Design of a New Chip and a Switch Architecture for a Home Gateway

Kwang-Soon Choi, Chan-Soo Yoon, and Kwang-Mo Jung

Abstract—As Internet is becoming popular to everyone recently, demands for higher-quality services such as VOD and home networking have been increasing. Especially, home networking system can interconnect and control home appliances which use different protocols via Internet. This means that a common protocol to communicate with each other and a new system architecture to implement the common protocol are needed. In this paper, we propose a common protocol, a novel chip architecture and a memory management scheme for packet-switching of a home gateway system.

Index Terms—Common Protocol, Home Gateway, Home Network, Shared-Memory Switch

I. INTRODUCTION

THE Internet is so popularized to everyone in every-where recently that the demands for accessing home appliances via Internet and communicating between them are increasing. As a result, new researches and efforts about home gateway (HG) and residential gateway (RG) has been made. In these systems, the most important thing is the interconnection between WAN and LAN at home. xDSL and cable modem can be WAN interfaces, and wireless LAN, IEEE1394, LonTalk, bluetooth, HomePNA and HomeRF can be LAN interfaces in the home. IEEE1394 is the protocol for the control of audio and video devices and the real-time data transfer between them [1]. Bluetooth is for the access of home appliances and low-rate data transfer between them like IEEE1394 [2]. Unlike, LonTalk is only for the control of home appliances [3].

All these protocols are targeted to their own special application areas such as high- or low-speed data transfer and control data transfer. Thus, differences and characteristics of these protocols should be considered in the design of HG or RG system to communicate with each other without failure.

In this paper, we propose a new HG chip architecture which supports IP, IEEE1394, LonTalk and Bluetooth protocols, a common protocol (CP) and a memory management scheme for packet-switching. In section II, a recommended system architecture for a HG is described. In section III, a new chip architecture and a CP for packet conversion are described. In section IV, a switch block and a common bus architecture for packet-switching are described.

II. ARCHITECTURE FOR A HG

Fig. 1 shows a general architecture for a HG which is proposed by ISO/IEC JTC1/SC25 WG1 [4]. This architecture consists of three main parts - Gateway Internal Protocol (GIP), WAN Gateway Interface (WGI) and LAN Gateway Interface (LGI). Both WGI and LGI have their own local processor and memories used to convert incoming packets of a specific protocol to CP packets. And then a Common Bus in GIP switches them which flow from GIP Common interface to another GIP Common interface of a target node which converts incoming CP packets to its corresponding protocol packets. Thus, these consecutive operations make it possible to communicate between interfaces using different protocol. But in reference [4], no detailed specification of each H/W block and CP is given.

Especially, protocol conversion is a main problem in Protocol Engineering since protocol standards include a very large number of options to take care of different service possibilities and to please all the people involved in the Standards Committees. More importantly, it takes polynomial (rather than exponential) time and space in the size of the protocol specifications [5]. Thus, in designing this system, the important things to consider are followings:

- design of an efficient CP
- algorithm for converting to the CP
- management of traffic with a different speed
- algorithm for QoS
- algorithm for security

Fig. 1. General Architecture for a HG

K. Choi is with Korea Electronics Technology Institute (KETI), Kyunggi-Do, 463-771 Korea (telephone: +82-31-780-7040, fax: +82-31-780-7060, e-mail: lenon@keti.re.kr).

C. Yoon is with Korea Electronics Technology Institute (KETI), Kyunggi-Do, 463-771 Korea (e-mail: yooncs@keti.re.kr).

K. Jung is with Korea Electronics Technology Institute (KETI), Kyunggi-Do, 463-771 Korea (e-mail: jungkm@keti.re.kr).

• algorithm for packet-switching

III. NEW CHIP ARCHITECTURE

A. Chip Architecture

Fig. 2 shows the proposed chip architecture. In this chip, incoming packets pass through two packet converters which are Anything-to-CP converter and CP-to-Anything converter. Each of them consists of a Header Converter and a Payload Converter. The Header Converter reads Address Table with its original address and gets new address which was already set when this chip started.

The Payload Converter which is similar to a RISC processor operated by conversion codes, which was already micro-coded in a non-volatile memory such as ROM or PLA, converts incoming payload to CP or target protocol payload. It processes an incoming packet at the layer 2-4 level simply and at the layer 5-7 level for more complex operations such as packet conversion [6]. And it has its own local memory or register files. The exact conversion algorithm is still being researched.

