MESSAGE QUEUING MIDDLEWARE ON VPN AND ITS TRAFFIC CONTROL SCHEME

Modeling Asynchronous Message Queuing Communication on MPLS VPN

Hiroshi Yamada
NTT Service Integration Laboratories, Communication Traffic Project,
3-9-11, Midori-Cho, Musashino-Shi, Tokyo, Japan

Akira Kawaguchi
NTT Advanced Technology Corporation, Core Networks Business Headquarters,
1-19-3, Nakacho, Musashino-Shi, Tokyo, Japan,

Keywords: Message queuing, Distributed application, SOA (Service Oriented Architecture), RFID (Radio frequency identification), Sensor network, MPLS VPN (Multiprotocol label switching virtual private network), Traffic controlInternet services, Dial-up networking.

Abstract: Message oriented middleware is being used as an advanced enterprise application integration tool. It can establish reliable asynchronous communication sessions between distributed application objects. In such enterprise communications, guaranteed message delivery is critical. In this paper, we consider message queuing communication on an MPLS VPN. In client VPN sites, a large amount of data is generated by computers and it is stored in the local message queue. Then data is processed, filtered, and sent to the remote message queue in the server VPN site. In the server VPN site, stored messages are dequeued and sent to the business application or database servers in the server VPN site. The data generating processes in the local VPN sites may be different. Data may be generated explosively during a short period in some local VPN sites. It is necessary to utilize the message queue resources in the whole VPN efficiently in order to avoid an overflow of messages and to guarantee the message delivery during such a bursty period. This paper describes a traffic control scheme that can adaptively change the rate at which data is sent from the local message queue in the VPN site to the remote message queue in the server VPN site. We created a OPNET simulation model that can precisely simulate the above control scheme on an MPLS VPN and showed its effectiveness through simulation studies.

1 INTRODUCTION

Many enterprises are introducing the service oriented architecture (SOA) concepts (Chappell, 2004) and distributed computing technologies to visualize the flow of information and money. Although enterprise application integration (EAI) tools have been introduced, the hub and spoke architecture that old EAI tools need is not scalable, so several expansions are needed to enable us to efficiently communicate among several distributed application objects. As an advanced infrastructure that provides such distributed application integration, message oriented middleware (MOM) is being introduced. It can establish reliable asynchronous communication sessions between distributed application objects.

Radio frequency identification (RFID) technologies enable direct machine-to-machine or device-to-device communications within distributed computer networks. RFID is being applied to enterprise applications (Clauberg, 2004), (Sikander); for example, supply chain management (SCM), baggage handling in airports, sensor networks for environmental monitoring, and condition-based maintenance systems. Compared with the data flow of Web access, where a large amount of data flows from a central web server to clients, such machine-to-machine or device-to-device communication traffic flows from devices to a few central servers. In addition, this traffic needs high transactional guarantees. If data transportation is disrupted, the system is required to perform rollback recovery to

Yamada H. and Kawaguchi A. (2006).
MESSAGE QUEUING MIDDLEWARE ON VPN AND ITS TRAFFIC CONTROL SCHEME - Modeling Asynchronous Message Queuing Communication on MPLS VPN.
In *Proceedings of the International Conference on e-Business*, pages 46-54
DOI: 10.5220/0001424700460054

maintain consistent integrity of the data. Guaranteed message delivery is critical in many systems like SCM. This is the reason that message queuing is adopted as the communication scheme. Message queuing communication can be divided into communication between the device and the server using the message queue. For example, let us consider that the data that is obtained by the RFID reader is processed in the remote application server. If the synchronous communication scheme is adopted, then when the remote application is busy, the application will be locked up and a lot of data will be lost. RFID cannot process other jobs while it is waiting for the acknowledgement from the remote application server. If asynchronous communication like message queuing communication is adopted, then even if the remote application is busy, the generated message is stored in the local message queue, sent to the remote message queue, and then sent to the remote application server. The message queue communication can achieve the guaranteed message delivery.

