E-manufacturing system: from design to performance evaluation

Sunil Pranit Lal*

Complex System Laboratory,
Department of Information Engineering,
Faculty of Engineering, University of the Ryukyus,
1 Senbaru, Nishihara, Okinawa 903-0213, Japan
Fax: +81-98-895-8727
E-mail: sunil@eva.ie.u-ryukyu.ac.jp
*Corresponding author

Godfrey C. Onwubolu

School of Engineering and Physics,
Faculty of Science and Technology,
University of the South Pacific, Private Mail Bag,
Laucala Campus, Suva, Fiji Islands
Fax: +679-32-31538
E-mail: onwubolu@gmail.com

Abstract: This paper documents the application development stages of an internet-based manufacturing system touching on issues ranging from motivation to system response. Preliminary investigations were carried out to get an overview of internet connections in terms of loss rate and temporal communication delays. These initial results lead to identification of the key requirements, which were factored into the final design, subsequently implemented as the three tiered web-based manufacturing system. The e-manufacturing system developed features platform independent and cost effective means of securely tele-operating a three-axis computer numerically controlled drilling machine, as well as providing an infrastructure to allow remote manufacturers to collaborate on product design and monitor the machining process in real time. Experimentations on the system performance indicated that the system provides fairness in handling multiple possibly concurrent users, and is highly responsive in terms of negligible processing delays as well as respectable submission job times largely dependent on remote users' bandwidth.

Keywords: web-based manufacturing; computer numerically controlled machine; CNC; tele-operation; distributed system; network security; performance evaluation.

Reference to this paper should be made as follows: Lal, S.P. and Onwubolu, G.C. (xxxx) 'E-manufacturing system: from design to performance evaluation', *Int. J. Internet Manufacturing and Services*, Vol. X, No. Y, pp.000–000.

Biographical notes: Sunil Lal is a PhD candidate at the University of the Ryukyus, Japan. He received BSc and MSc in Computing Science from the University of the South Pacific, Fiji. His research interests include machine learning, adaptive control and robotics. He is a member of Robotics Society of Japan, Society of Instrument and Control Engineers, IEEE Computational Intelligence Society and Swedish Artificial Intelligence Society.

Godfrey Onwubolu is a Professor of Engineering in School of Engineering and Physics at the University of the South Pacific, Fiji. He holds a BEng from the University of Benin, MSc and PhD from the University of Aston, UK. His research interests include mechatronics, modern manufacturing, modern optimisation, inductive modelling and data mining. He has published three books, and over 120 articles in international journals and conference proceedings. As a Chartered Engineer and a Chartered Member British Computer Society, he is also a Senior Member of the Institute of Industrial Engineers, and a Senior Member of the American Society of Manufacturing Engineers.

1 Introduction

The advent of communication networks, in particular the internet, has linked societies in so many ways and has redefined the ways in which people go about with their daily lives. Ranging from e-commerce sites for online shopping and auctioning to social networking platforms such as MySpace and YouTube, the vast possibilities of applications springing up on the internet seem to be only limited by one's imagination.

Fuelled by the growth of the internet, and faced with challenges of producing cost-effective products under strict time frame and quality control, the manufacturing industries have evolved from traditional self-contained style of manufacturing to a global customer-oriented approach giving rise to the concepts such as virtual enterprise (Jagdev and Browne, 1998). The latter approach looks at ways of bringing together the geographically separated human resource personnel and manufacturing equipment. The first step towards diminishing of this geographical separation is incorporating the possibility in a manufacturing equipment to be remotely controlled.

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 resource 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 web-based simulation for analysis of product design (Shi et al., 1998). Other attempts have gone as far as development of virtual environment for the simulation, optimisation and generation of prototype via the internet (Chen et al., 2002; Renton et al., 2002). Along similar line is the internet-based virtual manufacturing system using CORBA, which enables simulation of analysis, design and experimentations to take place prior to the actual production on the machining centre (Kong et al., 2002). CryberCut (Ahn et al., 2001) is a praiseworthy effort, 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 numerically controlled (CNC) milling machine.

Internet-based tele-operation 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. Our approach (Lal, 2004) 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 a cost-effective, platform independent means of tele-operating a PC-based CNC drilling machine (Onwubolu et al., 2002), which was developed in-house. More importantly, the system has a modular architecture and hence can be extended to support other genre of CNC machining processes.

2 Exploring the network

The journey of the data packets over the internet is filled with uncertainty about possible loss and the end-to-end delay as result of dynamic routing, queuing at routers and other factors such as transmission delays (Kurose and Ross, 2001). The effects of uncertain temporal delays and data loss in communication over the internet can seriously jeopardise the effectiveness of remote control applications, thus these characteristics are presented in this section to gain a better understanding towards developing an effective Internet application.

2.1 Packet dynamics

Communication networks are fundamentally based either on circuit switching or on packet switching (Kurose and Ross, 2001). In circuit switched networks, e.g., telephone networks, the resources between the communicating end systems are reserved for the duration of session, thus providing guarantees for the quality of service in terms of bandwidth allocation and time delays. For a packet switched network such as the internet, application messages exchanged between the communicating parties are decomposed into packets of data bits before being transmitted over links interconnected by packet switches (routers). Packet switching is widely used as it facilitates the interconnection of networks with different architectures; it provides flexibility in resource allocation and reliability against node and link failures (Clark, 1988). Since most packet switched networks do not reserve network resources, there is usually no guarantee if and when a packet reaches its destination.

When a host injects packets of data into the internet, they traverse along the communication link and through the routers, which decide the next hop the packet should take to reach its destination. The routers take into account the load on the interconnected networks in deciding which outgoing link to use in forwarding the packet. It is therefore not uncommon for packets originating from a source to travel through different routes before reaching the same destination, possibly out-of-order. Very often, the routers are busy processing other packets and as such, the packet has to wait in a queue. Sometimes, the queuing buffer on the router is full and the packet is discarded. Other times, the packet arrives after the expiry of its time to live (TTL) parameter (Postel, 1981) and it gets destroyed regardless. Eventually, when the packet does reach the destination host, it would not be surprising if it has been damaged as result of bit flips or noise during transmission.

