary storage locations for raw context information acquired from the various context sources. The Context Provider retrieves the context information stored in these databases through push and pull requests. The Context Provider also acts as the source of raw context information for the Context Computational Object (CCO), which aggregates and filters the received context information to produce complex context information. This information is readily availed to the Context Consumers through the Context Broker.

The User Context Database (UCD) contains User Context Information (UCI) consisting of user name, profession, sub-ID, crisis password, location, priority level and preferences. This information is available from the user profile database and the SIP server located within the service provider's network. Network Context Information (NCI) is composed of different pieces of information available in different network locations. These pieces of information have to be extracted from the VoIP service provider's network equipment and used to create a general network view.

FIGURE 2: CONTEXT ACQUISITION AND PROCESSING



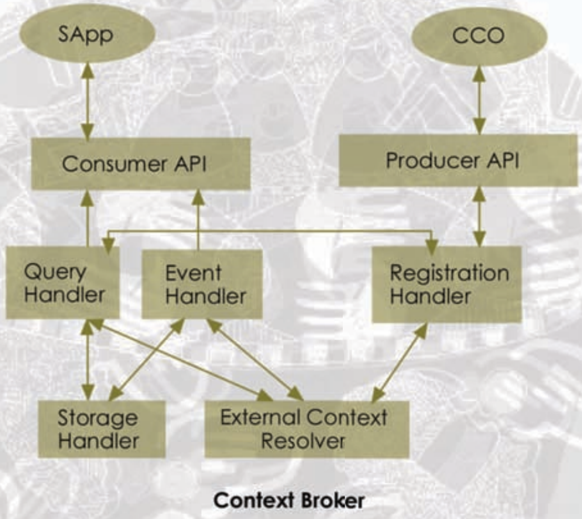
The VoIP service provider's network consists of the network elements such as call agent, media server, access network, access gateway, application server, signaling and trunking gateways. These components of the service provider's network provide the required network context for VCS. NCI consists of bandwidth, load, delay, SIP server ID, network domain ID, link states, node states, latency, jitter, loss, and error rate. NCI is the basis upon which a crisis situation is detected, which dictates that the network resources need to be managed in a different way. Contention or competition for resources is resolved by denying a given user category access.

Service Context Information (SCI) consists of service type, service priority, and minimum and maximum bandwidth allocated to the service. Under normal network operations the SCI is determined and assigned by the service provider. During crisis situations, the priority level and bandwidth are dynamically assigned by VCS, hence this determines which type of service is allowed during those situations.

CONTEXT DISSEMINATION

The context dissemination mechanism is achieved through the Context Information Dissemination Brokers. These brokers collect context information from the various context providers and disseminate the collected information to the context consumers. The brokers provide the interfaces through which the context providers publish their information and the context consumers access the provided context. This is achieved through the consumer and producer Application Programming Interface (API) as highlighted in Figure 3. The performance of the different delivery mechanisms determines which one to apply. This is guided by the achievable efficiency, scalability, and latency.

FIGURE 3: CONTEXT BROKER ARCHITECTURE



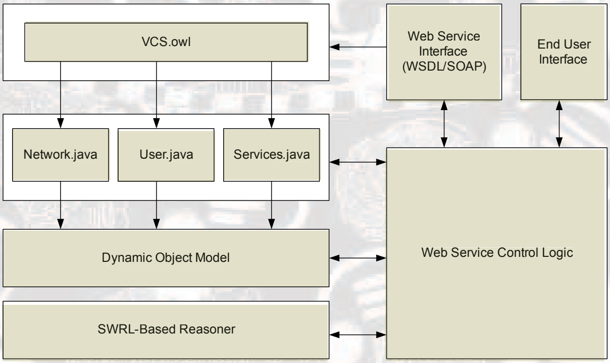
Context consumers are Semantic Web Applications (SApp) and services which interact with the system through the web service interface. These applications provide the system with network knowledge through which network resources are managed during crisis situations. The SApp deployed in the crisis area provides the SIP server with information on which calls to admit during crisis situations. The application executes by terminating all non-essential calls to and from the crisis domain and only allows privileged users access. This application stops executing when the crisis is over and relinquishes call admission control to the SIP servers. The inputs or the code in this application uses context information generated by the CCO to authenticate network access by validating user crisis access codes and priorities.

Context consumers make requests for context information to the context broker through the consumer APIs as highlighted in Figure 3. The Consumer APIs enable context consumers to access context through pull requests or push events. Also highlighted in Figure 3 is the producer API, which acts as the interface that enables context providers to deliver their produced information. The query handler resolves pull requests issued by the context consumers and triggers the external context resolver for non-local context. The event handler resolves push requests issued by the context consumers and also implements the APIs that enable the context consumers to subscribe for context events. Registration handler is the mechanism through which context producers register context data items. Storage handler retrieves context information from the context sources, stores the retrieved context, updates it, and disseminates it as required. The external context resolver is triggered whenever context is not locally available.