For enterprises that want to create an SCM system with other partners like suppliers, deliverers, and retailers, the virtual private network (VPN) service is a cost-effective solution. The MPLS VPN (multiprotocol label switching virtual private network) service can provide secure and any-to-any connectivity to subscribers (Pepelnjak et al., 2003). Therefore, most enterprises use a VPN service to connect to remote applications. The MPLS VPN is provided on a wide-area public network by the network provider. In the MPLS VPN, any node can belong to several VPNs if the policy of the import and export of the routing distinguisher is suitably configured. So, if the enterprises want to gather and process the data obtained from several enterprise VPNs, the network provider can provide the node as the root message queue, which can communicate with the enterprise client message queue nodes. Using the MPLS fast rerouting technology, when a failure occurs in nodes or links in the middle of the LSP (label switched path) in the MPLS network, the local protection mechanism for the LSP starts immediately and becomes effective in a short time. Compared with the enterprise network built by the enterprise itself, it is generally more cost-effective for users to create a message queue application overlay network on the MPLS VPN that is provided by the network provider.

In this paper, we consider message queuing communication on an MPLS VPN from the viewpoint of a network service provider. In some VPN sites, a large amount of data is generated and stored in the local message queue at once. The remote message queue is in the server VPN site. The database or application servers are in this server VPN site. The data stored in the local message queue in local VPN sites is sent to the remote message queue in the server VPN site according to the message queuing transport protocol. To keep the integrity of the data, the local message queue makes a copy of the original message and stores it when the local message queue sends the original message. The local message queue can discard the stored copy message only when an acknowledgement from the remote message queue has been received. The data generating processes in the local VPN sites may be different. Messages may be generated explosively during a short period in some local VPN sites. Because there is a limit on the length of the local message queue, an overflow can occur. In this paper, we propose a traffic control scheme. To avoid an overflow in such a case, it is necessary to utilize the message queue resources in the whole VPN efficiently. In this control scheme, the rate at which data from the local message queue in local VPN sites is sent to the remote server message queue is adaptively controlled according to a control rule and the volume of messages in all local message queues in the VPN. To confirm the effectiveness of such a traffic control scheme in message queuing communication, we created precise simulation models by OPNET (OPNET) and analyzed its effectiveness for these simulation models.

This paper is organized as follows. The architecture of an RFID application using the message queuing scheme is explained as an example in section 2 and an overview of the MPLS VPN is shown. The traffic control scheme and OPNET modeling are explained in section 3. In section 4, the effectiveness of the proposed traffic control scheme is examined through simulation studies. Section 5 summarizes the control scheme and mentions further studies.

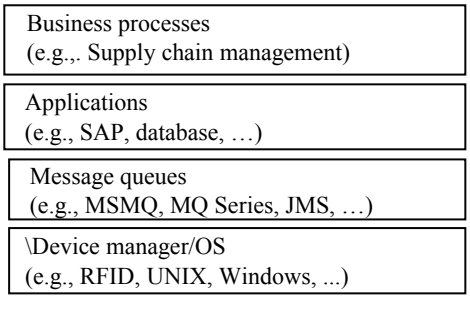


Figure 1: Software architecture.

2 REFERENCE ARCHITECTURE

2.1 Software Architecture

(Chappell, 2004), (Sikander), (Strauss), and (BEA) showed software architecture models for MOM systems or RFID application. They all considered the following four layers, as shown in Figure 1: device manager layer, message queuing communication layer, application layer, and business process layer. In the message queuing layer, the commercial message queuing products are used, for example, MSMQ (Microsoft message queue) and JMS (Java messaging service). The message queuing communication scheme is explained in detail in Section 3.

2.2 MPLS VPN

VPN is a network over the public service provider network. MPLS VPN is a networked-VPN that can provide any-to-any connectivity among several VPN sites. An MPLS LSP is established between the PE (provider edge) routers. The PE routers have the routing instance, VRF (VPN routing and forwarding) table. For each VPN, VRF is configured. Information about routes in the VPN sites is exchanged using the MP-iBGP (multiprotocol interior border gateway protocol) among all PE routers. Two labels, VPN label and LSP label, are attached to the arriving packet at the ingress PE router. In the provider network, a packet is forwarded according to the LSP labels instead of the destination address of the packet. At the egress PE routers, the VPN label is popped and forwarding to the appropriate VPN sites according to the VRF. To reduce the operation expenditure and capital expenditure, most enterprises use the provider VPN service instead of building their own network. Therefore, the message queuing communication traffic among the distributed message queues in several VPN sites may flow in the VPN. Figure 2 shows the MQ service on the MPLS VPN. In this figure, the MQ node works as the CE router and the message queuing (MQ) server.