In this system, the memory management scheme is important for a perfect cycle from reception to transmission of a packet. Considerations related to memory management are followings:

- position of buffers (internal or external)
- depth of buffers
- method for fast transmission of a packet between two packet-processing blocks

Fig. 2. Chip Architecture

Each Interface block interfaces external PHY/LINK chips. When processing incoming packets, it buffers eight serial bits and sends one-byte to an Input Buffer. When processing outgoing packets, it buffers one-byte and sends it to external PHY/LINK chips with the form of serial bit stream. Input/Output Buffers can buffer a whole incoming or outgoing packet which has maximum payload size. And a Traffic Management makes a Buffer Controller control a packet by the traffic management algorithm. It passes or discards a buffered packet. A Segmentation block segments an incoming CP packet into several 256-byte packets, which helps a Switch block to access shared memories. Using a fixed-size packet offers not only more easier and faster way, but also more higher memory-utilization way to a QoS/Priority Controller and a Scheduler. A Reassembly block assembles incoming switched packets and sends one assembled CP packet to a CP-to-Anything Converter according to the information in the CP header.

The architecture of a Switch Block and a shared memory is followed in the next section.

B. Common Packet Protocol

Fig. 3 shows the CP packet format, which can be modified according to the variations of chip architecture, conversion algorithms, traffic management, QoS, security and so on. Since we do not wholly focus on to the CP packet in this paper, the basic CP packet format is described.

![Fig. 3. Common Protocol Packet Format](image)

The packet size of each protocol that this chip supports is considered in Table 1. The payload size of the CP packet is important. It would be better for the CP packet to be able to contain an average size of all packets for the performance of the chip.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Max. Payload Size (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>1,500 (header included)</td>
</tr>
<tr>
<td>IEEE 1394</td>
<td>16,384 @ 3.2Gbps (asynchronous)</td>
</tr>
<tr>
<td></td>
<td>32,768 @ 3.2Gbps (isochronous)</td>
</tr>
<tr>
<td>LonTalk</td>
<td>229 (APDU)</td>
</tr>
<tr>
<td>Bluetooth</td>
<td>1691 (B-PAN)</td>
</tr>
</tbody>
</table>

The CP packet consists of header of 16 bytes and maximum payload of 2,048 bytes. This can contain not only any datagram of IP, LonTalk and Bluetooth but also an asynchronous packet of 2,048 bytes at 400Mbps rate and an isochronous packet of 2,048 bytes at 200Mbps rate in the case of IEEE 1394. In addition, Sequence and Last fields in the CP packet make it possible to carry a long packet over 2,048 bytes by fragmentation.

![Table 1. Maximum Payload Size](image)
IV. SWITCH ARCHITECTURE

This proposed switch architecture requires two external memories: one for storing of CP packets and the other for linked lists, QoS buffer and priority buffer. Both of them share an address bus and a data bus, and can be accessed by /CSn, /WEn, and /OEn (n = 0 or 1). These QoS/Priority Buffers and linked lists buffer which index the CP packet memory use the same external memory partitioned into three buffers.

Then the size of a CP packet is 256N bytes, where N is the maximum number of segmented packets which can be stored in a CP packet memory. And the size of the other one is 108N (12N for linked list buffer, 48N for QoS buffer and 48N for Priority buffer).

A. CP Packet Memory

Fig. 4 shows a CP packet memory. N Segmented CP packets can be stored in a CP packet memory, and Scheduler makes Reassembly block read them according to the adaptive-QoS algorithm which requires external QoS and Priority buffer.

B. Linked List Buffer

Linked List Buffer contains N linked lists. Each of them is composed of three fields as shown in Fig. 5: Previous Linked List Address (PLA), Packet Memory Address (PMA) and Next Linked List Address (NLA). Each consists of four-byte address field, which uses 32-bit addressing mode. These three fields indicate an address of a previous linked list, an address of a segmented CP packet and an address of a next linked list. These linked lists can be divided into two lists, Free-Space Linked Lists and Used-Space Linked List. And a header-pointer and a tail-pointer are used to point the header and the tail of each list respectively. All lists of Free-Space Linked List point to the free space of CP packet memory. And all lists of Used-Space Linked List point to the used space of CP packet memory.

