QoS Design and Its Implementation for Intelligent Industrial Ethernet

Lu Sheng

***Abstract*—This paper has an introduction on switch management system which can configure the switch into the desired operation state based on both the user input and the default setting (hardcoded). The configuration should include the ingress policy, egress policy, QoS (Quality of Server), IGMP snooping, rate limiting, address database setting, port state setting, aging time, and some other parameters to be determined at design time. The switch should be always turned on its QoS feature, IGMP Snooping feature, and Rate Limiting feature and should allow the users to specify which priority traffic or which traffic type to be rate limited. It also has a further analysis on the QoS design of the ICIE (Intelligent Controller for Industrial Ethernet)module architecture which adopts the standard IEEE 802.1D/Q tag and the Differentiated Services (DiffServ) QoS mechanisms to mark different application message packets with different relative priorities.**

***Index Terms*—QoS, industrial Ethernet, switch management.**

I. Introduction

For the future merge of networks, the traditional industrial message Broadcast business must supported by the computer network multicast. So how to support the multicast communication in the network is the network researcher's important direction [1]. We have known the IP Multicast has been implemented and used for a long time, But multicast in the Intranet has not got the same rapid development [2].

Previous Ethernet can't support the group communication. So Multicast is treated just as Broadcast. Few years ago the Switch Ethernet with industrial Ethernet capability can support the TRUE multicast [3]. By using industrial Ethernet, the Switch Ethernet can separate the network into several broadcast domains. In case of a multicast traffic, only those hosts in this broadcast domain can send and receive the multicast data. Compared with the IP Multicast, the Multicast over Switch Ethernet does not need to support the Multicast Route function [4]. It only needs a Dynamic Group Management Protocol to manage the relations between hosts and multicast groups.

II. Switch Management

A. Switch Configuration

At the ICIE (Intelligent Controller for Industrial Ethernet) system start up time, the Switch Management subsystem configures the switch into the desired operation state based on both the user input and the default setting (hardcoded). The configuration should include the ingress policy, egress policy, QoS, IGMP snooping, rate limiting, address database setting, port state setting, aging time, and some other parameters to be determined at design time [5]. The switch should be always turned on its QoS feature, IGMP Snooping feature, and Rate Limiting feature and should allow the users to specify which priority traffic or which traffic type to be rate limited.

B. Architecture and Functionalities

Fig. 1 shows the overall architecture of the Switch Management subsystem in the ICIE module system. It contains a layer 2 RSTP component which is performing calculating and maintaining the spanning tree network topology and avoiding an active loop and a separation of an active network segment. The Switch Management subsystem interacts with the Device Management Subsystem for the configuration and diagnosing of the switch and RSTP component and the event handling. It also interacts with the NetMux component for receiving and sending the RSTP management messages (BPDU). It directly performs the access operations on the switch through the switch driver. The RTOS subsystem provides the services of the task management and synchronization to the Switch Management subsystem [6].

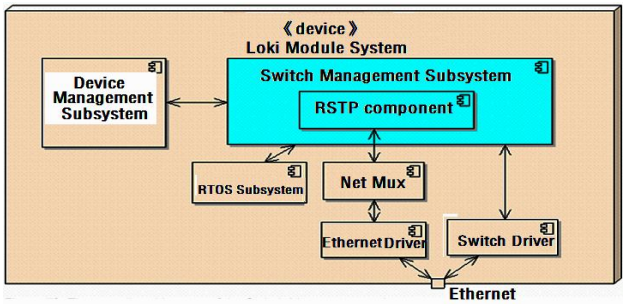

Fig. 1. The overall architecture of switch management.

The main functionalities of the Switch Management subsystem are shown in Fig. 2.

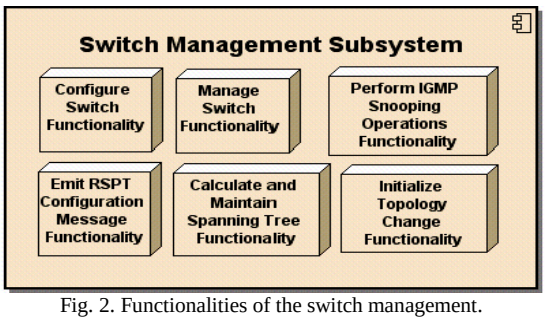

Fig. 2. Functionalities of the switch management.