3 MESSAGE QUEUING COMMUNICATION ON MPLS VPN VARIABLES

3.1 Overview

We explain how message queuing middleware can certainly deliver the messages using the example in

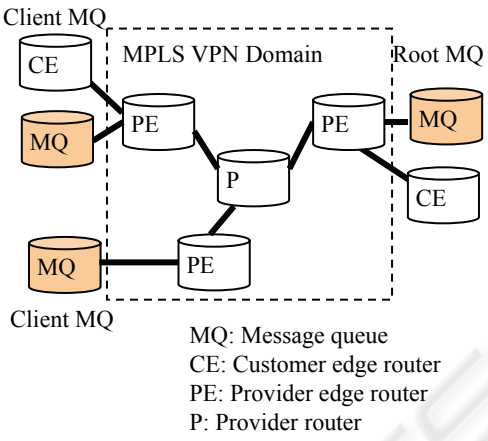


Figure 2: MQ service on VPN.

Figure 3. In this figure, the sender application tries to send a message to the receiver application. The message queuing communication is asynchronous. Therefore, this communication is treated as three transactions: (1) the sender application puts the message in the local message queue, (2) the local message queue moves the message to the remote message queue, (3) the receiver application gets message from the remote message queue. Because the above transactions are completely separate, recovery is possible in the event of a failure. In transaction (1), the first transaction is completed when the local message queue stores a message. In transaction (2), the second transaction is completed when the message is moved and surely stored in the remote message queue. Finally, in transaction (3), the third transaction is completed when the application gets the message. The remote application is ignorant of the states of the local application and message queues. Message queuing communications mainly corresponds to transaction (2) in the above example. The message stored in the sender message queue is not removed until an acknowledgement of its reception by the receiver message queue is received in the sender message queue.

Here, we remark on the features of traffic in RFID or sensor networks. One is the direction of the

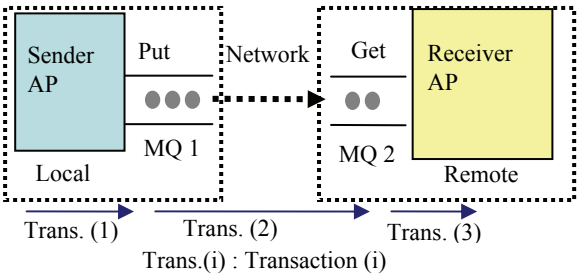
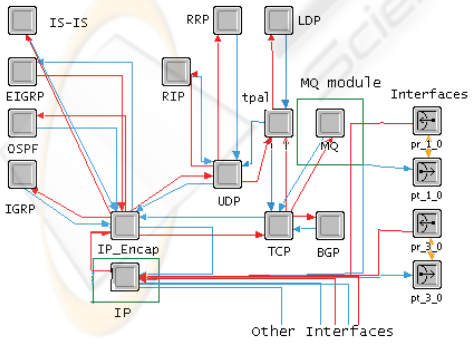


Figure 3: MQ communication.

traffic flow. In such networks, the direction of traffic flow is from multiple local sites to a few central server sites. Second, the message delivery is strongly required to be guaranteed. In the case of failures in a node or a link in the network, a rollback scheme is required to maintain the consistency of the data. Even if the messages are explosively generated in the local VPN sites, their delivery should be guaranteed. If synchronous communication like ftp or Web access is adopted, the local server must wait for the acknowledgement from the remote server. When the network is congested, the acknowledgement is delayed and the local server may lose some messages generated during this delay. Because the transactions are independently separate in the message queuing middleware in figure 3, these messages are stored in the message queue and never lost if we configure the queue size well.