E. VCS KNOWLEDGE MANAGEMENT SYSTEM ARCHITECTURE

VCS Knowledge Management System consists of an ontology-based knowledge management system, SWRL-based (Semantic Web Rule Language) rule reasoner, a dynamic object model, Web service control logic, Web service interface, and user interface as highlighted in Figure 4. The ontology-based knowledge management system consists of hard-coded knowledge about the domain of VCS ontology, which comprises various classes that determine the behavior of the VCS application. The SWRL-based reasoner provides the required intelligent behavior of VCS, whereas the Web service logic controls all the system interactions. Also, the dynamic object model provides an extension to the VCS core ontology concepts about specific class events. The web service interface is achieved through the use of the Simple Object Access Protocol (SOAP) as well as WSDL.

FIGURE 4: KNOWLEDGE MANAGEMENT SYSTEM



The VCS ontology-based knowledge management system is realised through a set of tools for developing and maintaining OWL ontologies and tools for defining the rule sets. It allows VCS administrators to do offline maintenance and define new rules. The tools include an ontology editor, a consistency checker and a rule editor. Below is a brief description of each tool.

ONTOLOGY EDITOR

Ontologies are formally specified using modelling languages. The formal specification ensures proper processing and automatic operations on ontologies. Formal Semantic Web representation languages that can be used for expressing ontologies include OWL, Web Service Modelling Language (WSML), RDFS, Darpa Agent Markup Language (DAML), and DAML+ Ontology Interchange Language (OIL). These languages support the description of classes, instances of those classes, the relationships between them and the constraints on their usage. The chosen language should not limit the development of the ontology and should allow full expression of all the distinctions that we require. For this work we use OWL as the ontology language, because OWL is well defined through a formal set of semantics and the existence of a variety of automated systems to process OWL documents. The semantic interoperability of OWL also ensures that OWL-based systems and services can exchange and share context knowledge. Also, OWL is a World Wide Web Consortium (W3C) standard, which is more expressive compared with other ontology languages such as DAML+OIL, RDF, and RDFS.

RDF is the basis of OWL language. Since RDF is XML-based, any text or Extensible Markup Language (XML) editor can be used for the development of OWL ontologies. We use Protégé as the framework for building the VCS OWL ontology. Protégé is a general purpose ontology management system and provides OWL plug-ins for constructing OWL classes, properties and restrictions among these. Protégé also provides ezOWL as a plug-in that provides a graphical editor besides the basic OWL plug-in. The capabilities of Protégé OWL are supplemented (whenever necessary) by Jena to build the required OWL ontologies. Jena is an open source Java-based framework for building Semantic Web applications. It provides a programming environment for SPARQL Protocol and RDF Query Language (SPARQL), OWL, RDF, and RDFS. It also includes a rule-based inference engine.

FIGURE 5: VOIP CONGESTION CONTROL SERVICE OWL ONTOLOGY REALISATION ON PROTÉGÉ

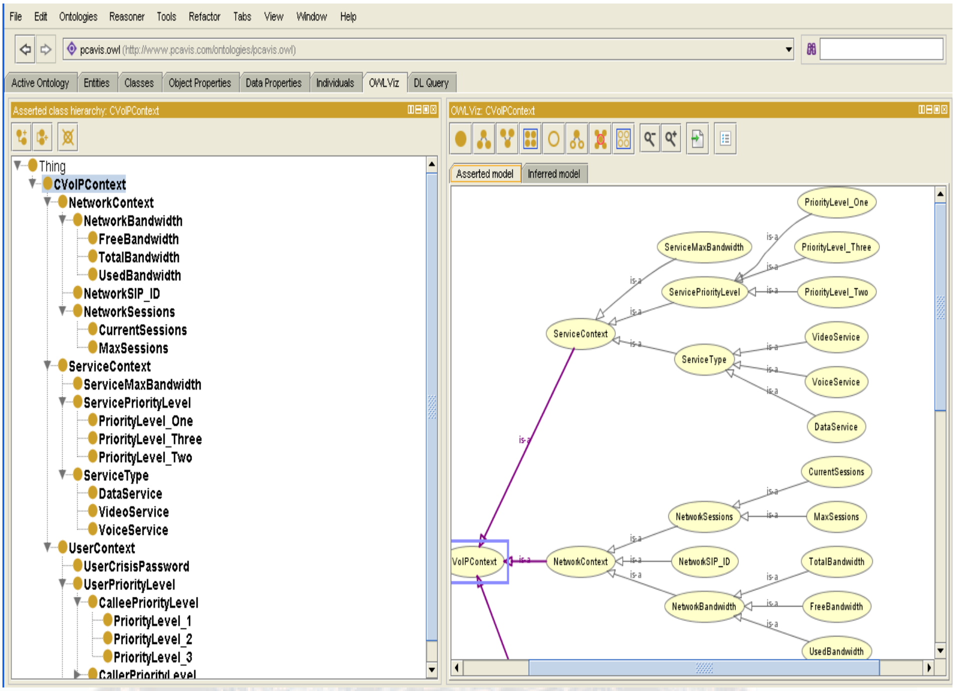


Figure 5 highlights the classes, subclasses, and the hierarchical graphical structure of the VCS ontology developed using the OWL plug-in in Protégé. OWL classes are sets that contain individuals. Individuals in this context represent objects in the domain of VCS. The binary relations between individuals are specified using properties. OWL classes are described using formal descriptions that clearly specify the requirements for membership of the class. VCS ontology hierarchy consists of super-classes and sub-classes. For example, the class ServiceContext consists of sub-classes such as ServiceType, ServicePriorityLevel, and ServiceMaxBandwidth. The hierarchy of super-class and sub-class is relative based on the hierarchy level. This implies that a sub-class of some given super-class might be a super-class to classes that appear below it in the hierarchy.

