Three tiered web-based manufacturing system—Part 1: System development

S. P. Lal[a] *, G.C. Onwubolu[b]

[a]Department of Mathematics and Computing Science, University of the South Pacific, Suva, Fiji Islands

[b]Department of Engineering, University of the South Pacific, Suva, Fiji Islands

This paper provides a demonstration of harnessing the power of the Internet to enable collaborative manufacturing in such a setup where the key components such as the human resource personnel and the manufacturing equipment are geographically separated. As part of the framework a platform independent and cost-effective means of remotely operating a 3-axis computer numerical control (CNC) drilling machine securely via the Internet has been developed. In addition, an infrastructure to allow remote manufacturers to collaborate on a product design as well to monitor the drilling process in real time is presented. The framework rests on sound security foundation implemented using technologies such as firewall, authentication modules and transport layer security (TLS) protocol to promote safe usage of the drilling machine. It is worth mentioning that the architectural design is generic and modular in nature and as such, any other genre of CNC machine such as milling and turning could be substituted in place of drilling with only machine-specific changes to the existing system.

Keywords: Virtual enterprise; Collaborative manufacturing; Web-based distributed system; CNC machine; Remote control; Security

1. Introduction

In the globally competitive market, the ability to produce cost-effective products under strict time frame and quality control dictates the survival of the manufacturing industry. This kind of external pressure from the customers has created the need to re-look at the way the manufacturing process is conducted in a typical factory setup. As a consequence there has been a gradual shift from the traditional self-contained style of manufacturing to a global customer driven manufacturing approach, which has given birth to the concept of virtual enterprise.

The concept of virtual enterprises [1] allows geographically separated manufacturers to build partnerships so as to embrace external resources and services without owning them. Partnerships can eventuate in many forms. Outsourcing the supply of parts and sub-assemblies to specialist firms is notably common. Another approach in forging such collaborative partnerships could involve geographically dispersed manufacturers developing products by operating manufacturing equipment belonging to other manufacturers, part of the virtual enterprise. Needless to say this would require the manufacturing equipment in question to have provisions for safe remote control.

Having remotely controlled manufacturing units bears profound implications on the manufacturing industry. For instance a product manufacturer can choose to locate a set of manufacturing equipment close to the customer market, while the set of skilled people to operate the manufacturing equipment can be stationed in any branch depending on their skill set and the needs of the company. Such an arrangement diminishes the geographical separation between human re-

*Corresponding author. Complex Systems Laboratory, Department of Information Engineering, Faculty of Engineering, University of the Ryukyus, 1 Senbaru, Nishihara, Okinawa 903-0213, Japan. Tel.: +8190 3797 3479; fax: +8198 895 8727. E-mail address: lal_s@usp.ac.fj (S.P. Lal).

source personnel and manufacturing equipment. Furthermore, if properly coordinated it can introduce significant cost cutting as products produced closer to the customer base will not incur massive shipment charges thereby giving a competitive edge to the product manufacturer.

Past research efforts to address the current global manufacturing trend has focused on development of webbased simulation for analysis of product design [2]. Other attempts have gone as far as development of virtual environment for the simulation, optimization and generation of prototype via the Internet [3,4]. Along similar line is the Internet-based virtual manufacturing system using CORBA which enables simulation of analysis, design and experimentations to take place prior the actual production on the machining center [5]. Some examples of praise worthy efforts are: CryberCut [6] which uses webCAD and process planning allows users to have finer control in product design in terms of choosing tools, materials, tolerance and finishing, before actually manufacturing it with the computer numerical control (CNC) milling machine. Internet Manufacturing (IMAN) [7] relies on the Internet for the implementation of distributed rapid prototyping, allowing platform independent means of remotely controlling a manufacturing process.

While some researchers have developed their own clientserver architecture for communication, others [8] have opted to use proprietary software like pcAnywheres to remotely control and monitor a given manufacturing process. The use of proprietary remote control software over customized client-server architecture may be easy to implement but there are often limitations that accompany such software. One of the concerning limitations is the inability to properly coordinate multi-user access to the machining unit.

The research reported in this paper focuses on the development of web-based manufacturing system for the remote operation and monitoring of the PC-based CNC drilling machine [9] which was developed at the Department of Engineering, University of the South Pacific, Fiji. The design approach we have adopted has obvious advantages, in that instead of developing customized client-server architecture, it relies on well-defined and seasoned client-server architecture running on the Internet, which is the worldwide web (www), to serve application logic in the form of Java applets, which provide platform independent means of accessing the drilling machine. Therefore, the clients only need to have a web browser to use our system.

The paper is organized as follows. Section 2 captures the design and development of the webbased manufacturing system. The security implication of deploying the webbased manufacturing system on the open Internet forms the subject of discussion in Section 3. Section 4 looks at the scalability of distributed architecture in incorporating other genre of CNC machining units in the web-based manufacturing system. Section 5 concludes the paper with pointers for further research. The full version of this paper can be referred from [10].

2. The web-based manufacturing system

The web-based manufacturing system reported herein has been developed with modular architecture as shown in Fig. 1. This section describes the design considerations, details of each of the modules and how these modules fit together to form the overall web-based manufacturing system.

2.1. Design considerations

Based on the research work of Lee et al. [8], preliminary investigations into the remote control of the CNC drilling machine were carried out using commercial off-the-shelve (COTS) remote control software such as NetOp, VNC and Telnet. There were some limitations on the part of the remote control software in its current configuration to fully control the manufacturing process. These shortcomingslead to identification of the following key requirements which were factored into the final design of the remote control architecture.

2.1.1. Capturing global audience

The users of the proposed system can be in any part of the world and therefore it is vital that the system be widely accessible. Developing a custom application protocol for teleoperation is bound to use unprivileged ports for communication which

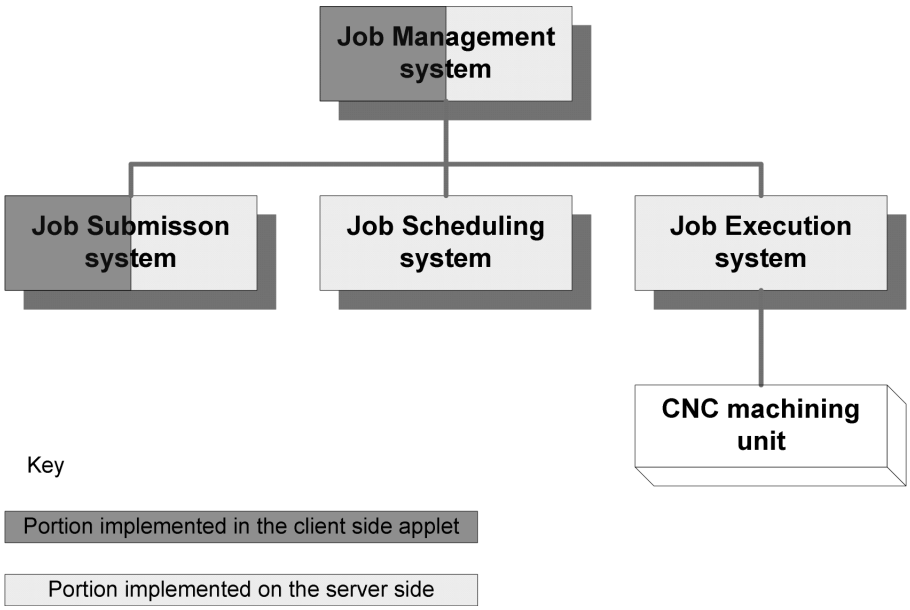

Fig. 1. Modular architecture of the web-based manufacturing system.

is not favored as such ports are typically blocked by network administrators. Network tests carried by Dalton [11] show 13% of connections made from remote hosts to an unprivileged port failed, the cause of which is attributed to prohibition by firewall and proxy configuration. Clearly this percentage is significant in missing out on capturing the global market. Moreover distribution, maintenance and administration of such applications in a distributed environment are some issues that need to be considered. Instead a widely accessible protocol, which provides an environment conducive for remote control operation is preferred.