3.2 Modeling of Message Queuing Communication

In order to simulate the message queuing communication on MPLS VPN that explained in the previous subsection, we developed a simulation model using OPNET. Here, we briefly explain the modeling. The OPNET node model of the message queuing node is shown in figure 4. This message queuing (MQ) node model consists of several protocol modules (for example, IP, TCP, UDP, and several routing protocols) and receiver and transmitter interfaces. Because the developed MQ node establishes a TCP connection between the other MQ nodes, the MQ module is connected to the TCP protocol module. The MQ module deals with the message queue communication protocol. Other routing and IP modules deal with the routing process



This is a snapshot of the OPNET node model in the root MQ node. In the client MQ node model, the traffic generator module is added and this module is connected to the MQ module.

Figure 4: Node model.

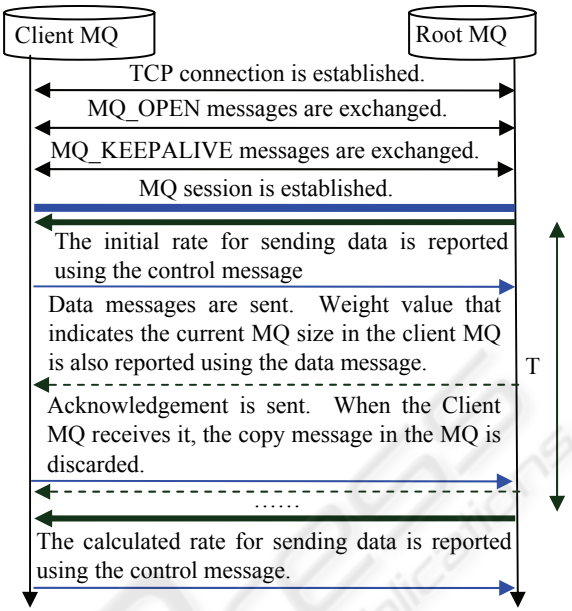


Figure 5: Procedure for establishing MQ connection.

as the CE router. Here we consider two types of MQ node, the client MQ node and root MQ node.

In the client MQ node, the message generating module is connected to the MQ module. In this simulation model, the messages are generated in the client MQ node. Generally, the client MQ node receives messages from many RFID devices or PCs by wireless or wired communication and stores them. Here, for the sake of simplicity, we do not model the above communications between client MQ nodes and other devices including the application server. Instead of modeling the above wireless or wired communication, we add a traffic generator module in the client MQ node model to generate messages, which are sent to the MQ module, which stores them in the message queue. In the root MQ node, the messages stored in the message queue are removed at the rate they are transferred to the application server in the server sites.

It is necessary to establish a secure connection between client MQ and root MQ nodes. The procedure for establishing the connection is shown in figure 5. It is similar to that of BGP (border gateway protocol). In our model, neighbour information is configured in each message queuing node. This information provides the IP address and network mask of the remote MQ node and the update source that is the source interface of the MQ connection. First, each MQ node establishes a TCP connection. After that, they exchange MQ_OPEN and MQ_KEEPALIVE messages. When they both receive a KEEPALIVE message, the MQ connection is established. The messages in the client MQ node are dequeued and sent to the remote root MQ node

through the established MQ connection. In order to guarantee secure delivery, a copy message is created when the original message is dequeued. If the client MQ node receives the acknowledgement of receiving the message from the root MQ node, then the copy is removed and the MQ transaction is completed. If the acknowledgement does not arrive at the client MQ node, the copy is not be removed. The control information is sent from the root MQ node to the client MQ node through the established MQ session. The management scheme of the MQ session is the same as that of the BGP session.

3.3 Control Scheme to Deal with Messages Generated Explosively During a Short Period

Congestion control has been an active research area for Internet protocols. These schemes are for best-effort delivery and assume that all receivers get the same set of messages. In contrast, as mentioned in (Pietzuch et al., 2003), compared with an Internet router, in a message queue, the ratio of maximum queue size to the message processing throughput is larger. The large queue can handle the burstiness due to the application-level scheduling. In message queuing communication, the messages may be filtered or processed in the intermediate message queue server. Because messaging communication is sensitive to loss rather than to delay, the rate at which messages are sent from the client MQ node to the root MQ node is determined not by the end-to-end delay but by the queue sizes of all client MQ nodes.