CONSISTENCY CHECKER

Ontology is said to be inconsistent if any one of its parts does not agree with another. Reasoning with an inconsistent ontology may lead to wrong conclusions, actions and reactions. OWL consistency checkers perform the task of automatically validating the constructed ontology for logical consistency. This ensures accuracy and consistency in the development of complex ontologies. We use ConsVisor, a Versatile Information Systems Inc standalone open Java application as the VCS OWL/RDF consistency checker. The tool analyses OWL and RDF documents by looking for any signs of semantic inconsistencies and also identifies incomplete specification of logical implications in a document. ConsVisor handles formal ontologies including recent ontology languages (such as OWL Full, OWL Description Logic (DL) and OWL Lite) and traditional data modelling languages.

RULE EDITOR

Rule editors assist with the construction and maintenance of Semantic Web Rule Language (SWRL) rules. SWRL combines OWL and Rule Markup Language (RML) sublanguages. Protégé-OWL provides the SWRLTab as the development environment for editing and executing SWRL rules. The SWRL editor component of SWRLTab supports manipulations and saving of SWRL rules in OWL ontology.

POLICY-BASED SERVICE MANAGEMENT

Policy-Based Service Management (PBSM) consists of the necessary policies, rules, procedures, guidelines, practices and standards that are used to support the business, business processes and their interrelationship. PBSM enables the service providers to know their customers, services and applications to which the customers have subscribed, when to give users access to the network, and how to offer the subscribed services to the users. In crisis situations, the PBSM gives guidelines on how to handle and manage the crisis. Different policies are applied depending on the type of crisis detected. The input to this system is the crisis information passed over by SApp. The output is a set of rules based on the business policies and guidelines to execute or ignore the request.

F. WEB SERVICE DEVELOPMENT LANGUAGE INTEGRATION

The technology in web services allows applications to communicate with each other independent of the platform on which they run and the programming language used. This technology is supported by the standardised architecture of web applications and various enabling technologies such as Web Services Development Language (WSDL), Simple Object Architecture Protocol (SOAP) and Universal Description, Discovery, and Integration (UDDI). Figure 6 highlights the relationship between these enabling technologies.

An industry standard web service employs enabling technologies for web services to provide a service description consisting of WSDL documents. The web service is also capable of transporting XML documents using SOAP over HTTP while at the same time providing service discovery through UDDI.

For Web services to be interfaced with and discovered by other services and applications, they need to be defined in a consistent manner. Web Services Definition Language (WSDL) is a W3c specification which provides the language for description of web service definitions. The integration layer introduced by the web services framework establishes a standard, universally recognised and supported programmatic interface. For VCS to interact with and take over call admission control from the SIP server, a WSDL interface is required.

FIGURE 6: RELATIONSHIP BETWEEN WEB SERVICES ENABLING TECHNOLOGIES

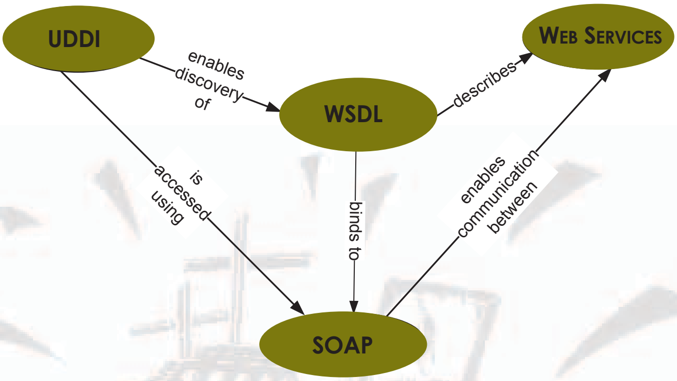
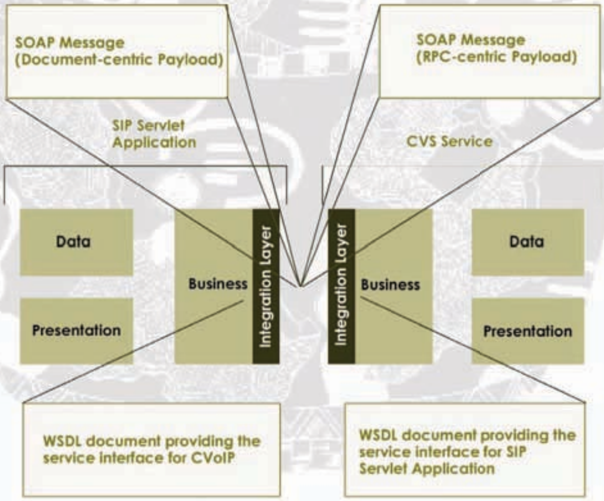


Figure 7 highlights how WSDL interfaces VCS and the SIP server. The presentation, business, and data layers shown in Figure 7 are the core layers of an enterprise web application. The presentation layer generates web pages and decodes web pages coming back from the client, thereby finding the user entered data and passing the contained information to the business logic layers. The layer also includes dynamic content, which originates from a database, in the web page. The role of the business layer includes performing all required calculations and validations, managing workflow, and managing all data access for the presentation layer. The data layer plays the role of managing data. It stores data whenever requested and provides the business layer with required data whenever needed. The data layer is available in the form of a relational database. It may also provide data access procedures to other data sources.