The World Wide Web is by far one of the most popular, well-defined and seasoned client-server architecture currently running on the Internet and has been used in some of the remote control applications in the literature. Not only Java provides platform independence, but also the Java Virtual Machine (JVM) which interprets byte codes is distributed by Sun Microsystems without any licensing or other fees. The synergy of creating applets using Java and distributing over the World Wide Web provides the ability to reach the widest audience in terms of cost, convenience and platform independent means of access.

2.1.2. Handling data loss and out-of-order delivery of data

Though the Internet offers a lucrative communication platform for remote control applications, it has its fair share of problems. One of the problems facing real-time remote control systems such as robots, particularly operating in direct control mode is uncertain data-loss problem [12].

Now that the web has been chosen as a platform for development, it should be noted that the application layer protocol for the web, HTTP runs over transport layer protocol, TCP which provides reliable (connection-oriented) in-order byte stream data transfer. Therefore, TCP elegantly takes care of the problem associated with data loss and out-of-order delivery of data.

2.1.3. Coordinating multi-user access

The nature of operation of the drilling machine is such that only one client should have full control over the drilling machine at any given time. Due to this restriction, coordination of multi-user access is vital in ensuring fairness of use and safe operation of the machining unit. The lack of

multi-user coordination has been highlighted as the major limitation of the COTS remote control software.

Allowing users real-time direct control over the machining unit is not necessary. Instead the system can allow users to submit job files via Common Gateway Interface (CGI) [13]. These job files can be placed in a queue and executed on the machining unit in the order in which they arrive to the queue.

2.1.4. Reduced susceptibility to temporal communication delays

An important consequence of deciding against real-time direct control of the machining unit is that the susceptibility of the proposed system's performance on the network condition is greatly reduced, thus addressing the uncertain time-delay problem highlighted by Luo et al. [12]. In submitting jobs, the time taken for the job to arrive to the manufacturing centre depends primarily on the effective connection bandwidth and size of the job submitted. Once the job arrives in queue, the network conditions have no bearing on the execution of the job.

2.1.5. Security issues

Remote control software, by the nature of their design gives more than the desired amount of control to the remote user. Apart from accessing the applications relating to the drilling machine, the remote user has access to the operating system utilities as well as the file system. Managing and often restricting access to other parts of the system is particularly difficult given that the remote user has to have administrative privileges in order to execute the drilling machine controller.

The proposed framework is built on a security foundation consisting of firewall, web server access control mechanisms and transport layer security (TLS) which provide adequate security measure for the safe operation of the drilling machine. Moreover, with the web-based manufacturing system the authorized users no longer need administrative privileges to submit jobs and accordingly access to the operating system utilities and file system is restricted to prevent abuse.

2.2. Conceptual framework

Web pages embedded with Java applets provide an interface for remote operation of the CNC drilling machine as part of the proposed three-tier architecture (Fig. 2). As Java provides platform independence, all that a remote user requires is a Java-enabled web browser and Internet connection to use the system. Using the web interface, the authorized remote clients (first tier) provide machining parameters which are relayed to the web server using the CGI.

The machining parameters upon reaching the server (middle tier) constitute a parts-program which is stored in a job file scheduled for execution on the drilling machine. The CNC drilling machine is a representation of the machining process, which is controlled by the server via the enhanced parallel port.

The machining process is monitored by the network camera, which is part of the middle tier providing real-time feedback to the remote users. Users can further collaborate on product design or other issues by means of a chat room hosted by the web server.

In addressing security issues pertaining to the web-based manufacturing system, the firewall is the first line of defense in restricting unauthorized access to the server and therefore the drilling machine. In addition to this, the web server authenticates users prior to granting permission to submit jobs, monitor the machining process or participate in virtual discussion. Finally all sensitive communication between the clients and the server is carried over TLS thereby ensuring confidentiality, data integrity and server authentication.

The physical setup of the web-based manufacturing system is shown in Fig. 3. The *S*erver for *I*nternet-based *S*ecured *MAN*ufacturing (*sisman*) which hosts the web-based manufacturing system has been configured to run the Red Hat Linux operating system, Apache web server and iptables firewall. *Sisman* has also been registered on USP's domain name server. Furthermore connections originating from outside USPNet destined to port 80 (HTTP) and port 443 (HTTPS) on *sisman* were allowed to pass through USPNet's firewall.

It is worth mentioning that the most influen-

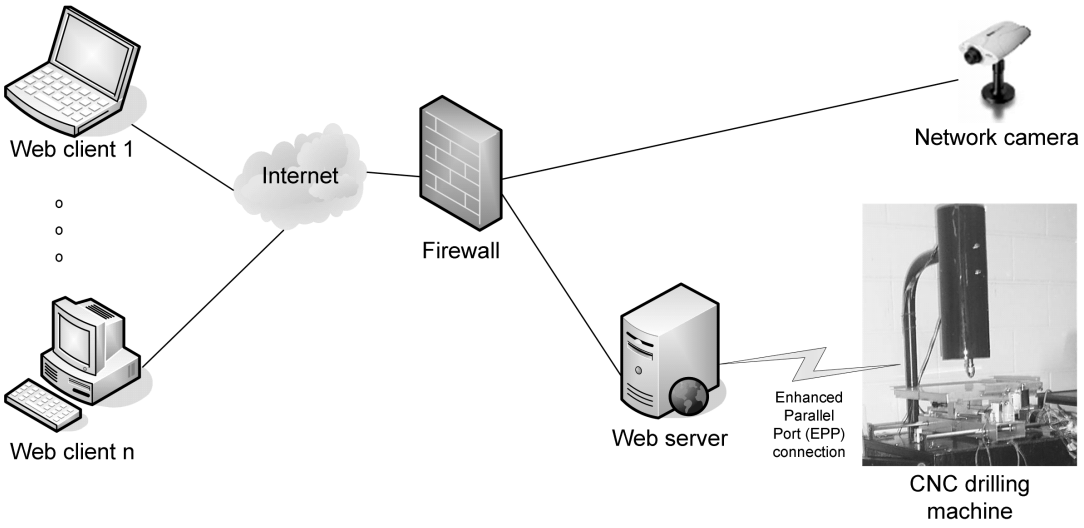

Fig. 2. Conceptual framework outlining the various components (subsystems) and their interaction in the proposed web-based manufacturing system.

tial reason for picking Linux over other operating systems such as Windows XP was the fact that it is open source software. The development of a web-based manufacturing system was like venturing into uncharted waters. As a last resort there was always the possibility of modifying the kernel to get some functionality of the system working.

2.3. Job submission

The process of job submission (Fig. 4) requires coordinated interaction between the client and the server. On the client side is a Java applet (Fig. 5) responsible for reading and validating input machining parameters prior to encoding and sending it over to the server via CGI.