(Pietzuch et al., 2003) proposed the congestion control scheme in the context of a publish/subscribe messaging model. In their model, a feedback loop control scheme is implemented between a publisher-hosting broker and all subscriber-hosting brokers, which is used to adjust the rate of publishing new messages. Messaging middleware is deployed as an overlay network of application-level routers.

The basic idea of our congestion control scheme is similar to that of (Pietzuch et al., 2003). In our model, the feedback protocol explained in the previous section is implemented between the client MQ and root MQ nodes. The rate at which data is sent from the local message queue in the VPN site to the remote message queue in the server VPN site is determined by the message size in each client MQ in order to utilize the message queue resources in the whole VPN efficiently. In this paper, we consider that the MPLS VPN network is used to make the application overlay network. We consider that the network service provider provides a customer edge node that has a large message queue and the function

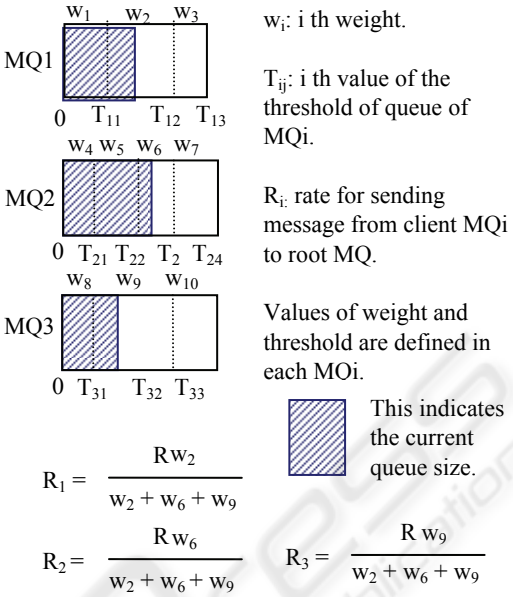


Figure 6: Calculation of the transmitting rate.

of the message queuing communication.

The processes of generating messages in the client MQ node are different and messages may be generated explosively during a short period in some client MQ nodes. Because there is a limit on the size of the message queue, an overflow will occur in this situation. In order to avoid overflows, it is necessary to utilize the MQ resources in the whole VPN efficiently.

In this paper, the following traffic control scheme is considered. In this control scheme, the rate at which data is sent from the client MQ node in VPN sites to the root MQ node in the VPN server sites is adaptively controlled according to the control rule and the volume of messages in all client MQ nodes. For simplicity, we explain it using the following example. We considered three client MQ nodes and one root MQ node in this simulation experiment. The stored messages in the root MQ node are sent to the server applications in the server site at the maximum R bps. An index like MQi (i = 1, 2, 3) is assigned. In the initialization, the root MQ sends a control packet to the i_th client MQi node to report the initial rate R_i (i = 1, 2, 3). The initial value, R_i, is equal to R/3. The client MQi node always measures the total volume of stored messages in the MQ node. Multiple threshold values and weights for the stored message size are configured, as shown in figure 6. When the client MQ node sends the message to the root MQ node, the current weight value is measured and this value is set in the field of the MQ packet. For example, the current message queue size in MQ1 in figure 6 is larger than T_{11} but less than T_{12},

so, the current weight is w_2. In the same way, the current weights of MQ2 and MQ3 are w_6 and w_9, respectively. The MQ packet has some fields. The original message is stored in the message field. The current weight value of the client MQ node is set in the weight field. In figure 6, MQ1 reports the current weight, w_2, to the root MQ. MQ2 and MQ3 report the current weights, w_6 and w_9, respectively.

The root MQ node maintains the table of the current weights of all client MQ nodes and the elapsed time after the last expired time. The timeout value is determined in the root MQ node. When the root MQ node receives the MQ packet, it updates this table. When the elapsed time exceeds the timeout value, the root MQ node calculates the new rate R_i according to the equations shown in figure 6. The new rate R_i is calculated as R multiplied by the ratio of the latest MQi node's weight to the summation of the latest total weights of all client MQ nodes. The calculated new rate value, R_i, is reported by sending a control packet to each client MQ node. The timer in the root MQ node is reset and restarted again.

