Proceedings of IMECE2004
2004 ASME International Mechanical Engineering Congress and RD&D Expo
November 13-19, 2004, Anaheim, California USA

IMECE2004-60064

HIGH SPEED RING-BASED DISTRIBUTED NETWORKED CONTROL SYSTEM FOR REAL-TIME MULTIVARIABLE APPLICATIONS

Sunghoon Kim

Graduate Research Assistant
Department of Mechanical Engineering
2168 Etcheverry Hall Berkeley, CA 94720
University of California, Berkeley
shkim@me.berkeley.edu

H. Kazerooni

Professor
Department of Mechanical Engineering
5124 Etcheverry Hall Berkeley, CA 94720
University of California, Berkeley
kazerooni@me.berkeley.edu

ABSTRACT

A networked control system (NCS) is a control architecture where sensors, actuators and controllers are distributed and interconnected. It is advantageous in terms of interoperability, expandability, installation, volume of wiring, maintenance, and cost-effectiveness. Many distributed network systems of various topologies and network types have been developed, but NCS systems tend to suffer from such issues as nondeterminism, long network delays, large overheads and unfairness.

This paper presents the ring-based protocol, called the ExoNet, and its network architecture which are built to achieve better performance as a distributed networked system. A Cypress transceiver CY7C924ADX is applied to the network as a communication unit. The protocol is based on the transceiver and developed to achieve fast communication and allowable latency for controls with high control loop frequency. Compared with other standard network types such as Ethernet, ControlNet or DeviceNet, the network is characterized by its ring-based architecture, simple message and packet formats, one-shot distribution of control data and collection of sensor data, multi-node transmission, echo of a message, and other features. The network also guarantees determinism, collision-free transmission, relatively small overhead, fairness between nodes and flexibility in configuration. Its analysis and comparison with these network types are also provided and its application on the Berkeley Lower-Extremity Exoskeleton (BLEEX) is described.

Keywords: ring architecture, distributed networked control system, BLEEX

INTRODUCTION

Traditional centralized control architectures where a supervisory controller directly interfaces in a point-to-point fashion with all sensors and actuators in the system have been successfully implemented in the past. They are generally feasible when a controller interfaces with small number of sensors and actuators and requires short wiring to them. However, it is often difficult to add, remove or reconfigure components in these systems. Moreover, the recent trend of control systems is dealing with much more complicated systems than before, so these systems require the control system to be easily reconfigurable, expandable and maintainable. Hence, the networked control system (NCS) is utilized as an alternative to the conventional centralized control system because of its advantages in flexibility, volume of wiring and capacity of distribution [1].

Various distributed NCSs have been applied in fields such as industrial automation, building automation, office and home automation, intelligent vehicles, and aircrafts and spacecrafts. [2]-[6]. The type of NCS that is used for an application is determined by the network type and its architecture. NCSs using the Ethernet bus with carrier sense multiple access with collision detection (CSMA/CD), token-passing bus (e.g. ControlNet), and controller area network (CAN) bus (e.g. DeviceNet) were described and compared [1]. Their properties are summarized in the Table 1 in the following section. They are based on the common bus architecture where all components are wired to one common shared bus. This architecture generally includes constraints; for example, transmitting nodes on the bus must not transfer messages simultaneously. Collision of messages will result in data loss, failure of data transfer and penalty for retransmission, which makes the network nondeterministic. And the common bus may also be occupied and saturated with data by high-priority components, so fairness to lower-priority components is another issue in this architecture. Thus, it is important how a NCS assures the determinism and fairness. There are more applications of network types such as a process field bus (PROFIBUS) [7], manufacturing automation protocol (MAP) [8], and fiber distributed data interface (FDDI) [9].

A NCS network has to be considered alongside the unavoidable network latencies caused by the transfer of information between a central controller and sensors /actuators. Because the performance of a NCS is determined by the sampling time and time delays [10], the tradeoff between network traffic and sampling periods must be considered. Generally, the performance of a digital control system increases as its sampling frequency increases. The performance of a NCS similarly increases as a sampling frequency increases until the network traffic becomes saturated. Since a high ratio of the network delay to the sampling period results in performance degradation and instability of the system, the performance of the NCS deteriorates after the best performance point at an optimal sampling frequency is reached [11]. There have been attempts to reduce network delay to improve the performance of a NCS so that it can approach that of the digital control system and obtain the higher optimal frequency. The implementation of adjustable deadband is introduced as a solution of reducing network traffic [12]. Several software changes have been suggested to attain the bounded network delay using Ethernet [13].