The validation of user input is one of the key operations and also a strong reason favoring the use of a Java applet over an HTML form in user interface design. The application logic for validating user input is present within the applet and therefore unlike the HTML form the applet does not need to send the user input to the server for validation. This saves on additional round trip time to the server each time the user makes a mistake.

Admittedly, user interface designed using HTML form is going to take less time to display on the user's browser compared to a similar interface based on Java applet. This is because of the increased time to download the applet, which is comparatively greater in size than the HTML form and the additional time taken by the JVM to load the classes and start the applet. Using Java archive (JAR) utility to compress applet classes into JAR files and also enabling caching on the browser reduces download time significantly.

The web server receives the machining parameters posted via CGI, sets up the necessary environment variables and executes the CGI script responsible for processing the job submission. The CGI script is responsible for storing the machining parameters in a job file, adding the job file to the scheduling system and then notifying the machining controller about the presence of the job file in the queue. Finally, the user is informed about the status of the job submission by means of a job ID uniquely identifying their job.

The machining parameters are stored in data files as opposed to data structures in the machine controller simply because files remain persistent across power outages. The format in which machining parameters are stored in the job file is shown in Fig. 6.

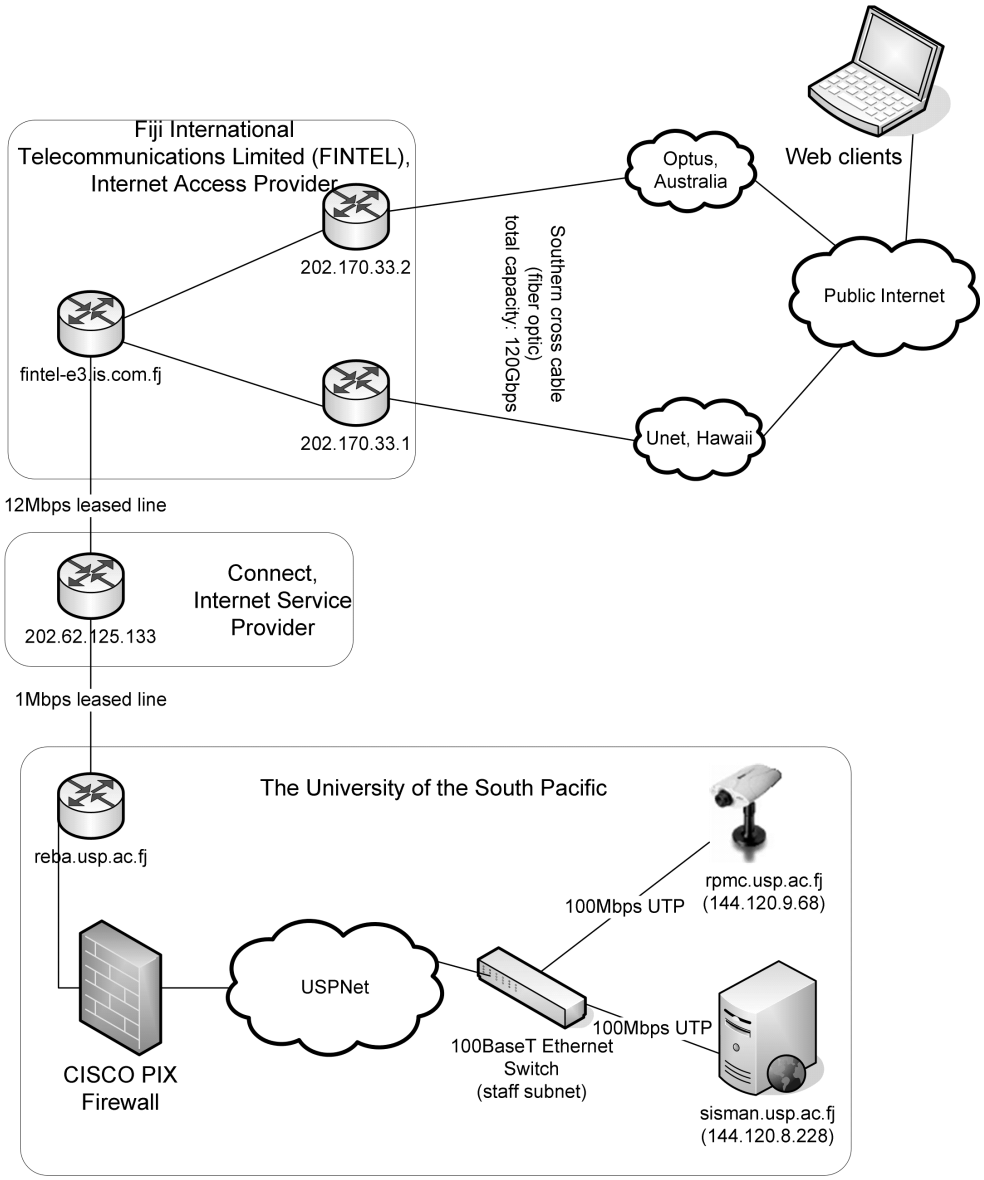

Fig. 3. Network setup used for the project.

2.4. Job scheduling system

For the web-based manufacturing system to get acceptance from users who are likely to be situated in different time zones, fairness in scheduling of jobs has been given the prime consideration. Accordingly the First Come First Serve (FCFS), a non-preemptive scheduling algorithm has been chosen for the scheduling system which has also provisions for administrative users to set priorities in managing urgent jobs.

2.4.1. Job filename and job ID

Given that jobs are stored in files in the same location, the names of files created should be unique. Thus the following file naming conven-