When the client MQ node receives this control packet, it changes the rate at which message data is sent to the root MQ node. As a result, when the number of messages in the client message queue increases because of an explosive generation of messages during a short period, more messages in the client MQ node are sent to the root MQ node than before. In this way, we can avoid an overflow in this client MQ node.

4 CASE STUDIES

4.1 MPLS VPN Configuration

In the case study, we considered the MPLS VPN shown in figure 7. The network model consisted of five P routers, four PE routers, four CE routers, six client MQ nodes, and 2 root MQ nodes. The OSPF (open shortest path first) protocol was running as the baseline IP routing protocol. MPLS (multiprotocol label switching) and LDP (label distribution protocol) were configured in the provider network, which consisted of P and PE routers. The LSPs were fully meshed among all PE routers. The VRF, router target, and route distinguisher were configured in the PE routers for each VPN. The static routing was configured between the PE and CE routers. In order to advertise the prefixes belonging to VPN sites, MP-iBGP was configured. MP-iBGP sessions were established among all PE routers in full mesh fashion. The static routing from PE router to CE router was redistributed in the MP-iBGP. In the VPN sites, RIP (routing information protocol) was configured. In our simulation model, when a packet from VPN sites arrived at the ingress PE router, two labels were assigned: a VPN label and an LSP label. In the provider network, label packets were forwarded according to the label forwarding table. When the packet arrived at the egress PE routers, the VPN label was popped and the packet was forwarded to the appropriate CE routers or MQ nodes according to the VRF table.

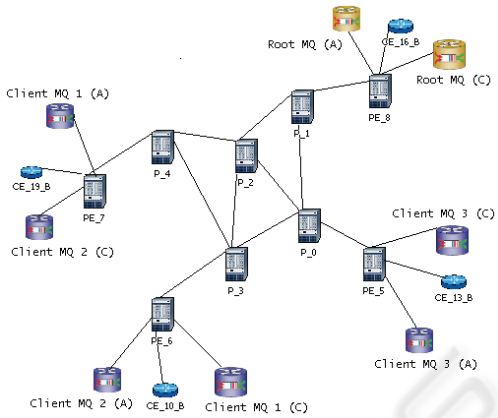


Figure 7: Network model (snapshot of the OPNET Network Editor).

4.2 MQ Node Configuration

The message queue sizes were set to 4 Mbytes for the root MQ node and 1 Mbytes for the client MQ node. The stored messages in the root MQ node were transmitted to the application server at 1 Mbps (R). In this simulation model, a block consisting of 25 messages was removed from the root MQ node every 1 second. The message size was fixed at 5k bytes. Therefore, the client MQ node could store a maximum of 200 messages. The interval of calculating the rate (T in figure 5) was set to 2 s in the root MQ node. In the client MQ node, the weight values and threshold values in the client MQ node were configured. When the current utilization of the client MQ node was equal to or greater than X * 10% and less than 10 * (X + 1)%, the weight value (w_i) was set to X (X = 0, 1, …, 9). The neighbour information about the client MQ node was similar to that of the BGP neighbour configuration. The IP address of the remote MQ node was set to that of the root MQ node in the client MQ node. The update source was set to the loopback interface. In the root MQ node, the number of neighbour client MQ nodes was three. The IP address of the remote MQ was set to that of the client MQ node. The update source was set to the loopback interface.