2.2 Experimental setup

Given the heterogeneity of the internet and many dynamic factors influencing the flow of messages between communicating entities, it is difficult to use analytical models to study the characteristics of the internet in terms of temporal delays and data loss rates (Paxson and Floyd, 1997), thus experimental approaches (Oboe and Fiorini, 1997; Bolot, 1993) are commonly used instead.

Mills (1983) describes experiments using internet control message protocol (ICMP) to study roundtrip transit delays over the internet with respect to packet length and the effectiveness of TCP retransmission-timeout algorithm. Sanghi et al. (1993) characterise network behaviour in terms of roundtrip time (RTT), losses, reordering and duplicates using user datagram protocol (UDP) datagram transmitted and echoed between hosts located in Maryland, Stanford and MIT in the USA. Bolot (1993) provides analysis of packet delay and loss, using roundtrip journey of UDP datagram between hosts located in France and USA. Oboe and Fiorini (1997) provide an approximate model of the internet characterised by the statistics of the packet delay and loss obtained using ICMP within Italy, and between Italy and the USA.

The experiments described in the literature were carried out decades ago and mostly over internet connections in the northern hemisphere centering on the USA. It may not be reasonable to extrapolate some of the results to the present time and location of this research due to rapid topological changes to the internet over these years, thus the need for renewed experimentation based on methods presented in the literature.

World Wide Web (WWW) servers of universities from different continents were chosen as part of this experimentation. An important selection criterion was for these servers to respond to our ICMP echo request datagram sent by ping and trace route utilities (Enger and Reynolds, 1993), which were used for the experiment. Ping measures the RTT and packet loss rate, whereas trace route provides details about the hops traversed by the packet en-route to the destination. RTT, which is the time it takes for a packet to reach its destination and return back to the sender effectively captures delays introduced as a result of asymmetric paths, typical in the internet due to dynamic routing. Note that transmission control protocol (TCP) is not used in carrying out experiments of this kind because the sender does not have complete control over its sending rate as this is governed by window-based flow control and congestion control mechanisms implemented in the operating system. Furthermore, TCP makes uses of acknowledgements and retransmissions to recover from damaged or lost packets and therefore the purpose of studying packet loss characteristics is defeated. ICMP and UDP unlike TCP do not have mechanisms for reliable data transmission and as such have minimal overhead and are therefore ideal choice.

The experiment was conducted between Suva, Fiji and seven selected locations on different continents (Africa, Australia, Colombia, Fiji, India, UK and USA) by sending 1000 ICMP echo datagram (probes) for each and every trial undertaken. The size of the datagram was varied between 32, 64, 96, 128 and 160 bytes, to show the effects of data size on the RTT. The intervals between successive ICMP echo requests, δ were varied between 1, 10, 50, 100, 200 and 500 ms to study its effects on the loss rate. All in all, 210 different trials were conducted lasting around ten hours. A selected subset of the entire experimentation was repeated at the same time (around 1000 hrs FJT) for different days of the week to monitor the weekly variation in RTTs. The trials were conducted sequentially so that traffic from one trial does not affect the result of the other.

2.3 Results and observations

Varying δ affects the amount of 'stress' induced on the network; hence it offers a possibility to observe the behaviour of the test connection under various degrees of congestion. Smaller values (1 ms, 10 ms) of δ induces greater stress as there is increased flow of closely 'spaced' probes which cause flooding effect on the link.