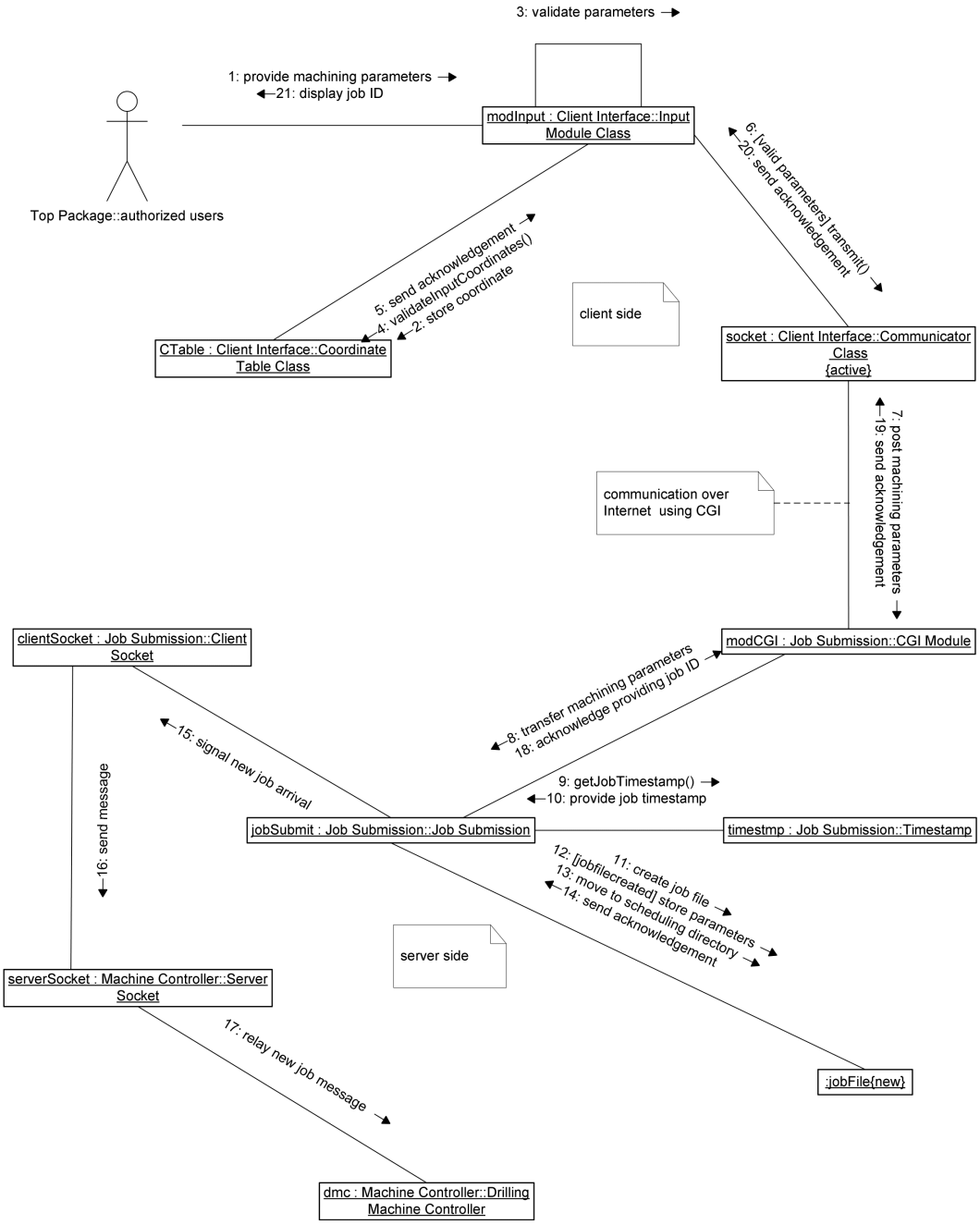

Fig. 4. Collaboration diagram showing the interaction of objects in the web-based manufacturing system to facilitate user's job submission request.

tion has been adopted.

`<priority>_<timestamp>_<6 random characters>`

The optional priority number can range from 00 (highest) to 09 (lowest). Only the systems adminis-

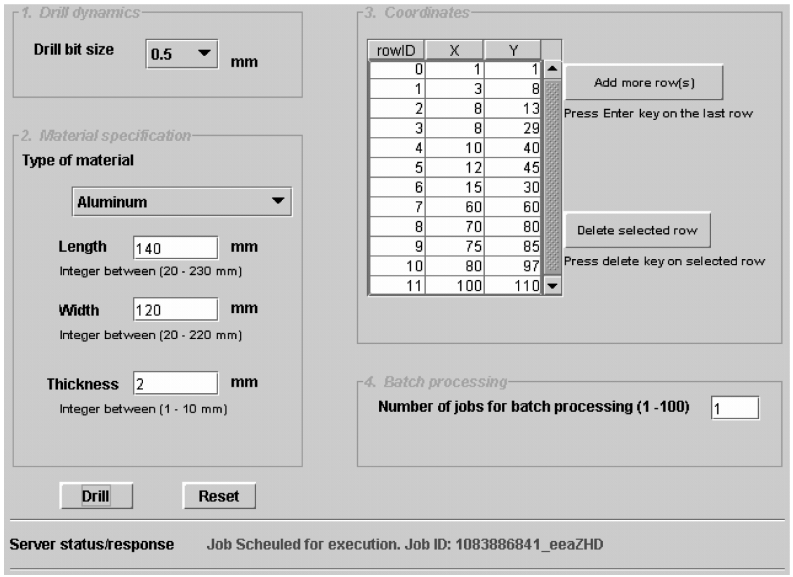

Fig. 5. User interface for job submission.

```
user: admin
bitSize: 0.5
material: 1
length: 35
width: 35
thickness: 2
batch: 1
---COORDINATES---
0 0
3 8
4 4
4 8
5 11
8 13
8 29
```

Fig. 6. Sample format of job file containing machining parameters for execution on the drilling machine.

trator is permitted to rename job files to include the priority information. The timestamp portion of the filename makes it simple to sort files according to the order in which they are to be executed. The random characters ensure jobs submitted simultaneously are unique. Consequently, getting a directory listing of alphabetically sorted job filenames yields the order in which the jobs should be executed as per FCFS algorithm.

2.4.2. Job storage

The storage structure (Fig. 7) for processing jobs consists of four directories under a common parent directory. The rationale for having such a structure is that while the user interface transmits machining parameters and the CGI script stores it in the job file, there is a possibility that the user can terminate the transmission by simply closing the browser. In such cases it is pointless to execute the partially complete job. So now the machining parameters written to the *buffer* directory and only when the job file is complete in its entirety it is moved to the *jobs* directory. Incomplete files are periodically flushed from the *buffer* directory. The *jobs* directory stores all jobs successfully received by the server and waiting execution. The scheduler picks up the next job in queue from the *jobs* directory and places it in the *execution* directory. Since only one job can be executed at any given time, the *execution* directory can have at most one job. The machine controller executes job from the *execution* directory and after completion the job file is moved to the *completed* directory which can be later used for billing the clients.

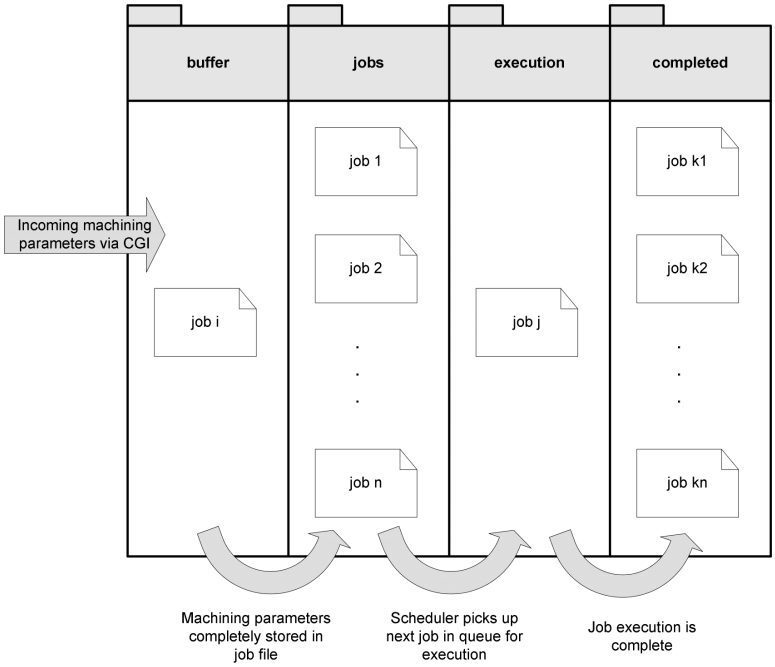

Fig. 7. Directories used for storing job files from initial submission to completion of jobs.

2.5. Job execution

Under the normal scenario, job execution commences once the job file has been placed in the *jobs* directory and the job submission system sends notification about the job file to the machine controller as shown in Fig. 8. The main program which coordinates the interaction between objects in executing a job (Fig. 8) is the job server which runs as a background process (standalone daemon) registered as a system service. Just like any other system service, the job server daemon is easily controllable using standard operating system utilities such as *chkconfig* and *service*.