In this paper, we will discuss the ring-based distributed NCS with its defined protocol. By the virtue of rapid growth of modern technologies, we could develop its architecture using a high-speed HOTLink transceiver, called CY7C924ADX and manufactured by Cypress. The NCS has been designed in the way to minimize network latency, guarantee fairness and collision-free operation, and achieve communication integrity in the control system between a single controller and several slave nodes that interface with many allocated sensors and actuators. We compare its performance with other common network types such as Ethernet, ControlNet and DeviceNet, and discuss its competitiveness and effectiveness with an application on the Exoskeleton.

Network Control System

HOTLink Transceiver CY7C924ADX

A 200Mbaud Cypress CY7C924ADX Hotlink [14] transceiver is selected for the communication building block that is capable of point-to-point communication and allows data transfer over high-speed serial links at speed ranging from 50 Mbaud to 200 Mbaud. The HOTLink transceiver is a device in which traditional parallel interfaces can be replaced with high-speed serial links. It also supports daisy-chain and ring topology, which is a major reason why it is used.

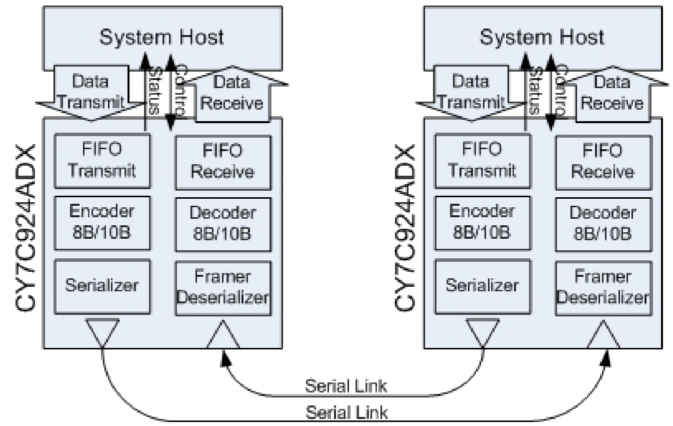

Fig. 1. HotLink System Connections

The structure of a transceiver is illustrated in Fig. 1. The main function of a transceiver as a transmitter is to accept 8-bit or pre-encoded 10-bit parallel characters on each reference clock time and transmit them serially bit by bit on each bit time. This bit rate is generated from the reference clock rate by a integrated clock multiplier. As a receiver, it accepts serial bit-stream and reconstructs them into parallel data at the recovered clock rate from bit-stream. Thus, it is a serial communication block interfacing with parallel data.

Inside the transceiver, the internal integrated 8B/10B encoder converts 8-bit data into encoded 10-bit data, the Transmit FIFO stores data from the host and the Receiver FIFO stores data from bit stream. They are enabled or bypassed according to a user's configuration. These FIFOs provide synchronous mode with internal components of a transceiver if bypassed or asynchronous mode if enabled. The encoder is enabled when the transceiver is in 8-bit parallel data mode or bypassed when it is in pre-encoded 10-bit data mode. The encoded 10-bit data in the form of bit stream is transferred through a serial link between two transceivers.

One of the advantageous features of this transceiver is that it can transmit either a special character code or a data code. This advantage results from encoding redundancy when the encoder converts 8-bit data and control signals to 10-bit encoded transmission code with the encoder enabled. On a receiver, a special character is outputted with a signal indicating that it is received. A special character code is distinguishable from data codes so it can be used as a delimiter in the message frame between two nodes.

In our architecture of the NCS, a 8B/10B encoder and all FIFOs are enabled, which provides most powerful operating mode of the transceiver.