4.3 Processes of Generating Messages

We considered the following three patterns for the process of generating messages in the VPN site. As mention before, in the simulation model, the generator module in the client MQ node generated messages. The average message generation rate for each pattern during the simulation is shown in figure 8. The average rates for patterns 1 and 3 were fixed during the simulation. The intervals of consecutively generated messages followed the exponential distribution with means of 0.24 s and 0.2 s for the patterns, 1 and 3, respectively. The message size was fixed at 5k bytes. Therefore, the average rate for pattern 1 was 167 kbps and that for pattern 3 is 200 kbps. In pattern 2, the rate was explosively increased between 2400 and 4200 during the simulation (we called this the bursty period). Outside such bursty periods, the intervals of consecutively generated messages followed the exponential distribution with mean of 0.24 s. During the bursty period, the intervals of consecutively generated messages followed the exponential distribution with mean 0.075 of s. The average message generation rate during the bursty period was about 534 kbps, which was about three times that during the non-bursty period. In this simulation, the message generation pattern in the client MQi node was pattern i (i = 1, 2, 3).

4.4 Effectiveness of the Traffic Control Scheme

In order to verify the effectiveness, we considered the number of overflow messages and message queue size as performance measures. In the scenario in which the control scheme was not implemented, the rate at which data was sent from the client MQ node to the root MQ node was fixed at the initial rate (R/3). In this simulation, R was set to 1 Mbps. When the control scheme was not implemented, the load in the MQ2 node during the non-bursty period was 0.5 (((5 kbytes *8)/0.24)/(R/3)) and that during the bursty period was 1.6 (((5 kbytes *8)/0.075)/(R/3)). Thus, there was an overload during the bursty period and we think that overflow occurred when the traffic control scheme was not implemented.

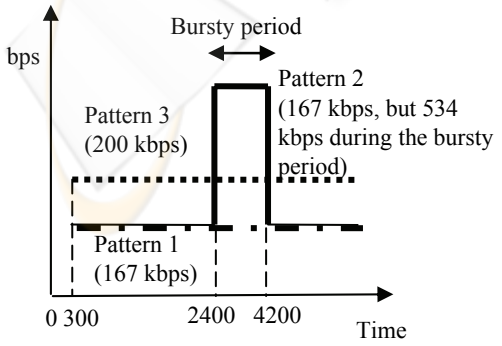


Figure 8: Traffic generating processes.

When the control scheme was not implemented, we could see that an overflow occurred in the MQ 2 node, as shown in figure 9. The size of the message queue in the client MQ2 node is shown in figure 10. When the control scheme was implemented, no overflow occurred during the bursty period. When we examined the size of the message queue in the client MQ 2 node, we found that the queue was smaller with the traffic control scheme than that without it. The size of the message queue in the client MQ 2 node without the traffic control scheme was equal to the total message queue size and some messages overflowed during the bursty period. When the adaptive traffic control scheme was implemented, the rate at which messages were sent from the client MQ 2 node to the root MQ node was changed as shown in figure 11. More messages in MQ 2 node sent to the root MQ node during the bursty period than during the non-bursty period. The moving average size of the message queue in the root MQ node is shown in figure 12. The moving average means the continuous average of the ordinate value over intervals of a specified width (60). The size of the message queue in the root MQ node increased when the adaptive traffic control was implemented. These results show that the messages were efficiently stored in all message queues in the whole network and this control scheme could avoid overflows and guarantee message delivery when the adaptive traffic control was implemented.

5 CONCLUSION

In this paper, we considered the message queuing communication on an MPLS VPN. The application overlay network is made on the MPLS VPN network. It is possible to configure the message queue node that belongs to several enterprise MPLS VPNs. The use of fast rerouting mechanisms of the MPLS network is expected to make the application overlay network more reliable. In order to efficiently utilize the message queue resources in the whole network and to avoid an overflow in the local message queue when a lot of messages are suddenly and explosively generated during a short period, we proposed the adaptive traffic control scheme. In this scheme, the client MQ nodes and the root MQ node can communicate with each other. The client MQ node notifies the root MQ node about the current

weight, which indicates the current size of the message queue in the client MQ node. The root MQ node calculates the rate for sending messages from the client MQ node to the root MQ node and notifies the client MQ node about the calculated value for each client MQ node. This control scheme can adjust the rate for sending messages from client MQ node to root MQ node. We showed its effectiveness through simulation studies. The described model is a centralized solution. The backup solution of the root MQ node should be used at the same time.