As mentioned earlier, an advantage of signaling the job server about the arrival of a new job is that loading and subsequent execution of that job can begin straight away. While a job is being executed other new job signals arriving are queued. However there is a serious problem in that if there is a power outage then all the notifications about pending jobs are lost. Upon restart the job server goes back to listening for new job notification when in fact there are pending jobs. Possible solutions include storing the new job notifications in a file, or the job server checking the contents *jobs* directory during startup. However, given that remote users can delete their own job files while it is in queue, the proposed solutions give rise to inconsistency in the state of the job queue perceived by the job server. What is needed is a dynamic way of detecting if jobs are pending even if the notification signal somehow gets lost, thus making the system robust.

The approach we have implemented is in the form of a thread as part of the job server which periodically polls at the *jobs* directory. In an effort to maintain a standard means of providing notification to the job server, the polling thread instantiates the client socket class and through it sends new job notification to the server port on the job server.

2.6. Job management system

After a job has been submitted it may have to wait in the job queue for some time before its turn comes up for execution. In the meanwhile the user submitting the job needs some way of knowing the position of their job in the queue and also whether their job has been completed. Furthermore, there has to be provision that allows users to delete their own job from the queue if they wish to cancel the job. These concerns are addressed as part of the job management interface as shown in Fig. 9.

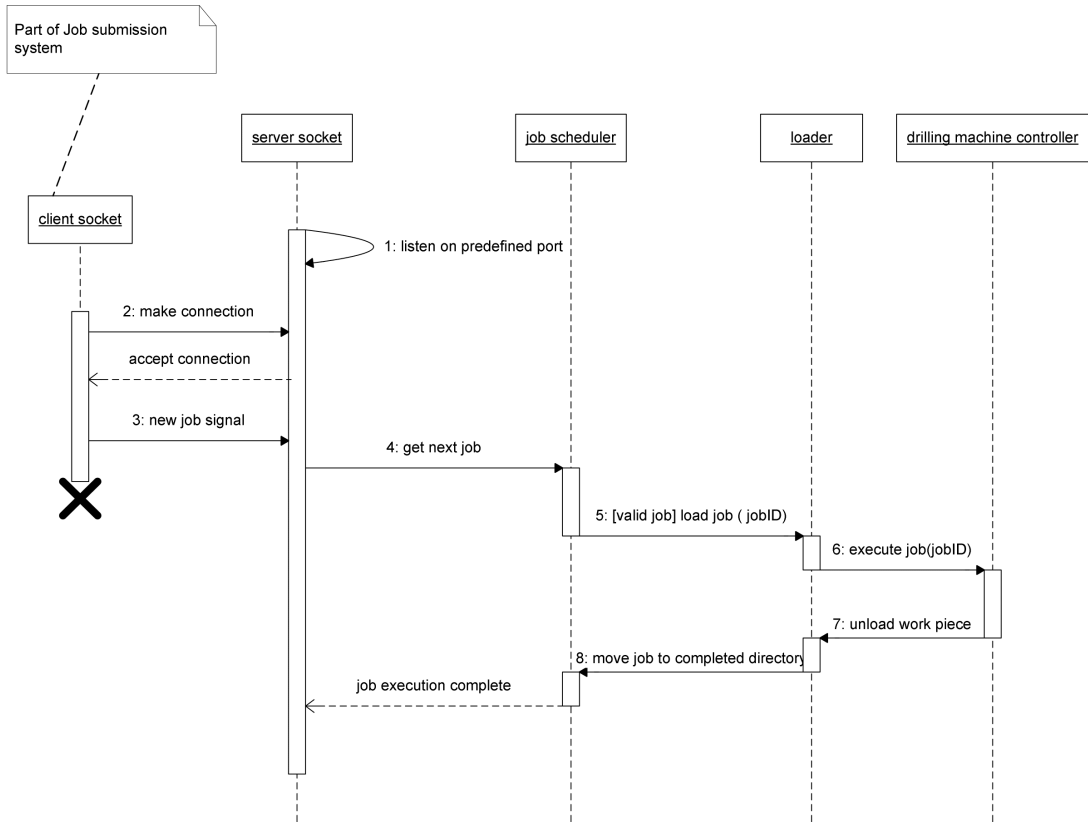

Fig. 8. Sequence diagram for depicting the job execution process.

2.6.1. Job status

The carefully designed job storage structure of the scheduling system seamlessly allows the remote users to check the status of their job at various stages of processing. Generating alphabetically sorted listing of job files in the *jobs* directory produces a report on the job queue (Fig. 10). In the similar fashion, a sorted directory listing of the *completed* directory enables the users to check if their job has been processed. Likewise listing the job file in the *execution* directory indicates the job that is currently being executed.

2.6.2. Job deletion

Deleting a job from the queue is as simple as selecting it from the job queue (Fig. 9) and pressing delete. The job deletion request along with the jobID of the job to be deleted is then forwarded to the server via CGI (Fig. 11). The CGI script responsible for performing deletion checks that the file is a valid one and then opens it. The next step is to validate the user. The identity of the user who requested for job deletion is obtained from the `REMOTE_USER` CGI request meta-variable. The owner of the job to be deleted is recorded within the job file itself (Fig. 6). If the user requesting job deletion is the administrator or the owner of the file then and only then the job file in question is deleted. Otherwise if the job file does not exists or the user requesting deletion is not the owner then suitable error messages are relayed back to the user.

2.7. Remote process monitoring & online collaboration

In an effort to lessen the impact of geographical separation between the remote users and the manufacturing equipment, the web-based manufacturing has provisions for remote process monitoring and virtual discussions between remote designers.

The AXIS 2100 [14] digital TCP/IP network cam-

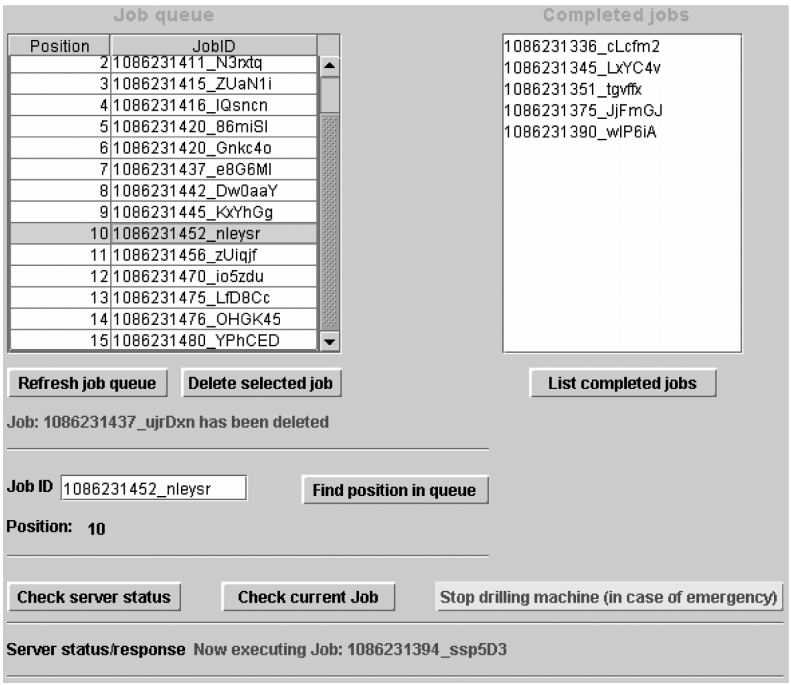

Fig. 9. Java applet-based interface for managing jobs.

era with embedded Linux operating system and built-in Boa web server has been configured to monitor the machining process. The visual feedback from the camera is relayed at a rate of 10 images/s which can be viewed directly from any browser supporting ActiveX or JVM.