Ring Topology

The ring topology architecture of our NCS is illustrated in Fig. 2. It consists of one master node and several slave nodes, here, N slave nodes, as shown in Fig. 2. Each serial link consists of 2 differential pair lines and additional power lines that are not depicted in Fig. 2. These differential pairs are used for high-speed communication in both transmission and reception.

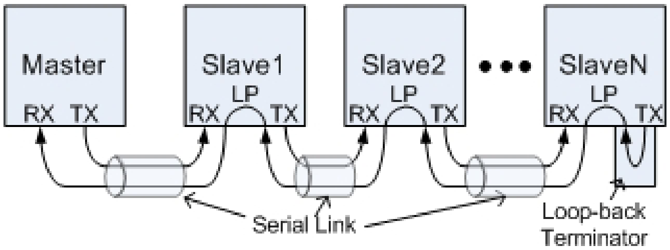

Fig. 2. Ring Topology

In building the ring topology using the transceiver, the internal loop (LP) lines are introduced in slave nodes. The direction of data stream flow in this network is hence the transmit port (TX) of the master node → the receive port (RX) of the slave1 node, the TX of the slave1 node → the RX of the slave2 node, and so forth. This path continues to the RX of the slaveN node. A loop-back terminator completes the ring by plugging into the TX of the slaveN, the last slave on the ring, so that data which leaves the TX of the slaveN can arrive at the RX of the master, passing through all LP lines of the slave nodes.

This ring topology provides flexibility and expandability for the network by easily adding or removing slave nodes, and then plugging a loop-back terminator at the TX of the last slave node on the ring. This eliminates the requirement of a single circulating cable connecting from the last slave node to the master node. It is particularly useful in cabling the network where all nodes are placed on a line rather than on a circle. There needs to be only one serial link cable between any two consecutive nodes and a loop-back terminator on the last node on the ring. However, this scheme may produce a bottleneck in transmission due to the longest path from the TX of the last node to the RX of the master node. This path is almost equal to the sum of all other paths from the TX of one node to the RX of another node on the ring. The propagation time through this path, thus, is the longest and the sum of the propagation time of all other paths but it is much smaller than other delays and not noteworthy in systems within a small area. This delay will be discussed in the following section.

Protocol Design

Packet Frame Format

A packet is defined as a basic element of a message in our protocol, and contains one valuable piece of information such as actuation data, sensor data, command data, or error data. The packet frame is depicted in Fig. 3.

TYPE	ADDR	DATA	CRC

Fig. 3. Packet frame format

The packet frame includes 4 fields: TYPE, ADDR, DATA and CRC. The TYPE determines what data in a packet frame represents; for example, actuation data, sensor data, command data, or error data. The size of the TYPE field is decided by the number of possible packet types in protocol design. In general, the maximum number of packet types in an actuation-sensor control system are the four types listed above, all of which require a 2-bit TYPE field. The second field contains the ADDR bits, which indicate an implicit logical address or an ID of a slave node on the ring. This ADDR field represents the source slave address that a packet has originated from and the target slave address that a packet is directed to. This is dependent on the TYPE field of a packet, so it indicates the source address when it is a packet of sensor or error data and the target address when a packet for actuation data or command data. The size of the ADDR field in design is decided by the maximum possible number of slave nodes on the ring. The third field contains DATA bits, which include one piece of data information, and the last field is the cyclic redundancy checksum (CRC) for a packet. Large size of CRC bits provides more accuracy and integrity for the error test of a packet, but increases the transfer time. In the application of the Exoskeleton employing the 8-bit parallel data mode using an encoder, the number of bits in a packet is chosen to be a multiple of 8. A packet is thus 24-bit, 3 bytes, such as 2-bit TYPE, 3-bit ADDR, 16-bit DATA and 3-bit CRC in our example for 16-bit actuation and sensor data.

Message frame format

A message is a data stream from a transmitting node to a receiving node. As shown in Fig. 4, each message starts with a Start-Of-Message code (SOM) and terminates with an End-Of-Message code (EOM). Packets that include data are located between these two codes. When a node receives a message, it begins decoding upon detecting a SOM and ends when it detects an EOM. These SOM and EOM delimiters in a message are distinguished from a packet because they are based on special character codes and a packet is based on data codes.