Manuscript received January 1, 2015; revised June 13, 2015.

Lu Sheng is with the Chongqing Engineering Laboratory for Detection, Control and Integrated System, Chongqing Technology and Business University, Chongqing, China (e-mail: lusheng8815@126.com).

DOI: 10.7763/IJMMM.2016.V4.222

III. Performance And Scalability

Next we discuss and document the ICIE system architecture from the performance and scalability perspectives at the system level, mainly covering the IO transaction throughput and its scalability the ICIE system is intended to cope with, the QoS architecture, the overload behavior, and the system schedulability of the ICIE module.

A. Overview of System Performance Model

From the architecture perspective, the ICIE module resident control network is constituted by the end devices and network node devices with various network topologies as shown in Fig. 3. The end devices originate and/or consume the network messages to perform the desired application functionalities, which include the PLC, the ICIE module, various IO devices, PCs, HMI, and SCADA. The network node devices are hubs, switches, and/or routers whose functions are to transport various message traffics between the ICIE module (PLC) and the other end devices. Each of these network node devices has its own ingress policy and egress policy to classify, schedule, and condition the message traffic [7].

Each device (including the end devices) on the network imposes different restrictions over the message traffics flowing through it. The notable restrictions that impact the ICIE module's system architecture and the whole system performance are throughput, latency, jitter, and packet loss. The throughput is the rate at which the message traffic can flow through the device. The system throughput is determined by the slowest device (bottleneck) [8]. The latency is the time of the message traffic going through the device. The accumulation of all the latencies in the whole message traffic flow path is called the response time. The jitter is the variation of the latency (or response time). The packet loss can occur due to many reasons, for example, hardware errors, congestions, rate limiting, protocol errors, and so on [9].

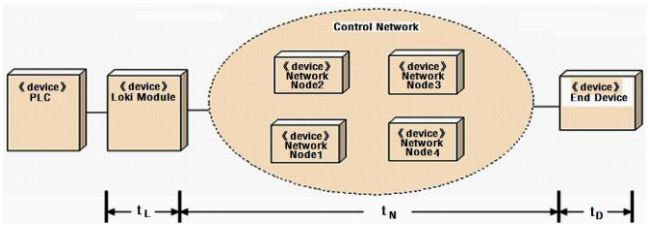

Fig. 3. The ICIE module resident control network model.

The response time t_R for one message traffic is determined by the formula

$$t_R=t_L+t_N+t_D \quad (1)$$

For the publisher-consumer model data transactions, for example, EIP IO data,

$$t_R=2\times(t_L+t_N+t_D) \quad (2)$$

For the client-server model data transactions, for example, MB/TCP IO data and EM data, where t_L is the time the ICIE module spends from preparing to sending out the message or from receiving to finishing processing the message, it includes the latency of the ICIE's Marvell switch; N is the time the data spend on the network; t_D is the time the end device spends from the receiving the data to finishing processing them or from preparing to sending out the data [10].

Different application messages can tolerate these restrictions to a different extent. Therefore, in the whole system architecture, the QoS, IGMP Snooping, and Rate limiting techniques and devices with these techniques are adopted to guarantee different throughputs, response times, jitters, and packet losses for different types of message traffics.

Another two important performance concepts are the IO scanner cycle time, t_{cycle}, and cycle number per MAST, Ncycle. The IO scanner cycle time is the time needed for IO scanner to scan each IO line exactly one times. It is the throughput dependent and is determined by

$$t_{cycle}=1000\times(\text{total IO lines})/(\text{IO throughput})/2 \quad (3)$$

where the total IO lines are the concurrent IO lines the IO scanner scans; the IO throughput is counted as the number of both the received and sent IO packets per second, the number 2 is used in the formula because each IO scan includes a request (sending) and a response (receiving) for MB/TCP IO scanning or an input (receiving) and an output (sending) for EIP IO scanning; the constant 1000 is used to make the t_{cycle} in the unit of ms [11].

The IO scanning cycle number per MAST, N_{cycle}, is the cycle number the Io scanner can scan all the IO devices during one PLC MAST cycle time. It is determined by

$$N_{cycle}=t_{MAST}/t_{cycle} \quad (4)$$

where t_{MAST} is the PLC MAST task cycle; t_{cycle} is the IO scanner cycle time. Therefore, the IO scanner cycle number per MAST is the throughput, concurrent IO line numbers, and the PLC MAST task cycle dependent.