FIGURE 7: VOIP CONGESTION CONTROL SERVICE INTERACTIONS WITH SIP SERVER THROUGH WEB SERVICES DEVELOPMENT LANGUAGE INTERFACE

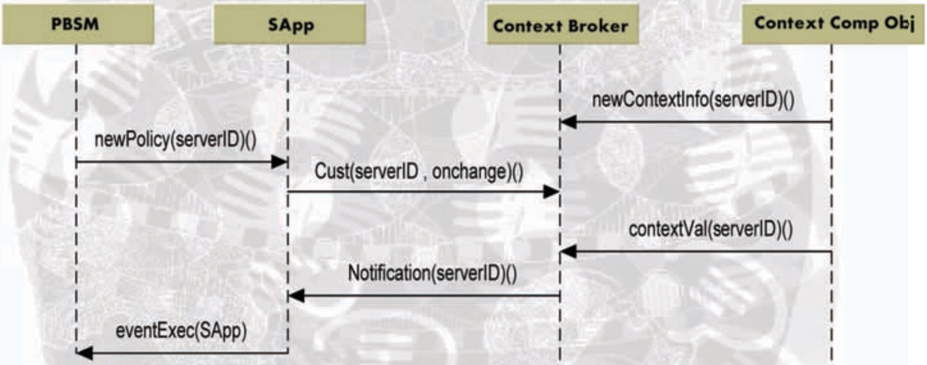


Also highlighted in Figure 7 is the transportation of XML documents using SOAP over HTTP. SOAP was originally designed as a bridge between Remote Procedure Call (RPC)-based communication platforms. It has evolved with time into a protocol for use with XML Web services and a widely supported messaging format. The messaging format consists of an XML document which hosts both document and RPC-centric data, hence supporting both synchronous and asynchronous data exchange models. UDDI provides the mechanism through which web services are discovered. It provides a central directory that hosts service descriptions.

G. VCS CRISIS DETECTION AND EXECUTION

Figure 8 illustrates the crisis detection interactions among some of the service components. The Context Computational Object continuously provides the Context Broker with new context information by executing the newContextInfo(serverID) function. The provided context information contains network parameters which determine whether a crisis situation has occurred or not. For VCS, a crisis situation is detected whenever the set call threshold for the emergency number 112 is exceeded. The set threshold varies with the SIP server domain under consideration. The Semantic Web Application (SApp), by executing cust(serverID,onchange) function, retrieves the computed context information through the Context Broker and evaluates it for crisis situations. At the same time the Context Broker continuously updates SApp with context information by executing the notification(serverID) function. SApp informs the PBSM whenever a crisis occurs through the eventExec(SApp) function.

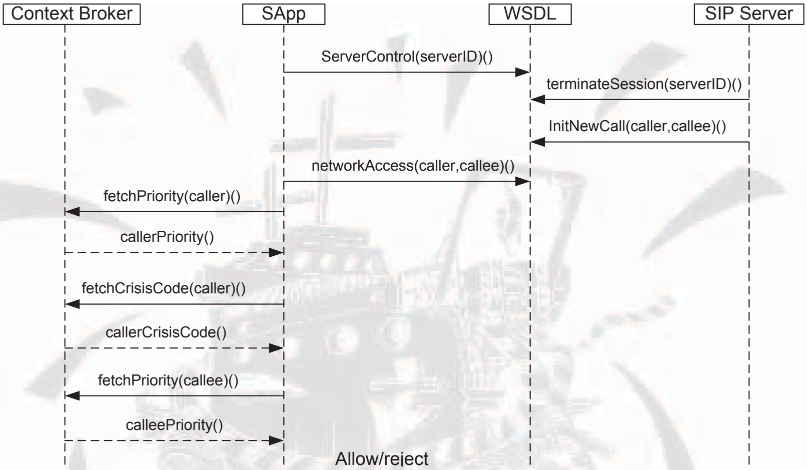
FIGURE 8: VOIP CONGESTION CONTROL SERVICE CRISIS DETECTION



PBSM consists of policies which determine and authorise various actions and reactions of network applications under different circumstances. These policies are applied by executing the newPolicy(serverID) function.

Figure 9 highlights VCS execution during crisis situations. Whenever a crisis is detected SApp takes over admission control from the SIP Server by executing serverControl(serverID) function. SIP Server and SApp interact with each other through WSDL. The SIP server responds to the serverControl request by executing terminateSession(serverID) function and consults SApp on every new call initiation request that it receives. Caller and callee priority levels must satisfy the set crisis priority levels for a call session to be established between them. Also, the caller crisis code must be authorised.