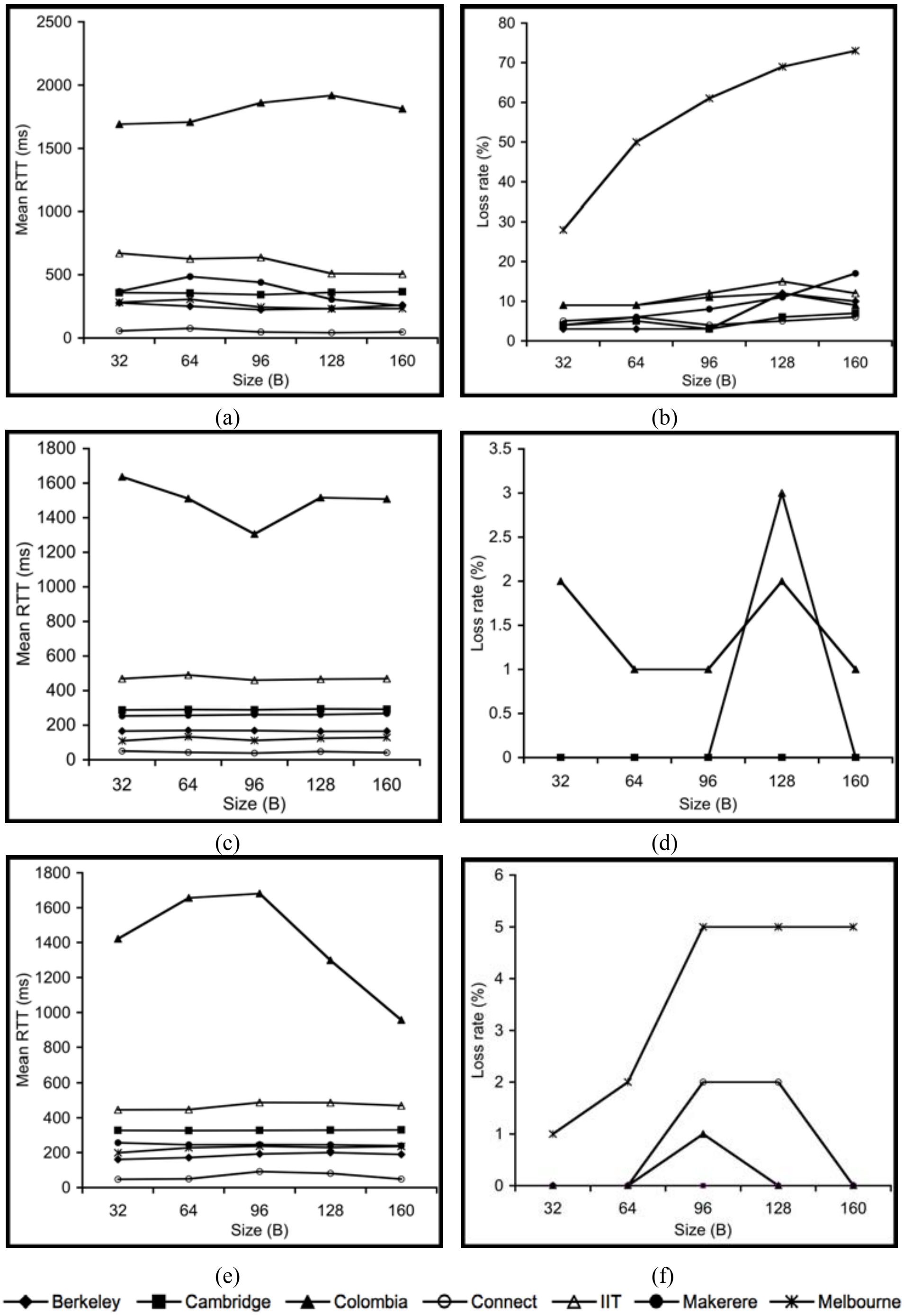
2.3.1 Trends in loss rate

It was observed that for low δ values, there is a strong positive correlation between the size of probe and the probability of loss for most of the connections [Figure 1(b)], for example, correlation of 0.97 was observed for Makerere. This can be explained based on the fact that transmission delay is a function of packet size and as such larger packets would take longer to transmit on the outbound link of the intermediate router. Packets already in the queue would have to wait slightly longer and those that arrive in a very short time interval are unlikely to find space in the limited queue buffer and therefore end up as statistics of loss rate.

As probing interval is increased, the loss rate for all connections decreases steadily and more or less settles within 0%–5% margin [Figures 1(d) and 1(f)]. Moreover, the correlation between size of probe and the loss rate diminishes and in most cases, it becomes zero. Nature of loss tends to become random. This is to be expected as increasing δ relives the load imposed by the probes on the network and therefore whatever small loss is witnessed, can among other factors, and be attributed to the interactions of other users on the internet.

Finally, apart from observing increased loss rate when testing was conducted under induced stress on the network, increase in out-of-order delivery of probe packets was also significant. For instance, in the test conducted on the link to Melbourne for probe size = 32B and δ = 1 ms, 8.8% of the probes arrived out-of-order. This strongly indicates fluctuations in routing path as the network tries to cope with the increased congestion levels. For large δ values, out-of-order delivery tends to be negligible.

Figure 1 Effects of packet size on mean RTT (a, c, e) and loss rate (b, d, f) by varying probing interval on the selected servers, (a) RTT vs probe size for δ = 1 ms (b) loss rate vs probe size for δ = 1 ms (c) RTT vs probe size for δ = 100 ms (d) loss rate vs probe size for δ = 100 ms (e) RTT vs probe size for δ = 500 ms (f) loss rate vs probe size for δ = 500 ms



2.3.2 Trends in mean RTT delay

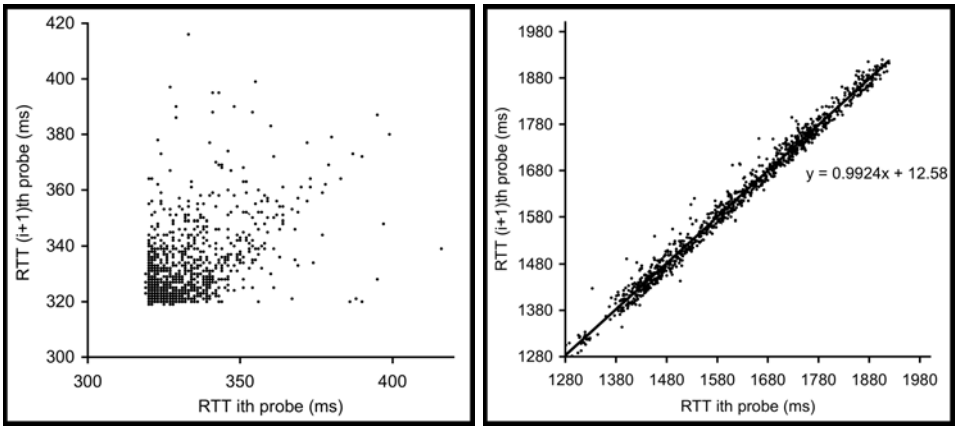
Transmission delay is one of the constituents of RTT delay and as such, the size of packets transmitted, in theory should have a bearing on the RTT experienced. In other words, larger packet size translates into longer RTT. Table 1 shows the correlation between the probe size and mean RTT for the two extreme values of δ namely 1 ms and 500 ms. As can be noted, δ = 500 ms shows better and in some cases, strong correlation compared with δ = 1 ms. The lack of much positive correlation for low δ gives an indication that queuing delay plays a dominant role in determining the RTT for congested networks.

Table 1 Correlation between probe size and RTT delay for δ = 1 ms and 500 ms

Test site	*Server name*	*Correlation*	
		δ	
		1 ms	*500 ms*
Berkeley	www.berkeley.edu	–0.40	0.84
Cambridge	www.cam.ac.uk	0.31	0.88
Colombia	www.unalmed.edu.co	0.74	–0.69
Connect	www.connect.com.fj	–0.60	0.25
IIT	www.iitd.ac.in	–0.92	0.67
Makerere	www.makerere.ac.ug	–0.68	–0.85
Melbourne	www.unimelb.edu.au	–0.82	0.75

Another notable trend is that as δ is increased mean RTT for most of the connections decreases. This is to be expected as lower congestion levels means lesser influence of queuing delay on RTT. The congestion levels can be observed on phase plots. In a phase plot (Bolot, 1993), a marker is placed at coordinate (x, y) if there exist a value n such that $x = rtt_n$ and $y = rtt_{n+1}$. The phase plot for Cambridge [Figure 2(a)] with test parameters of δ = 10 ms and probe size = 64B represents the typical distribution for non-congested communication channel with $rtt_{n+1} = rtt_n + \varepsilon_n$; where ε_n is a random process with mean zero and low variance (Bolot, 1993). This distribution is observed through out all trials with Cambridge, thus confirming the relative stability of the communication channel in terms of consistent mean RTT and loss rate depicted in Figure 1. The phase plot for Colombia [Figure 2(b)] with test parameters of δ = 10 ms and probe size = 64B is representative of a congested communication channel, thus shedding some light on reasons for relatively high mean RTTs observed in Figure 1. Note for this plot the points are centered around the line $rtt_{n+1} = rtt_n + P/\mu - \delta$, where P is the packet throughput and μ is the channel throughput (Bolot, 1993). From the y-intercept, the channel throughput for connection to Colombian test site can be computed which comes to 22 kbps.