B. IO Transaction Performance and Scalability

This section mainly addresses the IO transaction throughput goal of the ICIE module, the techniques taken in this architecture design to achieve the IO throughput goal, the scalability of the IO transaction architecture, and the relationships among IO throughput, the number of concurrent IO lines, the IO scanner cycle time, t_{cycle}, the RPI, the PLC MAST task cycle time, the fault tolerance, and the response time.

1) IO Throughput goal and scalability

The ICIE module system of this version needs to be capable of processing at least 12,000 IO PPS under the condition of 100% IO transaction traffic and also needs to be capable of processing the explicit messages at a rate of at least 1000 PPS while processing 9,600 IO PPS [12], [13]. This implies:

- Every one ms, the ICIE module needs to process 6 IO input packets and 6 IO output packets under the condition of 100% IO packet traffic environment.
- Every one ms, the ICIE module needs to process 1 EM message and 4.8 IO input packets and 4.8 IO output packets under the condition of 80% IO packet traffic environment.
- The ICIE CPU (not the PLC) needs to have enough

bandwidth and memories for achieving the required IO throughput goal.

The ICIE module can support up to 384 IO lines (256 EIP IO lines + 128 MB/TCP IO lines). As the number of the concurrent IO lines increases, the throughput goal will be met at the expense of the increase in the IO scanner cycle time as shown in Table I and the decrease of the IO transaction fault tolerance as shown in Table II.

With decreasing the PLC MAST task cycle, the number of the IO scanning cycle per MAST cycle and the fault tolerance are reduced. If the PLC MAST cycle is smaller than the t_{cycle}, the PLC will read some old input data from certain number of IO lines.

TABLE I: THE IO SCANNER CYCLE TIME UNDER DIFFERENT CONDITIONS (IO THROUGHPUTS)

Traffic Condition	IO throughput (PPS)	IO Scanning Cycle Time, t_{cycle} (ms)[1)]									
		32	64	96	128	160	192	224	256	320	384
100% IO	12,000	5.4	10.7	16	21.4	26.7	32	37.4	42.7	53.4	64
80% IO	9,600	6.7	13.3	20	26.7	33.3	40	46.7	53.3	66.7	80

2) IO throughput architecture

In order to achieve (and improve) the IO throughput goal, we take the following techniques in this version of the ICIE module architecture:

- Using pre-defined IO packets

This will reduce the processing time (the time of constructing the IO packets). For EIP IO packets, pre-define UDP + IP + Ethernet headers + some fixed fields in the EIP Common Packet Format header. For MB/TCP IO packets, pre-define TCP + IP + Ethernet headers [14].

TABLE II: THE CHANGES OF THE IO DATA TRANSACTION PROPERTY PARAMETERS WITH THE MAST TASK CYCLE

PLC MAST cycle (ms)	Number of IO liners	Number of IO scanning cycle per MAST		IO Falut Tolerance		CPU time per ms for IO deta transfer boteween DPRAM and IO buffer(μs)
		12.000 PPS	9.600 PPS	12.000 PPS	9.600 PPS	
250	32	46	37	45	36	8
	64	23	18	22	17	
	96	15	12	14	11	
	128	11	9	10	8	
200	32	37	29	36	28	10
	64	18	15	17	14	
	96	12	10	11	9	
	128	9	7	8	6	
150	32	27	22	26	21	13.3
	64	14	11	13	10	
	96	9	7	8	6	
	128	7	5	6	4	
100	32	18	14	17	13	20
	64	9	7	8	6	
	96	6	5	6	4	
	128	4	3	3	2	
50	32	9	7	8	6	40
	64	4	3	3	2	
	96	3	2	2	1	
	128	2	1	1	0	
20	32	3	3	2	2	100
	64	1	1	0	0	
	96	1	1	0	0	
	128	1	0	0	0	

- Using short stacks

This will reduce the processing time and avoid the wait time involved in the normal TCP/IP stack from the packet into the socket receive buffer to the application's beginning to read it. For EIP IO transactions using the Woodhead optimization stack. For MB/TCP IO transactions using the in-house implemented reduced stack [15].