The method of adjusting the rate of sending messages from the client MQ node to the root MQ node in this paper was simple. Developing more sophisticated methods for adjusting the rate is further study. We considered that the occurrence probability of the case in which messages may be suddenly and explosively generated at the same time in all the client MQ nodes, to be very small. In such a case, our current control scheme cannot avoid overflows, so we need to add another control scheme to deal with this situation. This is also for further study.

REFERENCES

David A. Chappell, Enterprise Service Bus, O'Reilly, 2004.

Rolf Clauberg, RFID and Sensor Networks – From Sensor/Actuator to Business Application, RFID workshop, University of St. Gallen, Switzerland, Sept. 27, 2004, http://www.zurich.ibm.com/pdf/sys/RFID_and_Sensor_Networks.pdf.

Peter R. Pietzuch and Sumeer Bhola, Congestion Control in a Reliable Scalable Message-Oriented Middleware, MIDDLEWARE2003, pp.202-221, 2003.

Javed Sikander, RFID Enabled Retail Supply Chain, http://msdn.microsoft.com/library/default/asp?url=/library/en-us/dnbda/html/RFIDRetSupChn.asp

Dipl.-Ing. Heinz Strauss, Towards A Ubiquitous Future Limited Only By Our Imagination…, http://www.itu.int/osg/spu/ni/ubiquitous/Presentations/13_strauss_future.pdf

Chris Britton, IT Architectures and Middleware: Strategies for Building Large, Integrated Systems, Pearson Education, 2001.

Ivan PepeInjak, Jim Guichard, and Jeff Apcar, MPLS and VPN Architectures I, II, Cisco Press, 2003.

Chris Loosely and Frank Douglas, High-Performance Client/Server – A Guide to building and managing robust distributed systems, John Wiley & Sins, Inc., 1988.

BEA, BEA RFID Technical Challenges and Reference Architecture, http://www.bea.com/content/news_evenets/white_papers/BEA_RFID_ref_arch_wp.pdf

OPNET Technologies, http://www.opnet.com.

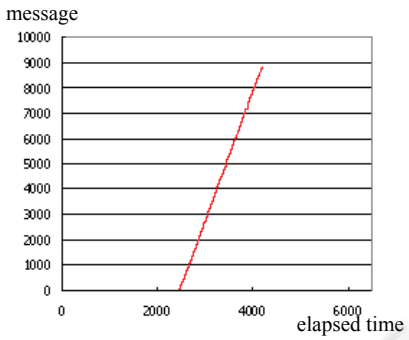


Figure 9: Cumulative overflowed messages.

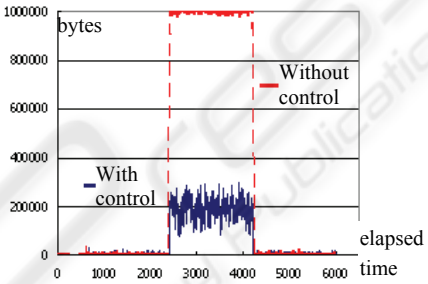


Figure 10: Size (bytes) of the message queue in the client MQ 2.

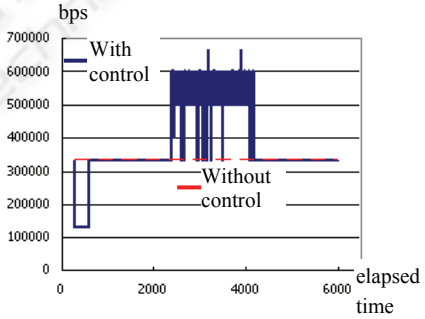


Figure 11: Rate (bps) for sending messages in the client MQ 2, when the adaptive traffic control was implemented.

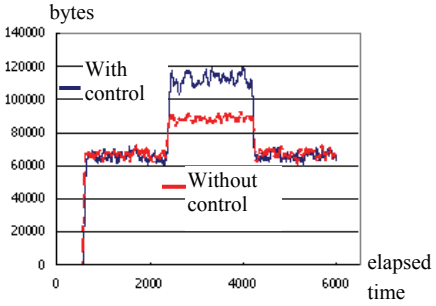


Figure 12: Moving average size (bytes) of the message queue in the root MQ.