Figure 2 Phase plots for connections to, (a) Cambridge (b) Colombia using probe size of 64B and δ = 10 ms



(a) (b)

2.3.3 Summary

The outcome of experiments described herein provides enriching insight into the behaviour of the internet, which can be characterised in term of mean RTT delay and loss rate. When congestion levels are high in the network, the loss rate and number of packets delivered out-of-order become too significant to be ignored. Furthermore, queuing delays show dominant presence in the round trip time experienced by packets.

In order to be effective and more importantly be considered safe, the remote control architecture should be so designed that it is able to recover from out-of-order data delivery and data loss. There are two ways in which this can be achieved:

1 incorporating reliable data transfer mechanisms as part of the application being designed

2 using a transport layer protocol, which provides reliable data transfer.

In choosing the suitable option, a trade-off is involved. Building application on top of some reliable transport layer such as TCP, which is provided by the network stack of the operating system, requires no extra effort in dealing with out-of-order data delivery and data loss. On the other hand, incorporating reliable data transfer mechanisms in the application running over some unreliable transport layer such as UDP offers the application greater control over transmission rate and message size.

On the basis of the results, there does not appear to be an 'optimum' packet size which ensures minimum RTT across different conditions. Therefore, instead of setting a fixed message size, which bundles together whatever control signals that need to be sent, variable sized messages can be used for transmitting independent control messages according to sensitivity of the message. Critical control messages such as EMERGENCY STOP can be coined in between 32 bytes–64 bytes (including headers), which provides reasonably good time for delivery, is less likely to get lost, and is very highly unlikely to get fragmented in transit. Less time critical messages like the coordinates for drilling can be sent in larger packets.

3 Design considerations

Based on the research work of Lee et al. (2000), 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 shortcomings as well as the experimental results from the previous section lead to identification of the following key requirements, which were factored into the final design of the remote control architecture.

3.1 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 tele-operation is bound to use unprivileged ports for communication, which is not favoured as such ports are typically blocked by network administrators. Network tests carried by Dalton (2001) 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 WWW 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 WWW provides the ability to reach the widest audience in terms of cost, convenience and platform independent means of access.

3.2 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 (Luo et al., 2003).

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.

3.3 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). 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.

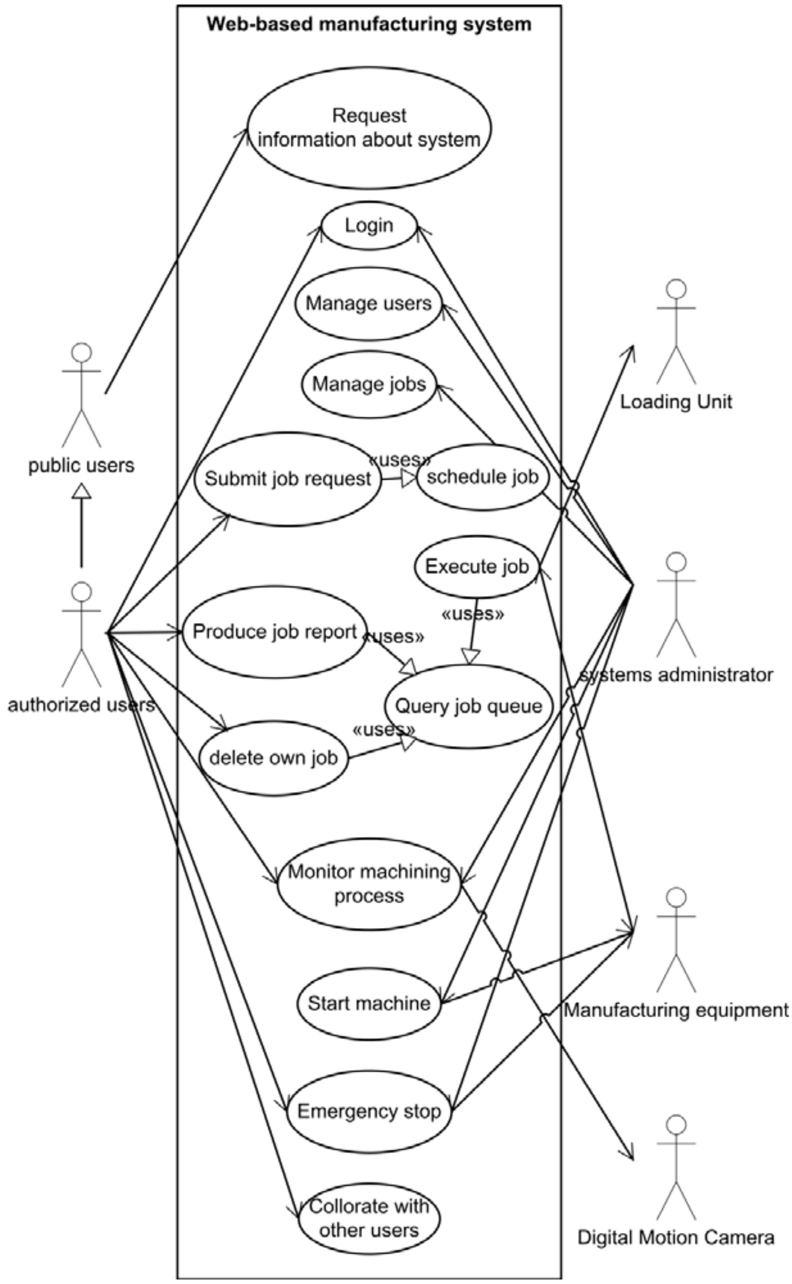
3.4 Reduced susceptibility to temporal communication delays

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. (2003). 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.

4 The three tiered web-based manufacturing system

In this section, the web-based manufacturing system developed is presented. The system enables remote operation of the machining process, real-time remote process monitoring and support for collaborative discussions among remote designers. In doing so, the system addresses issues raised in the previous section in regards to coordination of multiple user access, and dealing with network constraints, as well as provision of adequate security measures. The requirements model, which sets the foundation for design and implementation stages, is captured in Figure 3. Interested readers are directed to Lal and Onwubolu (2007), which documents the system development in greater detail.