- Using separate receiving queues

This will avoid the delay time by processing the explicit messages coming earlier than the IO packets. Separate the receiving queues for the EIP IO packets and MB/TCP IO packets from the queues for the explicit messages.

- Using a 1ms timer to drive the higher priority IO scanner task and EIP EM processing

This will guarantee the CPU bandwidth for processing the IO input packets and IO output packets in the same task cycle sequentially and 1 EIP EM per ms.

- Allow the user to configure the ICIE switch

The purpose is to reduce the CPU bandwidth used to process those unwanted packets. The options for the users to configure the ICIE switch are: QoS enable, IGMP snooping enable, port ingress rate limiting set up, and message type filtering and dropping [16].

- Support QoS

This will reduce the IO packet transaction time through the infrastructure devices in the network. In the receiving side, each four priority IO receive queues are designed for EIP IO and MB/TCP IO packets, respectively, as shown in Fig. 4.

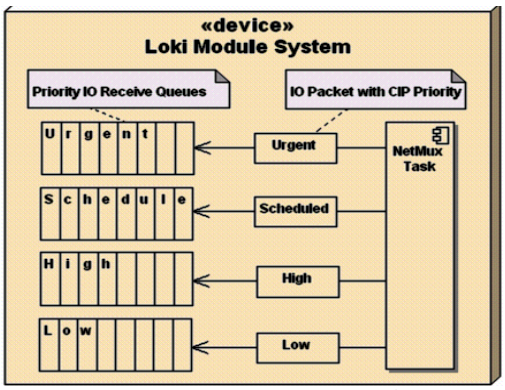

Fig. 4. The priority IO receive queue architecture of the ICIE module.

The priority IO receive queue architecture is only useful for the cases of smaller PLC MAST task cycle time and traffic overload. As discussed in the above section, the PLC may not read the most updated IO data from some IO devices if the PLC MAST task cycle time is below 20 ms in the required IO throughput condition [17]. In this case, assigning the important IO connections with higher priorities result in those corresponding IO input data being put in the higher priority receive queue, processed at the beginning of the IO scanner's cycle, and put into the input buffer before the PLC MAST task goes into its IN state. Thus, the priority IO receive queue architecture and QoS feature will guarantee the PLC to read the most updated IO data from higher priority IO devices in the condition of smaller PLC MAST cycle. In the traffic overload case, the rate limiting only allows the higher priority IO messages to be received.

For the case of the PLC with larger MAST task cycle time, the IO scanner task has enough time to process all the IO packets, therefore, the priority IO receive queue architecture may not provide any advantages.

IV. QOS ARCHITECTURE

As required by ODVA EIP Specification, the ICIE module architecture adopts the standard IEEE 802.1D/Q tag and the Differentiated Services (DiffServ) QoS mechanisms

to mark different application message packets with different relative priorities.

A. IEEE 802.1D/Q tag Format

With 802.1Q tag, the frame is identified as belonging to a specific subscriber on the Ethernet network and is specified to have a certain level of priority for the Ethernet switch to determine where and how the frame is to be delivered. Fig. 5 shows the frame structure of the Ethernet II with 802.1Q Tag. The 802.1Q Tag header is 4 bytes. The total frame size of the Ethernet II with 802.1Q Tag is from 64 bytes to 1522 bytes [18].

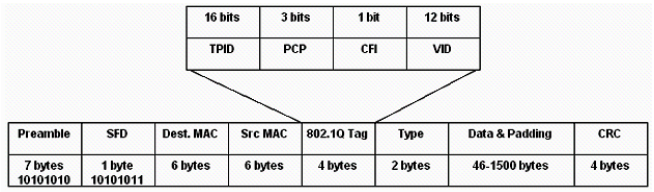

Fig. 5. The frame structure of the Ethernet II with 802.1Q Tag.