In promoting discussions on design or other issues between remote users, a simple chat room application called the virtual discussion room has been developed. The main obstacle in designing a web-based virtual discussion room is that "room" not only needs to keep track of users currently logged in but also has to be able to relay the message posted by one user instantaneous to others over HTTP which is a stateless protocol. A common approach [15] is to have multithreaded chat server which communicates with the chat clients via dynamically assigned port numbers. However the problem with this approach is that without prior knowledge of which port numbers are going to be used by the server thread it becomes difficult to configure a firewall to restrict access to other parts of the server hosting the web-based manufacturing system. Thus the approach we have adopted is based on client initiated refresh scheme whereby the chat server keeps list of connected users as well messages posted by each user as part of the chat history in data files. The chat client periodically refreshes its copy of the message history and the user list providing near real time conversation environment.

3. Network security

The web-based manufacturing system described in the previous section is open to the Internet, which by no means is a safe place. Crackers can by using appropriate tools carry out damaging tasks ranging from web site defacement to stealing or altering of sensitive information. Denning and Baugh [16] highlight numerous cases of Internet related crimes. A key objective of this research was to put in place *adequate* security measures to arrest if not to impede such situations which may compromise the web-based manufacturing system. Undoubtedly there are many facets to security which in itself encompasses various disciplines ranging from cryptography to social engineering. The primary means of accessing the web-based manufacturing system is through the Internet, thus the subject of discussion in this section focuses on network security.

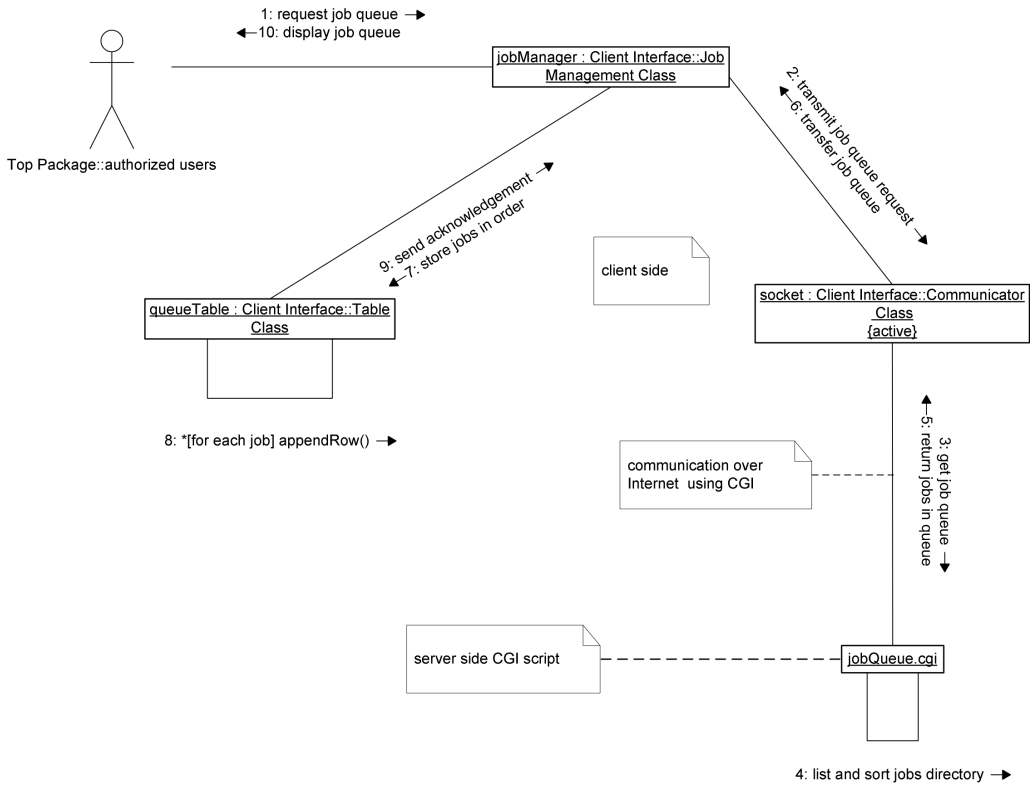

Fig. 10. Collaboration diagram for getting a listing of jobs in the queue.

3.1. Transport Layer Security (TLS)

In a normal HTTP session the data exchanged between the client and the server is transmitted in plaintext over the Internet. For the web-based manufacturing system this means that the machining parameters can be intercepted by malicious users with intention of stealing design information or even worse, the machining parameters can be altered without the knowledge of the client or the server.

Clearly such situations are detrimental and as such the web server has been configured to securely communicate with the authorized clients using the TLS protocol. The TLS protocol [17] uses encryption to promote secrecy of messages exchanged between the client and the server. Using secure hash functions in the computation of message authentication code (MAC), the integrity of the messages exchanged is preserved. Further it allows the communicating parties to mutually authenticate each other by means of digital certificates before communication takes place.

The TLS protocol has a layered architecture which is sandwiched between some reliable transport layer protocol such as TCP and an application layer protocol. The TLS record protocol and TLS handshake protocol are the two important layers of the TLS protocol.

The TLS Handshake protocol encompasses a suite of sub protocols namely change cipher spec protocol, alert protocol and the handshake protocol. The change cipher spec protocol is used for notifying the other party that subsequent records will be protected under newly negotiated encryption and MAC parameters. Alert protocol is used to report error conditions to the communicating parties. The handshake protocol allows the server and the client to authenticate each other and negotiate cryptographic algorithms and keys.

The TLS record protocol is responsible for processing (protecting) records received from a higher level protocol. Processing activities (Fig. 12) includes fragmentation, compression, MAC computation and encryption. The algorithms for these processing ac-