Figure 3 Use case diagram shows the interaction between the actors and the web-based manufacturing system



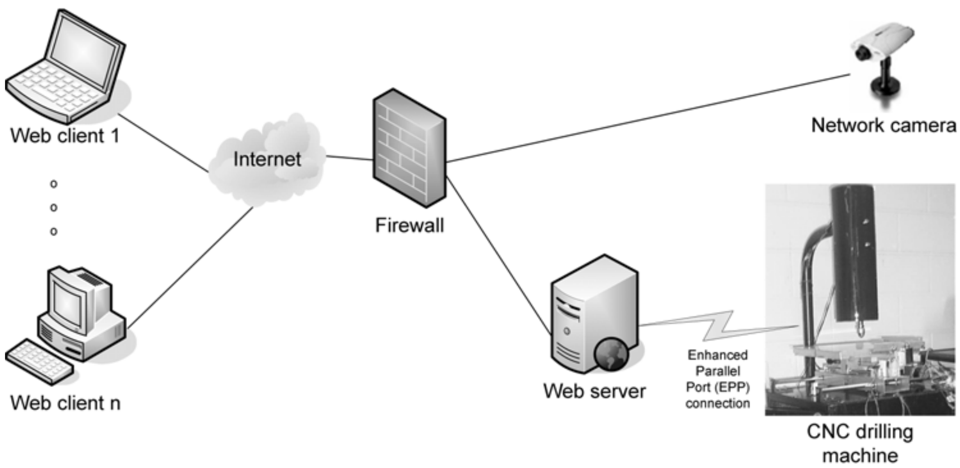
4.1 Conceptual framework

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 (Figure 4). As Java provides platform independence, all a remote user requires is a Java-enabled web browser and internet connection to use the system. Using the web interface, the authorised remote clients (first tier) provide machining parameters, which are relayed, to the web server using 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. It is worth mentioning that the architectural design of the e-manufacturing system is generic and modular in nature. Therefore, any other genres of CNC machine such milling and turning could be substituted in place of drilling with only machine-specific changes to the system.

The machining process is monitored by the network camera, which is part of the middle tier providing real time feedback to the remote users. This lessens the impact of geographical separation between the remote users and the manufacturing equipment. Users can further collaborate on product design or other issues by means of a virtual discussion room hosted by the web server.

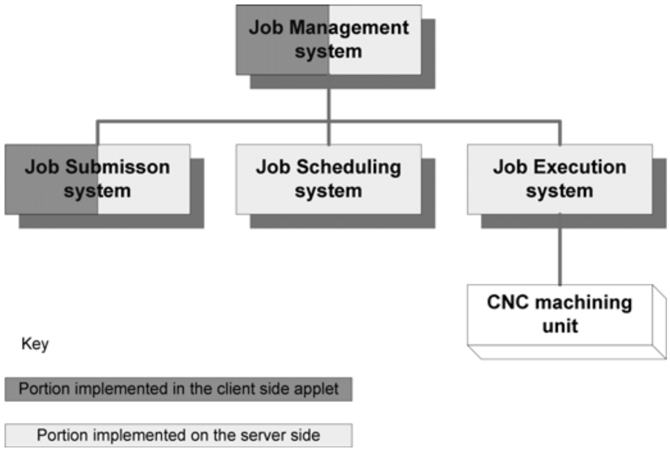
Figure 4 Conceptual framework outlining the various components in the proposed web-based manufacturing system



4.2 System outline

The web-based manufacturing system has a modular architecture as shown in Figure 5.

Figure 5 Modular architecture of the web-based manufacturing system



4.2.1 Job submission system

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

On the server side is a CGI script, which is executed by the web server on behalf of the remote user to process 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. 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. Finally, the user is informed about the status of the job submission by means of a job ID uniquely identifying their job.

After adding the job file to the scheduling system, the machine controller process needs to be made aware of the existence of the job file. There are two ways in which this can be achieved. The first option is for the controller to periodically poll the directory storing jobs. However, this option is not favourable as discovery of new jobs would take place at discrete intervals and as a consequence there is a high possibility of jobs waiting unnecessarily long in the queue. The second, much better option is to have an interrupt driven system, which enables the job submission system to signal the controller whenever a new job arrives so that execution of that job can begin straight away.

For the job submission system to notify the controller, inter process communication is required, which can be implemented using named pipes, a signal handling facility or socket communication. However, unlike named pipes and the signal handling mechanism, socket communication does not impose restrictions on the communicating processes to be placed on the same host. This allows for the possibility of forming a distributed environment, where by one node hosts the central job submission process and several other nodes on the network hosts machine controller processes for different types of machine being controlled. This approach would be instrumental in balancing load across the nodes as part of the machining centre. Thus, socket communication was used to facilitate the interaction between job submission system and job scheduling system.

4.2.2 Job scheduling system

The scheduling system developed is simple yet effective in coordinating requests from multiple users. Unlike preemptive scheduling algorithms used by modern day operating systems in scheduling processes to use the CPU, scheduling jobs on the drilling machine is very much like scheduling jobs on a network printer. Only non-preemptive scheduling algorithms are applicable; that is, once a job starts executing on the drilling machine, it should be allowed to complete before the next job is loaded. Under the class of non-preemptive scheduling algorithms first come first serve (FCFS) and shortest job first (SJF) are the classical ones (Tanenbaum, 2001).

SJF schedules the job, which takes the shortest amount of time to complete first. It requires prior knowledge about the expected completion time of jobs. In comparison with FCFS, SJF algorithm no doubt minimises average turnaround time even when its performance is compromised due to the sporadic nature of job submissions made by the internet users. However, considering the fact that users from different time zones would be submitting jobs, for the web-based manufacturing system to be acceptable, fairness in scheduling of jobs is the prime consideration. SJF policy tends to discriminate against large jobs and as such, FCFS scheduling is the preferred option currently used with provisions for administrative users to set priorities in managing urgent jobs.

4.2.3 Job execution system

Under the normal scenario, job execution commences once the job submission system sends notification about the job file to the machine controller. The drilling machine controller is responsible for interpreting the contents of the job file and transmitting appropriate control signals to the drilling machine connected via enhanced parallel port.

The main program, which coordinates the interaction between objects in executing a job, is known as the job server. The job server runs as a background process (standalone daemon) registered as a system service, which makes it easily controllable using standard operating system utilities. This approach is instrumental in the realisation of *start machine* and *emergency stop* use cases (Figure 3).