- Tag Protocol Identifier (TPID): a 16-bit field set to a value of 0×8100 in order to identify the frame as an IEEE 802.1Q-tagged frame. These two byte value 0x8100 must never be changed when creating the 802.1Q tagged frame.
- Priority Code Point (PCP): a 3-bit field which refers to the IEEE 802.1p priority. It indicates the frame priority level from 0 (lowest) to 7 (highest), which can be used to prioritize and forward different classes of traffic by the switch. The PCP field should be set based on the user input for each Ethernet frame to be sent. This will be addressed in detail in section 8.2 QoS.
- Canonical Format Indicator (CFI): a 1-bit field. If the value of this field is 1, the MAC address is in non-canonical format. If the value is 0, the MAC address is in canonical format. It must be always set to zero in the ICIE module when creating the 802.1Q tagged frame. CFI is used for compatibility between Ethernet and Token Ring networks. If a frame received at an Ethernet port has a CFI set to 1, then that frame should not be bridged to an untagged port.
- VLAN Identifier (VID): a 12-bit field specifying the VLAN to which the frame belongs. A value of 0 means that the frame doesn't belong to any VLAN; in this case the 802.1Q tag specifies only a priority and is referred to as a priority tag. A value of hex FFF is reserved for implementation use. All other values may be used as VLAN identifiers, allowing up to 4094 VLANs. On bridges, VLAN 1 is often reserved for management.

With the IEEE 802.1D/Q tag, the ICIE module can mark any Ethernet II and 802.3 types message packets, including the non-IP RSTP BPDU packets, non-IP ARP packets, IP packets, and so on.

B. Differentiated Services (DiffServ) Format

The Differentiated Services (DiffServ) only specifies the relative priority of IP type message packet in the original type of service (TOS) field of the IPv4 header (in this version of ICIE architecture, no IP V6 support). With the Differentiated Services mechanism, the original TOS field is called DS field now and the first most significant 6 bits of it are used to mark the packet and called DiffServ Codepoint (DSCP) as shown in Fig. 6. The lower 2 bits are not used currently. The network node devices can route the IP packets based on the DSCP and the defined Per-Hop Behavior (PHB) characteristics.

The DSCP marking is not suitable to the non-IP RSTP BPDU and ARP packets.

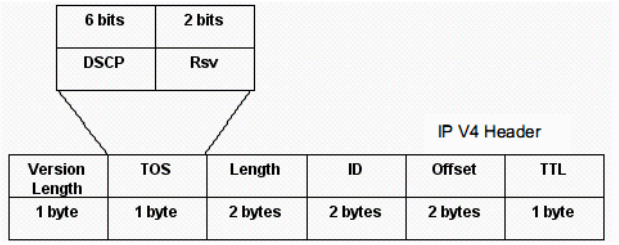

Fig. 6. The DS field in IP V4 header.

C. Message Classification

In this version of the ICIE module architecture, three categories of messages are handled, real time IO data, non-real-time standard network protocol messages, and real time management frames.

The real time IO data are the EIP IO data and the MB/TCP IO data. These IO data need to be transmitted with high throughput, lower responses time, and little or no packet loss. Therefore, they should have higher priority in the QoS mapping.

The non-real-time messages are mainly responsible for the system configuration, diagnosing, monitoring, device parameterization, negotiation of communication connection for transmission of data, and acyclic data exchange. In this architecture design, they are exchanged on the standard channel (that is, going through the normal TCP/IP stack) [19]. The non-real-time messages the ICIE module supports include: EIP EM, MB/TCP EM, HTTP messages, DHCP messages, FTP messages, SNMP messages, TFTP Messages., ICMP Messages, and ARP Messages. These non-real-time messages should be transmitted in QoS lower priority.

The network management frames are mainly used to monitor and manage the network topology changes which require the ICIE module to react at a real-time way. The network management messages the ICIE module supports include RSTP BPDU messages and IGMP Messages. They should have the QoS highest priority.

Based on the ODVA EIP Specification and the consideration of the ICIE module architecture, the default mapping of the ICIE module messages to DSCP and 802.1D priority is shown in Table III. In this version of the ICIE module architecture, the real mappings of both EIP IO and MB/TCP IO messages should be user configurable through the unity Pro and QoS object of the EIP Stack subsystem. But for all the other messages, the ICIE module should use the default mapping values in Table III [20].

In this architecture design, the NetMux component acts as an ingress policy executer to perform the message classification on the receiving side. For the management message frames, the Netmux component should ignore the priority value in the 802.1Q tag (for both RTSP BPDU messages and IGMP packets) and the DSCP value (for

IGMP only) because they will process with their corresponding higher priority tasks and only one queue for each type. For the non real-time message frames, the Netmux component should also ignore the priority value in the 802.1Q tag and the DSCP value because they are lower priority.