FIGURE 9: VOIP CONGESTION CONTROL SERVICE EXECUTION INTERACTIONS



The SIP server, WSDL, SApp, and Context Broker interact with each other by executing various functions. These functions take in caller, callee, or both as the input parameters. InitNewCall(caller,callee) function is executed by the SIP server whenever it is seeking network access authorisation for a new call. The interactions between SApp and the Context Broker through functions fetchPriority(caller), fetchCrisisCode(caller), and fetchPriority(callee) determine whether a call initiation request is accepted or rejected.

LABORATORY TEST BED IMPLEMENTATION AND SIMULATION RESULTS

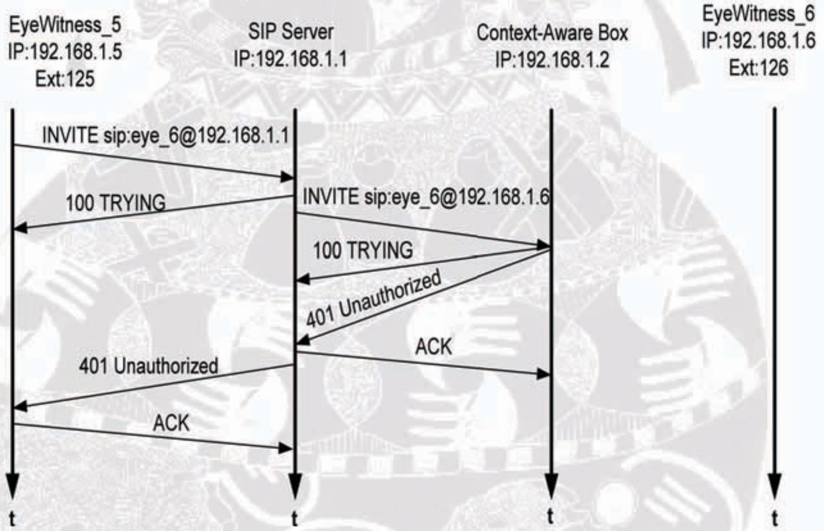
In this section we demonstrate how the service functions through laboratory test scenarios and OPNet simulations. Our main focus is how the service takes over call admission control from the SIP Server, rejects non-privileged calls and drops non-privileged ongoing call sessions. The laboratory test bed consists of a SIP Server, two network switches, 10 client computers, a web server and application server. We installed and configured OpenSER on a Linux (Debian) platform. Our clients are notebook computers running X-Lite SIP client application. We further carried out simulations on OPNet for network performance analysis using parameters such as delay, throughput, connected calls, rejected calls and dropped calls. Some of these results are presented in the subsections that follow.

H. CALL MANAGEMENT DURING CRISIS SITUATIONS (LAB TEST)

During crisis situations the SIP Proxy server consults the Context-Aware System (Box) for every session request received from the clients. This is to ensure that only privileged users are authenticated by the Context-Aware Box to access the network. We present two scenarios where call initiation requests from eyewitnesses are either rejected or an ongoing session terminated. The termination of an ongoing session results in dropped calls. The Context-Aware Box determines which calls to allow based on the ontology-based knowledge management system and updated captured context information. These systems consist of user priorities that are assigned based on the crisis situation. Whenever a call initiation request is received by the Context-Aware Box, it checks both the caller and callee priorities to determine if the session should be established by the SIP Proxy server or not. The session request is automatically rejected if either of the involved parties is a non-privileged user.

Figure 10 presents a scenario where an eyewitness tries to make a call to another eyewitness. Both parties in this case are non-privileged users. The presence of a crisis situation dictates that the call initiation request should be rejected. Eye_5 sends an INVITE request to Eye_6 through the SIP Proxy server. The SIP Proxy server sends back a 100 TRYING message back to Eye_5 indicating that the request is being processed. At the same time the SIP Proxy server sends the INVITE request to the Context-Aware Box for user priority confirmation and authentication.

FIGURE 10: SIP TRANSACTIONS (NETWORK ACCESS DENIED)