Fig. 6 shows a CP packet memory and linked list buffers. In order to utilize these linked lists, they should be initialized as shown in Fig. 7. Initially all lists should be linked as Free-Space Linked Lists.

In case of storing a packet to CP packet memory, the header of Free-Space Linked Lists should be read before storing a segmented CP packet to a CP packet memory. This header indicates the location where an incoming packet is stored. If a packet is stored, the first list of Free-Space Linked Lists should
be detached and then attached to the end of Used-Space Linked Lists as shown in Fig. 8.

In case of reading a packet from a CP packet memory, the corresponding linked list of Used-Space Linked Lists should be detached and then attached to the end of Free-Space Linked List as shown in Fig. 9.

Fig. 9. Updating Linked Lists (After Reading a Segmented CP Packet)

C. QoS and Priority Buffers

This chip uses adaptive-QoS algorithm which is especially proposed for a HG system. Since this paper is not focused on the QoS algorithm, a brief introduction will be followed.

All incoming packets with different protocol format are classified into three classes: class 1 (control data), class 2 (multimedia data) and class 3 (Internet data). These classified packets are not switched directly in the switch block, but reordered in the Priority Buffer according to the line state. Fig. 10 shows a QoS buffer, a Priority Buffer and a format of each entry.

This format of QoS/Priority entry is similar to the format of linked lists. Once a segmented CP packet is stored, the corresponding linked list is updated. And then the information and an address of this linked list are stored to a QoS Buffer according to the QoS field in CP packet header and reordered in a Priority Buffer according to the priority which is decided by the line state.

A Scheduler reads a PMA field of one entry in Priority Buffer and finds the destination address in a CP packet header. And Bus Controller reads the Bus Status register and decides if it sends a /BGn towards the Reassembly block of destination node or not.

D. Memory Controller

Fig. 11 shows the timing diagrams for accessing two external SRAMs. Each access requires 780ns and consists of four steps: storing (or reading) a segmented CP packet, updating linked lists, updating a QoS Buffer and updating a Priority Buffer.

A Memory Controller interfaces two external SRAMs via 32-bit address bus, 32-bit data bus and control signals such as chip select (/CSn), write enable (/WEn) and output enable (/OEn) according to the contents of internal ring counter. n indicates a CP packet memory (n=0) or the other one(n=1).

E. Common Bus and Bus Controller

A Common Bus provides a common path between SARs and a Switch Block. Converted packets flow from a SAR to a Switch Block through this bus and then are forwarded to another SAR of a destination node by the Switch Block. This bus consists of 32-bit data bus and control signals such as /BRn.
In this paper, we have proposed a new chip architecture, a CP, a shared memory architecture for packet-switching and a common bus architecture of a HG system.

This chip uses IP for WAN interface, and IP, IEEE 1394, LonTalk and Bluetooth protocols for LAN interfaces. And it consists of two packet converters which have a header converter and a payload converter. These converters have the architecture of RISC microprocessor operated by the micro-coded conversion algorithms. The CP packet format has been designed but not fixed yet. And the management of shared memory by using linked lists and the adaptive-QoS is designed.

We conclude that the packet-conversion time, switching time and the packet forwarding time can meet the conditions required on this proposed chip architecture. And the complexity and the cost of the chip can be reduced by using external SRAMs for switching packets.

And not only the algorithms for packet conversion, traffic management and security but also the common protocol and the implementation of packet converters are being studied.

V. CONCLUSIONS

To estimate chip performance, we assumed that there income packets with maximum payload size at the full line-speed on each interface and RISC CPUs in CP-to-Anything (or Anything-to-CP) Converters can convert payload in 512 machine cycles. And internal chip architecture can be divided into four logical blocks, i.e. an interface, an Anything-to-CP (or a CP-to-Anything) converter, a SAR and a Switch Block. Table 2 shows the packet-processing performance of each internal logical block.

In order to process incoming packets without failure, it is important to estimate the packet-processing performance \( P_n \) of each internal block and to make sure that \( P_1 \leq P_2 \leq P_3 \leq P_4 \).

REFERENCES