TABLE III: THE DEFAULT DSCP AND 802.1D MAPPING FOR ICIE MODULE MESSAGE TRAFFICS

Traffic Type	CIP Priority	DSCP	802.1D Priority	Traffic Usage
RSTP BPDU	n/a	n/a	7	Maintain network topology
IGMP Query	n/a	59 ('111011')	7	IGMP Snooping
IGMP Report	n/a	55 ('110111')	6	IGMP Snooping
CIP class 0/1 MB/TCP IO	Urgent (3)	55 ('110111')	6	CIP motion
	Scheduled (2)	47 ('101111')	5	IO
	High (1)	43 ('101011')	4	IO
	Low(0)	31 ('011111')	3	HMI IO access
CIP UCMM CIP class 3 All other Ethernet/IP encapsulation messages	All	31('011011')	3	CIP messaging
ARP message	n/a	n/a	3	Address Resolution
All other messages	n/a	31 ('011111')	3	Other protocol messages, for example, HTTP, SNMP, FTP, and so on

For the IO data, the NetMux component should check the priority value in the 802.1Q tag and the DSCP value and put the IO packet into the corresponding priority receive queue. If both the priority value in the 802.1Q tag and the DSCP value exist in the received IO packet, the NetMux component should use the DSCP value for EIP IO packet and the priority value in the 802.1Q tag for the MB/TCP IO packet.

If the higher priority queue is full, the NetMux component should put the received IO packet to the next higher priority queue and should put it in the front. If all the four receive queues are all full, the NetMux component should drop the IO packet, which should rarely occur.

D. Message Marking

In this architecture design, the NetMux component acts as an egress policy executer to mark the messages to be sent out the ICIE module with DSCP values and 802.1D priorities if tagged. For the EIP IO and MB/TCP IO packets, the NetMux should not perform the marking, because these IO packets are predefined and already marked with the right DSCP values and 802.1D priorities if tagged. For all the other packets, the NetMux marks them with the default values as shown in Table III. No matter whether the message frames to be sent are tagged or not, the DSCP values should always be marked because many switches in the network use the DSCP value to remark the non tagged frame and schedule it accordingly.

E. QoS Object Conflict

ODVA EIP Specification requires any Ethernet/IP end devices to support the QoS feature and provide a QoS object. Through the QoS object, the user can configure the devices to send tagged IO message frames or non tagged IO packet frames with the "Clss 0/1 Tag Enable" attribute. If this attribute is enabled (set its value to 1), the ODVA EIP Specification requires the device to send the 802.1Q frames for all CIP transport class 0/1 connections. But this specification conflicts with this ICIE module architecture design. Because the ICIE module needs to communicate with many different IO devices concurrently, some of which only accept 802.1Q tagged frames and some of which do not, in this version of the ICIE module architecture, when the "Class 0/1 Tag nable" attribute is set to 1, it indicates that the ICIE module can send the 802.1Q tagged frames to the 801.1Q IO devices and at the same time send non 802.1Q frame to the other non 802.1Q devices [21].

V. CONCLUSIONS AND PERSPECTIVE

We proposed a new mechanism to minimize the congestion which is based on the taking an adaptive decision during transferring multicast messages. Proposed approach is that a device requesting to start and stop the reception of the multicast streams is accomplished through IGMP join and Leave message requests. The IGMP Snooping component monitors (snoops) these join and leave messages to allow it to know which streams to prune from which ports. This process uses a device performing a manager role to periodically query all devices in the subnet and subsequently cause them to re-join the multicast group of listeners for any stream in which they may be interested. The management role is known as an "IGMP Snooping Querier" and it is a service provided by most managed Ethernet switches. However, the ICIE does not provide this capability and requires that another device in the network supports the querier functionality. It is through the external devices solicitation of join messages that allow the IGMP Snooping component to correctly decipher on which ports the downstream listeners are connected.

ACKNOWLEDGMENT

The authors wish to thank my course mate Wen Chuan. This work was supported in part by a grant from Chongqing Engineering Laboratory for Detection, Control and Integrated System, Chongqing Technology and Business University under contract No. DCIS20150302.