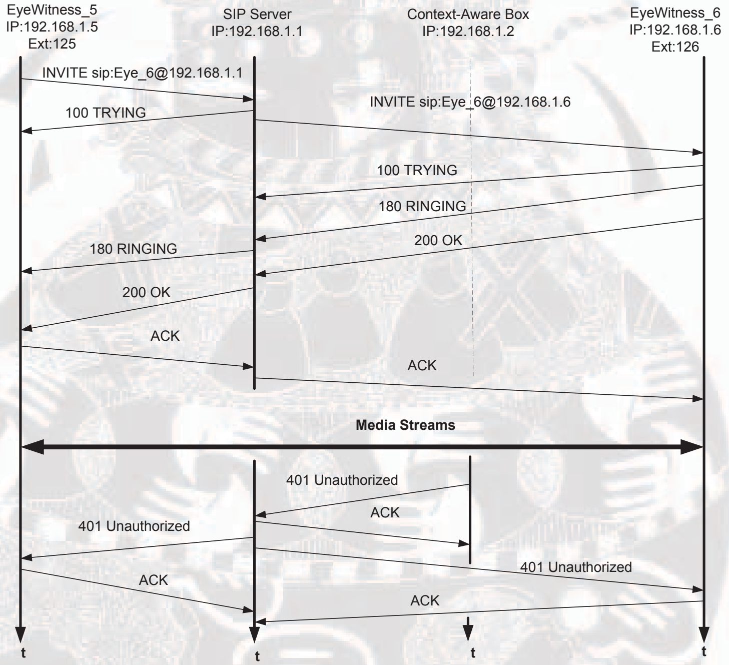


The Context-Aware Box sends back a 100 TRYING message back to the SIP Proxy server as it processes the INVITE request. The user is determined to be a non-privileged user and the Context-Aware Box sends a 401 UNAUTHORISED message to the SIP Proxy server. The SIP Proxy server acknowledges the UNAUTHORISED message by sending an ACK to the Context-Aware Box and at the same time sending the UNAUTHORISED message to Eye_5. Eye_5 sends an ACK to the SIP Proxy server hence marking the end of the call initiation request.

Eye_6 does not receive any information or communication about the aborted call initiation hence ensuring that bandwidth and other network resources are not consumed unnecessarily. Other scenarios related to this include a privileged user calling a non-privileged user such as a doctor calling an eyewitness, a non-privileged user placing a call to a privileged user, such as an eyewitness calling a doctor. The same process, illustrated in Figure 10 takes place before any non-privileged call can be established.

The scenario highlighted in Figure 11 illustrates a premature termination of an ongoing media exchange session between two non-privileged users. The session started when the network was operating under normal conditions without involving the Context-Aware Box. Immediately a crisis situation is detected the Context-Aware Box comes into play and determines which sessions are to be terminated. The termination is dictated by network resource requirements during the crisis situation. The system continuously checks the ongoing sessions for privileges and priorities of the involved parties. This results in dropping of non-privileged users' calls.

FIGURE 11: SIP TRANSACTIONS (DROPPED CALL)



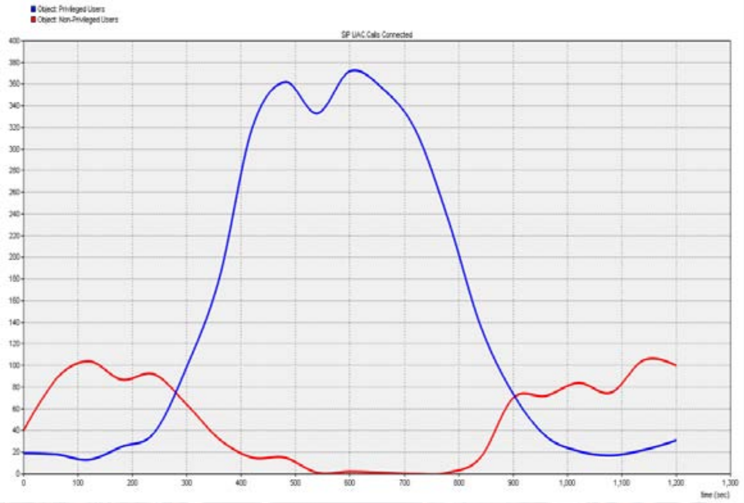
Before detection of the crisis situation, Eye_5 and Eye_6 were exchanging media information. Immediately the crisis situation is detected, the Context-Aware Box takes admission control from the SIP Proxy server and also checks the ongoing sessions to determine which calls should be dropped. In this scenario, the Context-Aware Box determined that the call session between Eye_5 and Eye_6 is non-privileged and should be terminated. It then sends a 401 UNAUTHORISED message to the SIP Proxy server, which acknowledges the message through an ACK. The Proxy server at the same time sends a 401 UNAUTHORISED message to Eye_5 and Eye_6. Both Eye_5 and Eye_6 acknowledge the 401 UNAUTHORISED message by sending an ACK to the SIP Proxy server, hence terminating the media exchange session between the two parties.

Another scenario related to the one illustrated in Figure 11 includes termination of non-privileged sessions initially authenticated by the Context-Aware Box during the crisis situation. For example, a media exchange session between a doctor and an eyewitness may be initially authenticated if network resources are available to sustain it. If the crisis situation worsens and more resources are required the session is automatically dropped in favor of a new session between two doctors.

i. Connected and rejected calls (OPNET simulations)

Figure 12 shows the total number of calls connected for both the privileged users and non-privileged users. Connected privileged user calls comprise the total of connected calls originating from or destined to medical doctors, nurses, police officers, military personnel, rescue experts, and fire brigade personnel. Non-privileged user calls consist of connected calls originating from or destined to eyewitnesses and the general public. From the figure we can see that the network operates with normal traffic flow from 0 seconds to 250 seconds. The non-privileged user traffic during this period is higher than privileged user traffic. This is due to their heavy calling characteristic and general categorisation of some privileged users as non-privileged.