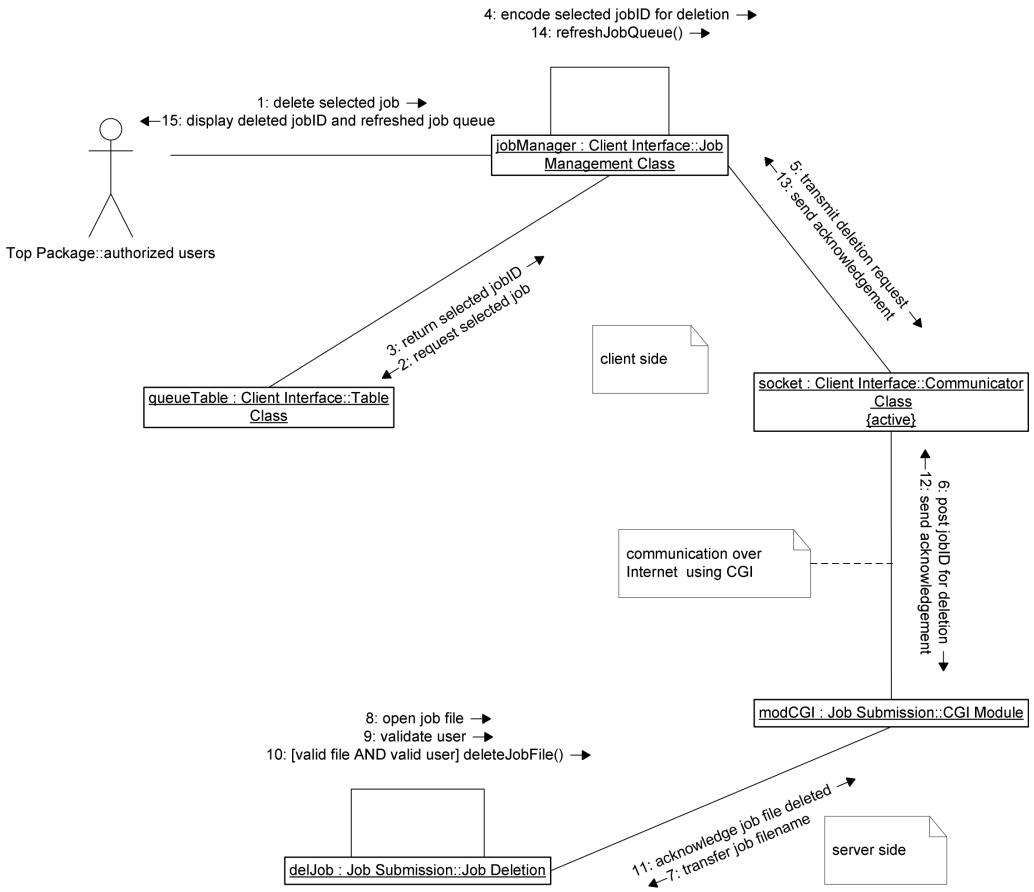

Fig. 11. Collaboration diagram showing interaction between objects in successfully deleting a job requested by the authorized user.

tivities are specified in the operating environment of the TLS Record protocol known as a TLS connection state. The TLS Handshake protocol is responsible for negotiating security parameters between the client and server and initializing the connection state

3.2. Client authentication

Restricting user access to the web server directories housing the web-based manufacturing requires some form of authentication mechanism to be deployed. Though the TLS protocol has provisions for client authentication this option was not chosen for the reason that generating a Certificate Signature Request (CSR) and getting a certification authority (CA) to issue the certificate apart from being costly and time consuming is likely to be beyond average users. Instead a password based authentication scheme is preferred.

The Apache HTTP server provides two authentication modules; namely `mod_auth` for Basic authentication scheme and `mod_auth_digest` for Digest authenticate scheme. The difference between Basic and Digest authentication scheme [18] is that the Basic scheme sends the username and password pair as cleartext whereas in Digest scheme checksum (typically MD5 is used) of the username, password, given nonce value, HTTP method and requested URI is transmitted. The Digest authentication scheme being more secure than the Basic scheme was chosen. It is worth mentioning that both the schemes are vulnerable to the man-in-the-middle attack; however since authentication takes place over a TLS session the vulnerability is eliminated.

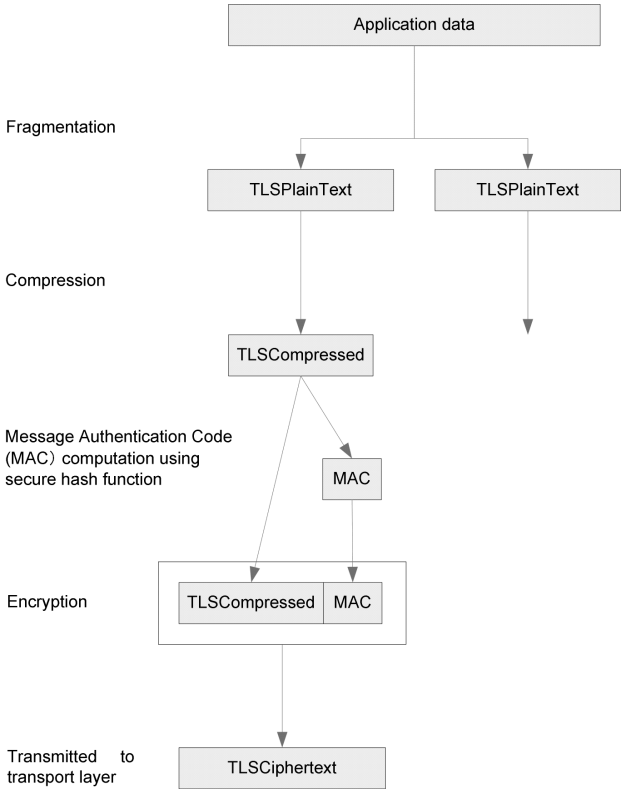

Fig. 12. TLS record processing.

3.3. Firewall

A firewall is essentially a collection of hardware and software which controls the flow of traffic in and out of a network based on certain predetermined rules. We have used the Iptables [19] firewall which is the filtering module part of network filter stack (netfilter) in the Linux kernel, providing powerful features such as connection tracking and advanced filtering rules.

The iptables packet filtering firewall on *sisman* was configured according to the following policies as highlighted in Table **??**. It is worth mentioning that HTTPS stands for secure HTTP where by a TLS connection is established and the default server port 443 is used instead of the standard web server port 80 for all communications.

Members of public can view general (unclassified) information about the project. So a connection from anywhere to port 80 is allowed (rule 1).

Rule 2 allows the authorized user to connect to port 443 in order to access the web-based manufacturing system.

Rule 3 enabled *sisman* to make connections to the outside world. In particular this was used to connect to the Red Hat network for updating the installed packages with security fixes. This rule works by examining the ACK bit in the TCP header. The initial TCP connection establishment packet does not have the ACK bit set whereas packets that are part of an ongoing conversation have the ACK bit set [20]. Therefore packets from outside hosts sent in response to a connection initiated by *sisman* are allowed to pass through the filter. However packets with initial connection request from outside hosts are subjected to the above rules (1-2).

The default policy is to drop packets for any connection which does not match the above rules.

In concluding remarks, the security settings of the web server and the firewall create a synergistic barrier. In the manner in which the web server has been configured, the only way for remote user to login and submit or manage jobs and participate in the virtual discussions is via HTTPS which requires a connec-

Table 1
Firewall configuration rules

Rule	Action	Source	Port	Destination	Port(s)	Flags	Comment
1	Allow	*	*	sisman.usp.ac.fj	80		Incoming HTTP connection from anywhere
2	Allow	pc0530.usp.ac.fj	*	siman.usp.ac.fj	443		Incoming HTTPS connection from the authorized user's computer
3	Allow	*	*	sisman.usp.ac.fj	*	ACK	Allow packets from established or related connection originated by *sisman*
4	Block	*	*	*	*		Drop all other packets which do not meet above rules

tion to be initiated by the client to port 443 of the server. For someone to access the secure web server directory, a password is needed. However the server only accepts passwords via HTTPS. Unless and until that someone has been specifically given access to make connection to port 443 they will not even get to see the login prompt let alone attempt a brute force attack in cracking the password.

4. Distributed Architecture

In its present form the framework for the web-based manufacturing system supports the CNC drilling machine. However it is worth reiterating that the architectural design is generic and modular in nature and as such any other genre of CNC machine such milling and turning could be substituted in place of drilling with only machine-specific changes. Moreover in a commercial environment there may be strong need to support multiple types of machining processes, thus the current conceptual framework can be extended to facilitate this as shown in Fig. 13.

As part of the extended framework the web server handles job submissions and also hosts the virtual discussion room. The jobs are stored in storage directories resident in physical job servers running the separate job server daemons. The machine controller component of the job server daemon will naturally be different for different machines along with some modifications to the existing user interface.