REFERENCES

[1] *Specifications for 2.3GHz band Portable Internet Service*, TTAS KO-06 0064 TTA, April 2004.
[2] A. Dutta, J. Chennikara, and W. Chen,"Multicasting streaming media to mobile users," *IEEE Communications Magazine*, vol. 41, no. 10, pp. 81-89, Oct. 2003.
[3] A. Dutta, S. Das, W. Chen, and A. MacAuley, "MarconiNet supporting streaming media over localized wireless multicast," in *Proc. WMC'02*, Sept. 2002, pp. 61-69.
[4] B. Fenner, H. He, B. Haberman, and H. Sandick, "IETF draft: IGMP/MLD-based multicast forwarding," *IETF*, Apr. 2004.
[5] B. Liang and J. Haas, "Predictive distance-based mobility management for multidimensional PCS networks," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, Oct. 2003.
[6] C. Cho, S. Jun, E. Paik, and K. Park, "Rate control for streaming services based on mobility prediction in wireless mobile networks," in *Proc. IEEE WCNC05*, Mar. 2005.
[7] Legout and E. Biersack, "PLM: Fast convergency for cumulative layered multicast transmission schemes," in *Proc. ACM SIGMETRICS'2000*, Santa Clra, CA, USA, June 2000, pp. 113-22.
[8] M. Jain and C. Dovrolis, "End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput," *IEEE/ACM Trans. on Networking*, vol. 11, no. 4, pp. 537-549, 2003.
[9] M. Welz, *Network Congestion Control Managing Internet Traffic*, Wiley, India, 2005, pp. 7-15, 69-77, 93-96.
[10] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven layered multicast," in *Proc. ACM SIGCOMM*, August 1996, pp. 117-130.

[11] Q. Zhang, Q. J. Guo, Q. Ni, W. W. Zhu, and Y. Q. Zhang, "Source adaptive multi-layered multicast algorithms for realtime video distribution," *IEEE/ACM Transactions on Networking*, vol. 8, no. 6, pp. 720-733, 2006.
[12] S. Kumar, P. Radoslavov, D. Thaler, C. Alaettinoglu, D. Estrin, and M. Handley, "The MASCBGMP architecture for inter-domain multicast routing," in *Proc. ACM SIGCOMM*, April 1998, pp. 93-104.
[13] S. Johansen, A. N. Kim, and A. Perkis, "Quality incentive assisted congestion control for receiver-driven multicast," in *Proc. IEEE International Conference on Communications*, 2007, pp. 1642-1647.
[14] S. Deering, "Multicasting routing in internetwork and extended LANs," *ACM Transactions on Computer Systems*, vol. 8, pp. 85-110.
[15] J. Byers, M. Frumin, *et al.*, "FLID-DL: Congestion control for layered multicast," in *Proc. NGC2000*, Palo Alto, USA, Nov. 2000, pp. 71-81.
[16] M. Johanson, "Scalable video conferencing using subband transform coding and layered multicast transmission," in *Proc. International Conference on Signal Processing Applications and Technology*, Orlando, Florida, Nov. 1-4, 1999.
[17] K. Singh, R. S. Yadav, M. Manjul, and R. Dhir, "Bandwidth delay quality parameter based multicast congestion control," presented at the International Conference on Advanced Computing and Communication, Department of Information Technology, MIT, Anna University, Chennai, 2008.
[18] J. Kimura, F. A. Tobagi, J. M. Pulido, and P. J. Emstad, "Perceived quality and bandwidth characterization of layered MPEG-2 video encoding," in *Proc. the SPIE International Symposium on Voice, Video and Data Communications*, Boston, Sept. 1999.
[19] G. I. Kwon and J. Byers, "Smooth multirate multicast congestion control," in *Proc. IEEE Infocom*, March 2003, vol. 2, pp. 1022-1032.
[20] L. Vicisano, L. Rizzo, and J. Crowcroft, "TCPlike congestion control for layeredmulticast data transfer," in *Proc. Conference on Computer Communications*, March 1998, pp. 996-1003.
[21] A. Legout and E. W. Biersack, "Pathological behaviors for RLM and RLC," in *Proc. International Conference on Network and Operating System Support for Digital Audio and Video*, Chapel Hill, NC, USA, June 2000, pp. 164-172.

Lu Sheng was born in 1974. He is an associate professor, a graduate student instructors, and a post-doctoral fellow. In July 2006, he graduated from the School of Microelectronics and Solid State Electronics, Tianjin University and received a doctorate degree in science. His main research directions are new functional materials and new electronic components. He had published more than 70 papers in academic journals.