FIGURE 12: CALLS CONNECTED BY THE SIP SERVER

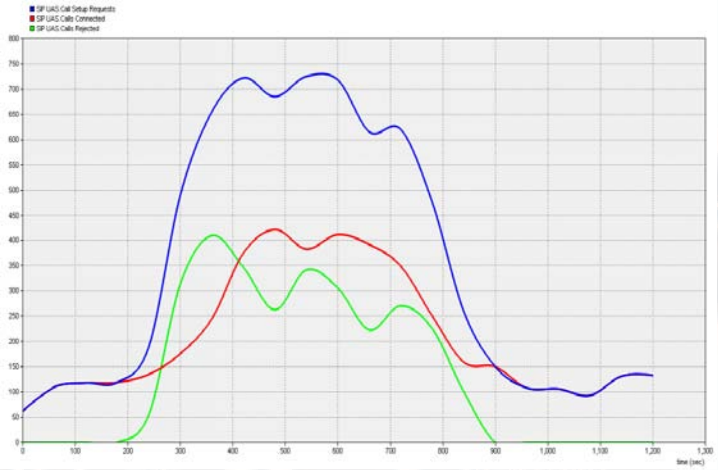


When a crisis is detected, the network through the SIP server slowly restricts network access to privileged users. This is observed between 250 seconds and 850 seconds. The peak of the crisis situation is observed around the 600th second where the total number of connected privileged user calls rises to 375 seconds, whereas the total number of connected non-privileged user calls reduces to zero. Zero connected calls for the non-privileged users implies that the ongoing sessions were dropped and any new call initiation attempts were blocked by the SIP server.

At the end of the crisis situation, the network reverts to its normal traffic loads where the connected non-privileged user calls are higher than the connected privileged user calls. This is observed around the 800th and 1 200th seconds, where the connected privileged user calls slowly drops and later stabilises around the 1 000th second. The drop in connected privileged user calls is due to successfully terminated call sessions and a drop in new call initiation requests. The rise in connected non-privileged user calls is due to new successful call initiation requests and a reduction in the number of calls dropped.

The drop in connected non-privileged user calls and the rise in connected privileged user calls implies that some call requests were accepted and others denied. Figure 13 highlights the total number of call initiation requests, total connected calls and total rejected calls from both privileged and non-privileged users. This is an observation of activities at the SIP server. It is observed that all call connection requests are successful between 0 seconds and 200 seconds of the simulation period. During this period the total of connected calls is equal to the total number of call initiation requests and rejected calls are zero. This implies that the network was operating below its resource availability and SIP server maximum call connection constraints. The crisis situation dictates that some calls are connected while others are rejected. The connection or rejection is determined by the defined network policies and the captured network context information. A crisis is observed between the 250th second and the 850th second of the simulation period. During this time the total number of calls connected is lower compared to the total number of call requests, hence implying a rise in total number of rejected calls.

FIGURE 13: TOTAL CALLS CONNECTED, REJECTED, AND REQUESTED AS OBSERVED AT THE SIP SERVER



An observation of the configured client nodes indicates that none of the rejected calls originated from or was destined to the privileged users. All the observed call rejections were at the public and eyewitness client nodes. At the end of the crisis situation (800-1 200 seconds) the network reverts to its default state where no call rejections are observed and all call initiation requests are successful.

FIGURE 14: CONNECTED CALLS AS OBSERVED AT EACH SIP CLIENT NODE

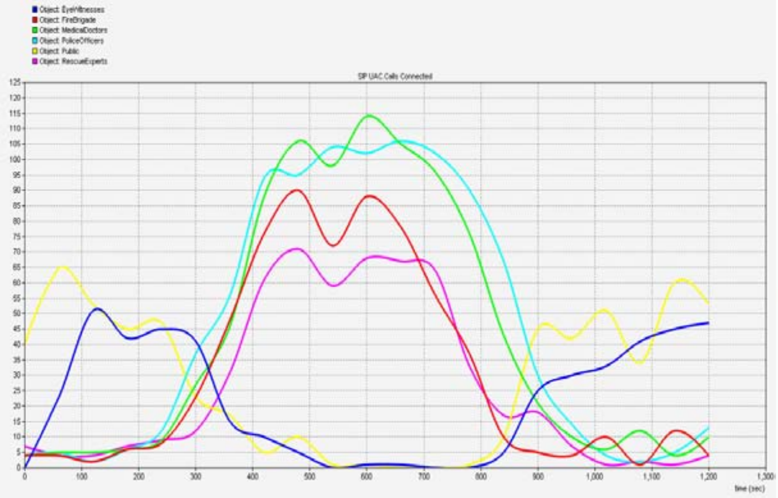


Figure 14 highlights a decomposed comparison of connected calls observed at each of the client nodes. The number of connected privileged user calls rapidly rises during the crisis situation and the number of connected non-privileged user calls rapidly reduces to zero or almost zero for the reasons stated above. Our interest here is the comparison of total connected calls among the privileged users. From the figure, the authors observe that doctors and police personnel were the most urgently required personnel during the crisis situation. The total number of military and rescue experts present was a little lower, compared to doctors and police.