4.2.4 Job management system

After a job has been submitted, it may have to wait in the job queue for sometime 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. The integrated job management system seamlessly allows the remote users to check the status of their job at various stages of processing and to remove their jobs from the queue provided they are authorised to do so. In addition, it enables the users to check the status of the job server as well as to stop the drilling machine in case of emergency.

4.3 Network security

Establishing an e-manufacturing system on the internet can be subject to numerous types of threats (Furnell, 2005). Crackers can, by using appropriate tools, carry out damaging tasks ranging from website defacement to stealing or altering of sensitive information. Therefore, it is unequivocal to put in place adequate security measures to arrest if not to impede such situations which may compromise the e-manufacturing system

The first line of defence in restricting unauthorised access to the web server and therefore the manufacturing setup is the firewall. The firewall was configured to control the flow of traffic in and out of our network based on certain predetermined rules as follows. Firstly, members of public should be allowed to view general (unclassified) information about the project, so a HTTP connection from anywhere to port 80 on the server is allowed. Secondly, the web-based manufacturing system is only accessible via port 443 (HTTPS connection) thus incoming connections from only the authorised user's computer bound for port 443 is allowed. Thirdly, the web server should be allowed to make connections to anywhere, for instance to download necessary security updates. To enforce this rule, we examine 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 (Cheswick and Bellovin, 1994). Therefore, packets from outside hosts sent in response to a connection initiated by the server are allowed to pass through the filter. Finally, the default policy is to drop packets for any connection, which does not match the above rules.

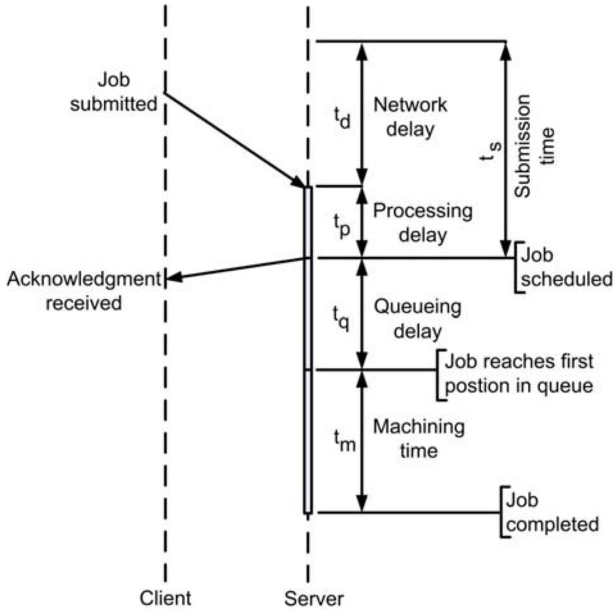
After passing the firewall, the web server authenticates users prior to granting permission to submit jobs, monitor the machining process or participate in virtual discussion. The challenge-response authentication scheme ensures that an unauthorised person using an authorised user's computer cannot use the system.

In a normal HTTP session, the data exchanged between the client and the server is transmitted in plaintext over the internet. For the e-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. Thus, all sensitive communication between the clients and the server including challenge-response authentication is carried over transport layer security (TLS). The TLS protocol (Dierks and Allen, 1999) 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.

5 System response

The various temporal delays from job submission to completion are highlighted in Figure 6. The processing delay is the time it takes the server in decoding the machining parameters in the job posted via CGI, writing these parameters in a job file and then finally transferring it to the scheduling system. The rest of the time delays have their usual meanings.

Figure 6 Components of time delay from job submission to job completion



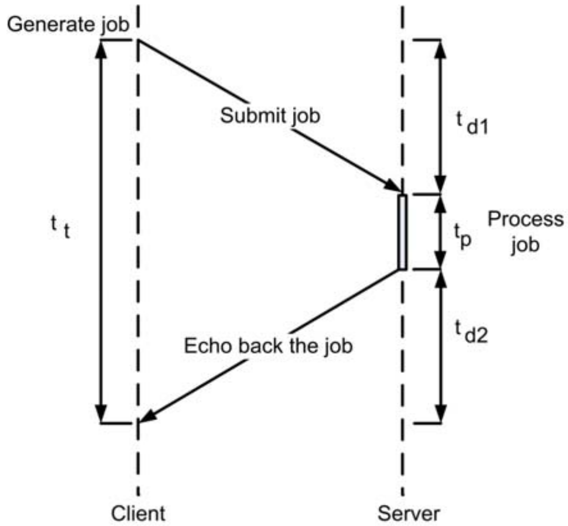
In designing the experiment to measure these delays, software timers were systematically placed in the server modules of the e-manufacturing system. To measure the network delay, the server needs to know the precise time the client submitted the job and also the time the job reached the server. By taking the difference between these times, the network delay can be established. However, the problem with this approach is that the clocks on the client and the server need to be synchronised which is difficult to achieve within the constraints of the Java sandbox security model. Thus, a common approach (Dalton, 2001) is to approximate the end-to-end delay as half the RTT delay.

$$t_d \approx RTT / 2 \tag{1}$$

5.1 Submission time and network delay

Based on the approximation in equation (1), Figure 7 shows the revised approach to measuring job submission time such that the need for clock synchronisation is eliminated.

Figure 7 Approach in approximating job submission time



The client generates and submits a job containing random coordinates within acceptable range. The server measures the processing time, t_p and then responds by bouncing the job back to the client along with the processing time. The client records the total time between sending the job and receiving response from the server, t_t. This process is repeated numerous times to get good measure of t_p and t_t which are then used in approximating submission time as follows.

From (1) it follows that

$$t_d \approx \frac{t_{d1} + t_{d2}}{2} \tag{2}$$

From Figure 7 it is clear that

$$t_t - t_p = t_{d1} + t_{d2} \tag{3}$$

Combining (2) and (3) we get

$$t_d \approx \frac{t_t - t_p}{2} \tag{4}$$

Now, the definition of submission time from Figure 6 is

$$t_s = t_d + t_p \tag{5}$$

Thus, combining (4) and (5) we get the following expression for submission time

$$\begin{gathered} t_s \approx \frac{t_t - t_p}{2} + t_p \\ t_s \approx \frac{t_t + t_p}{2} \end{gathered} \tag{6}$$

To measure job submission times, a Java test applet was designed based on the above approach. Internet users from anywhere were allowed to participate in testing. First, the test applet gathered information from the participating user about their internet connection type, transmission speed and location. The applet recorded t_t and the server recorded t_p for different jobs sizes ranging from 100 to 600 coordinates in increments of 100. For each job size, the experiment was repeated at least 100 times from the same connection. Finally, all the results accumulated by the test applet are posted on to the server for analysis.