The notion of distributing the major subsystems of the web-based manufacturing system on physical machines or nodes balances load which otherwise would have been placed on a single node. Furthermore in the distributed setup the job server nodes operate independent of each other therefore if the job server for the drilling process goes down then milling and turning operations are unaffected. To increase the availability and reliability of the system even more, multiple web servers can be deployed in parallel.

5. Conclusion

Internet-based teleoperation of manufacturing systems is not a new concept as is demonstrated by the variety of such applications springing up on the Internet. These applications emanating from different ideologies have varied focal points in addressing different issues. For instance, none of the research discussed in the literature review provide detailed consideration of security issues in communicating over the Internet. This paper unifies the design considerations such as capturing global audience, handling data loss and out-of-order delivery of data, coordinating multi-user access, reducing susceptibility of temporal communication delays, and security into a single framework. This framework has been implemented in the form of web-based manufacturing system using free open source software thus providing cost effective platform independent means of teleoperating a CNC drilling process. More importantly the system has a modular architecture and hence can be extended to support other genre of CNC machining processes.

Moving the discussions to the virtual enterprise setup where let us say there are m remotely operable manufacturing systems shared by n geographically dispersed collaborating partners (manufacturing nodes). In such a setup the manufacturing nodes can be located in any time zone and as such it is critical for the manufacturing systems to be available online at all times. The drilling machine setup in its current form does not meet this criterion because of lack of

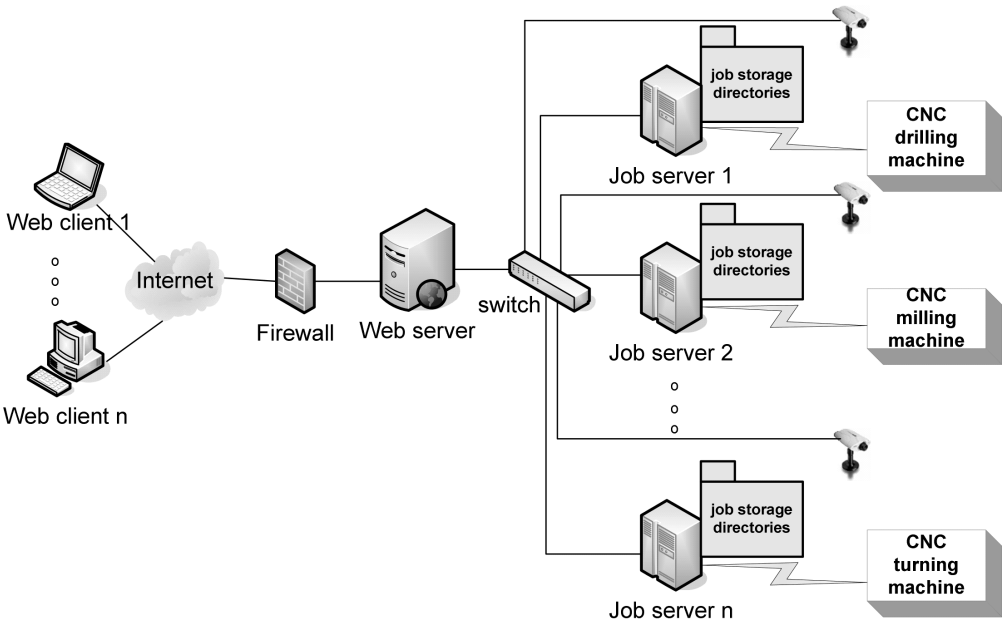

Fig. 13. Extended framework for supporting multiple manufacturing processes.

provisions for unattended machining. In order to facilitate this, further work needs to be undertaken in developing an automated loading and unloading unit and integrating it with the existing setup.

Finally, in development of the web-based manufacturing system, the focus was on fabrication of parts such as printed circuit boards using drilling process which is quite straightforward. Future work to integrate fabrication of complex shapes and forms which include combinations of drilling, milling and turning with the web-based manufacturing system can improve on the existing system by incorporating process simulation and planning.

Acknowledgements

This research has been funded by the School of Pure and Applied Science Research Committee, University of the South Pacific, under Grant No. 6C028-1341-71141.

REFERENCES

1. Jagdev HS, Browne J. The extended enterprise—a context for manufacturing. Prod Plann Control 1998;9(3):216-29.
2. Shi J, Huang GQ, Mak KL. Web-based design for assembly. In: Proceedings of the third annual international conference on industrial engineering theories, applications and practice, Honk Kong, December 1998.
3. Chen L, Bender P, Renton P, El-Wardany T. Integrated virtual manufacturing systems for process optimisation and monitoring. Ann CIRP Spain 2002;51(1).
4. Renton P, Bender P, Veldhuis S, Renton D, Elbestawi MA, Teltz R, et al. Internet-based manufacturing process optimization and monitoring system. In: Proceedings of the IEEE international conference on robotics and automation (ICRA02), Washington, DC, May 2002.
5. Kong SH, Park J, Han Y, Kim G, Lee K. The Internet-based virtual machining system using CORBA. Integrated Manuf Syst 2002;13(5):340-4.
6. Ahn SH, Sundarajan V, Smith C, Kannan B, D ' Souza R, Sun G, et al. CyberCut: an Internet-based CAD/CAM system. J Comput Inform Sci Eng 2001;1:52-9.
7. Tay FEH, Khanal YP, Kwong KK, Tan KC. Distributed rapid prototyping—a framework for Internet prototyping and manufacturing. Integrated Manuf Syst 2001;12(6):409-15.
8. Lee RS, Tsai JP, Lee JN, Kao YC, Lin GCI,

Lu TF. Collaborative virtual cutting verification and remote robot machining through the Internet. Proc Inst Mech Eng 2000;214B:635-44.
9. Onwubolu GC, Aborhey S, Singh R, Reddy H, Prasad M, Kumar S, et al. Development of a PC-based computer numerical control drilling machine. Proc Inst Mech Eng 2002;216B:1509-15.
10. Lal SP. Internet-based secured remote control of computer numerically controlled drilling machine. MSc dissertation, University of South Pacific, Suva, 2004.
11. Dalton B. Techniques for web telerobotics. PhD dissertation, University of Western Australia, Perth, 2001. p. 85-6.
12. Luo RC, Su KL, Shen SH, Tsai KH. Network intelligent robots through the Internet: issues and opportunities. Proc IEEE 2003;91(3):371-81.
13. Robinson D, Coar KAL. The Common Gateway Interface (CGI), version 1.1. Internet-draft (work in progress) 2003.
14. Axis 2100 user's guide. 2001. Axis Communications AB. URL: http://www2.axis.com/files/userguide/2100/2100ug.pdf, February 2005.
15. McCain J. Java chat room. 2000; URL: http://www.worldofjon.com/ java/java4.html, February 2005.
16. Denning DE, Baugh WE. Encryption and evolving technologies as tools of organized crime and terrorism. National Strategy Information Center 's US Working Group on Organized Crime (WGOC); 1997.
17. Dierks T, Allen C. The TLS protocol version 1.0. RFC 2246, 1999.
18. Franks J, Baker PH, Hostetler J, Lawrence S, Leach P, Luotonen A, et al. HTTP authentication: basic and digest access authentication. RFC 2069, 1999.
19. RH253 Red Hat linux networking and security administration. Red Hat Inc., 2002.
20. Cheswick WR, Bellovin SM. Firewalls and Internet security: repelling the wily hacker. New York: Addison-Wesley; 1994.