CONCLUSION AND RECOMMENDATION FOR FUTURE WORK

J. CONCLUSION

This paper presented the design, implementation and simulation of a Context-Aware VoIP congestion control service. The paper presented detailed descriptions of the Context-Aware system and the knowledge management system that are core to VCS. In the design section, the authors highlighted the various function executions in monitoring, detecting, and taking over admission control from the SIP servers. The laboratory experiments concentrated on how the service responds to crisis situations by taking over call admission control. Simulation results have shown that privileged calls are given more priority compared with non-privileged calls. Other results, which are not presented in this paper, show that network performance requirements are met during crisis situations if VCS is deployed, but suffers severe quality of service degradation without VCS. This observation is based on the end-to-end delay time and throughput performance achieved with and without VCS. The high level of performance is achieved by dropping and rejecting non-privileged calls and at the same time ensuring that all privileged calls are connected.

With the realised improvement in the environment for networked readiness in Africa, the innovative service developed and reported in this paper, if implemented, could relatively improve the quality of VoIP in a manner that matches the qualities experienced in other networks, such as the Public Switched Telephone Network (PSTN). Also, the dynamic technology of VoIP as evidenced in this paper should be considered as part of many other approaches to addressing telecommunications problems in Africa and other parts of the world.

K. FUTURE WORK

This work opens up several interesting research issues for future work. Firstly, in this paper the authors addressed congestion control in VoIP networks only. It would be very interesting to make adjustments to VCS to manage congestion and call admission control challenges in other networks, such as the everyday evolving GSM-based networks. Secondly, this paper considered only voice and video traffic. It would be interesting to incorporate other services such as Instant Messaging. Lastly, network convergence is the way forward. Further research work should exploit the evolving technologies in Semantic Web and Context-Awareness to address admission control issues which come with network convergence.

REFERENCES

Ankolekar, A., Krötzsch, M., Tran, T. & Vrandecic, D. (2008). *The two cultures: Mashing up Web 2.0 and the semantic web. Web Semantics: Science, Services and Agents on the World Wide Web,* Volume 6, 70-75.

Battle, R. & Benson, E. (2008). *Bridging the semantic web and Web 2.0 with representational state transfer (REST). Web Semantics: Science, Services and Agents on the World Wide Web, V*olume 6, No.1, 61-69.

Baumgartner, N., Retschitzegger, W. & Schwinger, W. (2008). *A software architecture for ontology-driven situation awareness. In Proceedings of the 23rd Annual ACM Symposium on Applied Computing Fortaleza,* Ceará, Brazil, 16-20 March.

Chen, H., Finin, T. and Joshi, A. (2003) *Using OWL in a Pervasive Computing Broker. In Proceedings of the Workshop on Ontologies in Agent Systems (OAS 2003)*, Melbourne, Australia.

Chung, S. & Chiu, C. (2002). *Joint call admission control/congestion control for wireless integrated voice/data networks. Computer Communications*, Volume 25, 1653-1664.

Costa, P. (2003). *Towards a service platform for context-aware applications.* Thesis for a Master of Science degree in Telematics, The University of Twente, Netherlands.

Estepa, A. & Estepa, R. (2008). *Accurate resource estimation for homogeneous VoIP aggregated traffic. Computer Networks*, Volume 52, No.13, 2505-2517.

Gao, D., Cai, J. & King, N. (2005). *Admission control in IEEE 802.11e wireless LANs. Network, IEEE*, Vol. 19, No. 4. (25 July 2005), 6-13.

Houck, D. & Meempat, G. (2002). *Call admission control and load balancing for Voice over IP. Performance Evaluation*, Volume 47, Issue 4, 243-253.

Jean, K. & Galis, A. (2005). *A programmable context-aware voice service. London Communications Symposium.* University College London, Department of Electronic & Electrical Engineering, Torrington Place, London.

Jones, G. (2005). *Challenges and opportunities of context-aware information access.* In Proceedings of the 2005 IEEE International Workshop on Ubiquitous Data Management (UDM), 53-60, 4 April.

Matheus, J., Kokar, M., Baclawski, K. & Letkowski, J.L. (2005). *An application of semantic web technologies to situation awareness. In Proceedings of the Fourth International Semantic Web Conference (ISWC)*, 944-958, 6-10 November, Galway Ireland.

Ram, L., DaSilva, L.A., & Varadarajan, S. (2003). *Admission control by implicit signaling in support of Voice over IP over ADSL. The International Journal of Computer and Telecommunications Networking*. Volume 44 Issue 6, 22 April.

Sheriff, I., Aravinda, P., Acharya, P. & Belding, M. (2008). *Measurement-driven admission control on wireless backhaul networks. Computer Communications*, Volume 31, Issue 7, 1354-1371.

Sotoodeh, M. (2007). *Ontology-based semantic interoperability in emergency management*. Thesis for a Doctor of Philosophy degree in the

Department Electrical and Computer Engineering, University of British Columbia, Vancouver.

Wang, H., He, B. & Agrawal, D. (2007). *Above packet level admission control and bandwidth allocation for IEEE 802.16 wireless MAN. Simulation Modelling Practice and Theory*, Volume 15, 366-382.

Wei, H., Kim, K., Kashyap, A. & Ganguly, S. (2006). *On admission of VoIP calls over wireless mesh network*. In Proceedings of IEEE International Conference on Communications, 11-15 June, Istanbul, 1990-1995.