The comparison of job submission times across various locations and internet connections is shown in Table 2. As was expected, generally job submission time is directly proportional to the job size. In addition to this, the physical distance from the machining centre and the transmission speed of remote user's internet connection greatly influences the submission times. In some of the trials, there was a lack of positive correlation between job size and submission time, cause of which can be attributed to dynamic factors such as queuing delay (§2) which predominate end-to-end delays in typically congested networks.

Table 2 Comparison of job submission times across various internet connections

Job size (coordinates)	*Mean submission time (ms)*					
	Suva	Port Vila	Melbourne	Pakistan	Bangalore	USA
	2 Mbps LAN	100 Mbps LAN	100 Mbps LAN	56 kbps dialup	128 kbps ISDN	1 Mbps DSL
100	447.3	2351.5	1123.1	3608.3	951.1	535.3
200	581.4	3030.9	1494.7	5802.0	1456.3	835.5
300	831.8	3493.4	2132.9	7199.3	2059.1	1115.3
400	906.3	4836.8	2120.5	6989.1	3037.9	1235.7
500	996.5	5106.6	2143.3	11843.1	2444.6	1499.1
600	764.4	6291.4	2517.9	14917.4	3253.7	1722.9

With the current best effort service model of the internet, it is difficult to predict the instantaneous end-to-end delays, which is a major constituent of job submission time. However, the advent of QoS mechanisms based on service models such as IntServ and DiffServ enable certain guarantees to be made on provision of network resources thus introducing element of predictability in the level of service desired by the application (El-Gendy et al., 2003).

5.2 *Processing delay*

The processing delay t_p has been noted to be solely dependent on the job size without regard to the internet connection or location from which the job originated. Processing delay (*ms*) which is related to the job file size, f (*bytes*) by the following expression is a negligible element of delay (Table 3).

$$t_p \approx 0.0009f - 0.12 \tag{7}$$

Table 3 Processing delay and machining times for sample job sizes

Job size coordinates	*Job file size, f (bytes)*	*Processing delay, t_p (ms)*	*Machining time,t_m (s)*
100	795	0.8	2,405.7
200	1,484	1.1	4,572.8
300	2,208	2	6,950.2
400	2,920	2.1	9,193.8
500	3,621	3	11,478.6
600	4,297	4	13,708.0

5.3 Queuing delay and machining times

For the job scheduling system implemented using FCFS policy, the queuing delay experienced by the i^{th} job, t_{qi} is simply expressed as:

$$t_{qi} = \sum_{j=1}^{i-1} t_{mj} \tag{8}$$

Where t_{mj} is the machining time for the j^{th} job.

The machining time for a given job primarily depends on the time to setup up work piece and the total time taken to move in x, y direction and drill individual holes. Table 3 presents machining times for sample jobs.

Putting all things together, the total process time from submission to the completion of the i^{th} job is expressed as

$$t_{total} = t_d + \sum_{j=1}^{i} t_{mj} \tag{9}$$

5.4 Fairness

The main reason for electing FCFS policy for job scheduling stems from the fact that user of the system can be situated in any time zone and as such, fairness in scheduling is of prime consideration. While FCFS algorithm is known to provide fairness in scheduling, we are interested in investigating whether the job submission and scheduling system implemented gives a fair chance to all the users in a concurrent job submission scenario. To this effect, a test Java applet was developed which first generates a job consisting of 100 randomly generated coordinate pairs and then sequentially submits the same job to the server 20 times. The test applet was simultaneously executed by five different user logins namely; user1, user2, user3, user4 and user5 on five computers on the same subnet of the USP network. Once all the applets had finished submitting their jobs, the trial was then repeated numerous times to check for consistency in results obtained.

Consider the job queue to be empty and a single user starts submitting a series of n jobs. The probability that the i^{th} position in the queue will be filled by the i^{th} job from the user is one. Now, suppose there are m users simultaneously submitting n similar jobs each. In competing for the empty slots in the queue, the probability that a job from j^{th} user occupies i^{th} position in the queue is $1/m$. Furthermore, if the job submission system treats all users fairly, then the mean positional spread, that is the distance between the i^{th} and $(i+1)^{th}$ job in the queue belonging to the j^{th} user should be m (Figure 8). We define this mean positional spread as fairness index of a given user, which can be obtained from the slope of the graph of position in queue against job number for that user.

Figure 8 Derivation of fairness index for the case of concurrent job submissions

	user:j, job: i	user:1, job: i+1	user:2, job: i+1	user:3, job: i+1	. . .	user:m, job: i+1	user:j, job: i+1	
Position in queue	p	p + 1	p + 2		. . .		p + m	

The job distribution of the users in this experiment is presented in Figure 9. By applying linear regression to the distributions, the fairness index for the different users is approximated in Table 4. The users have comparable fairness indexes, which mean that our system is not discriminant to any particular user. It is worth mentioning that the execution of the test applet by the users may not have been precisely synchronised to the exact millisecond thus the deviation of fairness index from the ideal value of five.

Figure 9 Job distribution for the different users

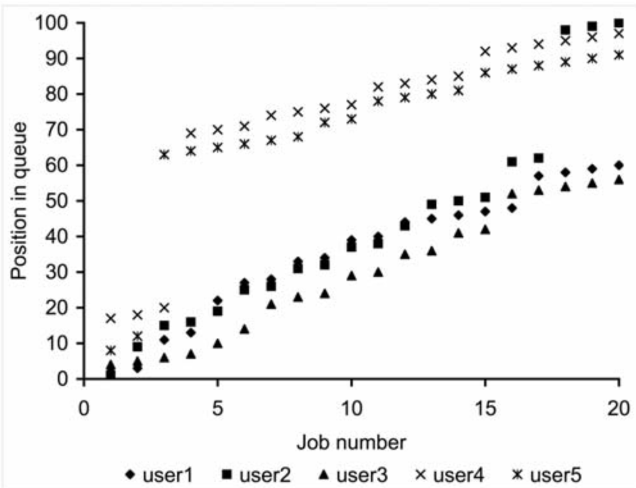


Table 4 Fairness index for the different users

User	*Fairness index*
user1	3.02
user2	4.69
user3	3.09
user4	3.67
user5	3.17

6 Conclusions

Presented with challenges in safely tele-operating a CNC drilling machine within the framework of a virtual manufacturing enterprise, a three-tiered e-manufacturing system was designed and implemented. The developed system promotes fairness in coordinating multi user access to the machining unit in accordance with the first come first serve scheduling policy. In addition to this, the integrity and safety of the machining unit is maintained by means of network security provisions, which include the TLS protocol, web server authentication mechanism and firewall.

The interaction between the remote users and the manufacturing system by means of job files is asynchronous and therefore the susceptibility of performance degradation of the web-based manufacturing system as result of uncertain temporal delays in network communication is significantly reduced. In other words, the effects of temporal delay may be felt during job submission. However, the time taken to submit a job is generally much less than the time taken to machine it and more importantly during machining all jobs already in queue are treated equal irrespective of where they originated or how long it took to arrive to the machining centre.

References

Ahn, S.H., Sundarajan, V., Smith, C., Kannan, B., D'Souza, R., Sun, G., Mohole, A., Wright, P.K., Kim, J.H., McMains, S., Smith, J. and Séquin, C.H. (2001) 'CyberCut: an internet-based CAD/CAM system', *Journal of Computing and Information Science in Engineering*, Vol. 1, pp.52–59.

Bolot, J.C. (1993) 'Characterizing end-to-end packet delay and loss in the internet', *Journal of High-Speed Networks*, Vol. 2, No. 3, pp.305–323.

Chen, L., Bender, P., Renton, P. and El-Wardany, T. (2002) 'Integrated virtual manufacturing systems for process optimisation and monitoring', *Annals of CIRP Spain*, Vol. 51, No. 1.

Cheswick, W.R. and Bellovin, S.M. (1994) *Firewalls and Internet Security: Repelling the Wily Hacker*, Addison Wesley, New York.

Clark, D.D. (1988) 'The design philosophy of the DARPA internet protocols', *Proc. SIGCOMM '88 Computer Communications Review*, Vol. 18, No. 4, pp.106–114.

Dalton, B. (2001) *Techniques for Web Telerobotics*, PhD dissertation, University of Western Australia, Perth, Australia.

Dierks, T. and Allen, C. (1999) *The TLS Protocol Version 1.0*, RFC 2246.

El-Gendy, M.A., Bose, A. and Shin, K.G. (2003) 'Evolution of the internet QoS and support for soft real-time applications', *Proceedings of the IEEE*, Vol. 91, No. 7, pp.1086–1104.

Enger, R. and Reynolds, J. (1993) FYI on a Network Management Tool Catalog: Tools for Monitoring and Debugging TCP/IP Internets and Interconnected Devices', FYI 2, RFC 1470.

Furnell, S. (2005) 'Internet threats to end-users: hunting easy prey', *Network Security*, Vol. 2005, No. 7, pp.5–9.

Jagdev, H.S. and Browne, J. (1998) 'The extended enterprise – a context for manufacturing', *Production Planning and Control*, Vol. 9, No. 3, pp.216–229.

Kong, S.H., Park, J., Han, Y., Kim, G. and Lee, K. (2002) 'The internet-based virtual machining system using CORBA', *Integrated Manufacturing Systems*, Vol. 13, No. 5, pp.340–344.

Kurose, J.F. and Ross, K.W. (2001) *Computer Networking: A Top-down Approach Featuring the Internet*, Addison Wesley Longman Inc., New York.

Lal, S.P. (2004) 'Internet-based secured remote control of computer numerically controlled drilling machine', Master's thesis, University of the South Pacific, Suva, Fiji.

Lal, S.P. and Onwubolu, G.C. (2007) 'Three tiered web-based manufacturing system – Part 1: system development', *Robotics and Computer-Integrated Manufacturing*, Vol. 23, No. 1, pp.138–151.

Lee, R.S., Tsai, J.P., Lee, J.N., Kao, Y.C., Lin, G.C.I. and Lu, T.F. (2000) 'Collaborative virtual cutting verification and remote robot machining through the internet', *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, Vol. 214, No. 7, pp.635–644.

Luo, R.C., Su, K.L., Shen, S.H. and Tsai, K.H. (2003) 'Network intelligent robots through the internet: issues and opportunities', *Proceedings of the IEEE*, Vol. 91, No. 3, pp.371–381.

Mills, D.L. (1983) *Internet Delay Experiments*, RFC 889.

Oboe, R. and Fiorini, P. (1997) 'Issues on internet-based teleoperation', *Proceeding of SYROCO*, Nantes, France, pp.611–617.

Onwubolu, G.C., Aborhey, S., Singh, R., Reddy, H., Prasad, M., Kumar, S. and Singh, S. (2002) 'Development of a PC-based computer numerical control drilling machine,' *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, Vol. 216, No. 11, pp.1509–1515.

Paxson, V. and Floyd, S. (1997) 'Why we don't know how to simulate the internet,' *Proceeding of Winter Simulation Conference*, Atlanta, GA, pp.1037–1044.

Postel, J. (ed.) (1981) *Internet Protocol*, STD 5, RFC 791.

Renton, P., Bender, P., Veldhuis, S., Renton, D., Elbestawi, M.A., Teltz, R. and Bailey, T. (2002) 'Internet-based manufacturing process optimization and monitoring system', *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA02)*, Washington DC.

Sanghi, D., Agrawala, A.K., Gudmundsson, O. and Jain, B.N. (1993) 'Experimental assessment of end-to-end behavior on internet', *Proceedings of IEEE Infocom*, San Fransisco, CA, pp.867–874.

Shi, J., Huang, G.Q. and Mak, K.L. (1998) 'Web-based design for assembly', *Proceedings of the Third Annual International Conference on Industrial Engineering Theories, Applications and Practice*, Hong Kong.

Tanenbaum, A.S. (2001) *Modern Operating Systems*, 2nd ed., Prentice Hall Inc., New Jersey.