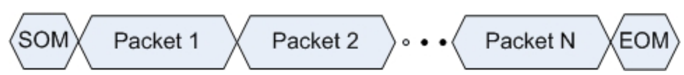

Fig. 4. Message frame format

This message format provides flexibility for the data transferred, so a message can be composed of any number of packets as long as both a transmitter and a receiver retain storage capacity to store them and maintain consistency of the order of data. The number of packets within a message is also realistically limited by the tolerable communication latency in the control system.

Communication procedure

The communication procedures in our NCS can be summarized into four steps: TEST, RESET, ASSIGN IDs, and DATA TRANSFER. During the TEST step, the master tests its subordinate ring network to see if it's properly established or not. This can be done simply by sending a dummy message and waiting for its echo message. If an echo message returns before timeout and is identical with its source message, the network is assumed to be ready for the next steps. As a more complex method, we can make use of the Built-in Self-Test mode, one of the capabilities of the transceiver. In this mode, a transmitting transceiver is enabled to generate a known 511-character repeating sequence, which is compared, character-by-character in a receiving transceiver. Thus, each node can fully test its subsequent node in a round-robin fashion; for example, the master node tests the slave1 node, then if it is verified the slave1 node tests slave2 node, and so forth. At last, the slaveN node tests the master node, and if it is successful, a ring network will be completely verified. During the RESET step, the master node issues a command for requesting all slave nodes to reset their registers or storage elements in order to prevent invalid data from being retained and transferred. This step may be ignored if necessary. The final step for initialization of the network is the ASSIGN IDs to all slave nodes on the ring connected to a single master. The master assigns implicit IDs to each slave node by issuing another command including an ID assigned to a receiving slave node. Starting from the master the 1st subsequent slave node downstream is assigned ID1. Likewise, ID2 is assigned to the 2nd slave and so forth. ID8 is the last ID for the 8th slave using the 3-bit ADDR field defined in the packet format. Therefore, there can be 8 slave nodes at maximum on a ring with 3-bit ADDR field. This automatically assigning ID method network makes the network flexible with the number and the order of slave nodes.

After the initialization, the network is available to exchange data among nodes on the network, a method dubbed DATA TRANSFER. Actual system data including actuation and sensor data or user commands can traverse the network until errors in a message are detected or timeout of the network occurs. Fig. 5 shows an example of a timing diagram during data message transfer on a ring network. As in Fig. 2, this is a case where there is only one master and two slave nodes on the ring structure, and each slave is associated with one piece of actuation data and three pieces of sensor data. In Fig. 5, the "S" denotes the SOM code of a message and "E" the EOM code. "C#" is the designated actuator command packet for the slave# node and "T#%" is the %th sensor data sourcing from the slave# node. For example, "C1" and "C2" are actuation data designated for slave1 node and slave2 node, respectively, and "T11", "T12", and "T13" are the 1st, 2nd and 3rd sensor data generated from the slave1 node.

The fundamental function of the master is to provide actuation data for all slave nodes on the network and receive sensor data from them. It initiates communication by sending actuation data in the form of a packet in a message. The message consists of actuation data packets as many as the number of actuation data for all slave nodes on the ring. In this case, the number of actuation data packets is two, C1 and C2, one for each node. Then when a slave node receives the message, it decodes packets from the message into data. During decoding, if the dedicated actuation data to the receiving slave node is detected, the node stores it and prepares to transmit its sensor data. After original packets are retransmitted to a node in neighborhood toward downstream, corresponding sensor data of the node are transmitted in packet format, too. This occurs when the node receives its own actuation data without an error. When any packet is decoded with a CRC error, it is replaced with an error data packet and then transmitted. Therefore, the function of a slave node can be characterized by sending its sensor data upon detecting its designated actuation data, or otherwise retransmitting a packet as received or an error packet if an original packet contains errors. Likewise, the initial message containing only actuation data packets is augmented with sensor data packets from all slave nodes. After leaving the last node on the ring, this chain of packets in a message then returns to the master, decoded and stored in order at the memory element of the master. The controller can then access the data and use it for computing actuation data of the system at the following control loop cycle.

The advantages of the network can be summarized as follows: first, the network is deterministic without collision of messages because the communication latency, or the network

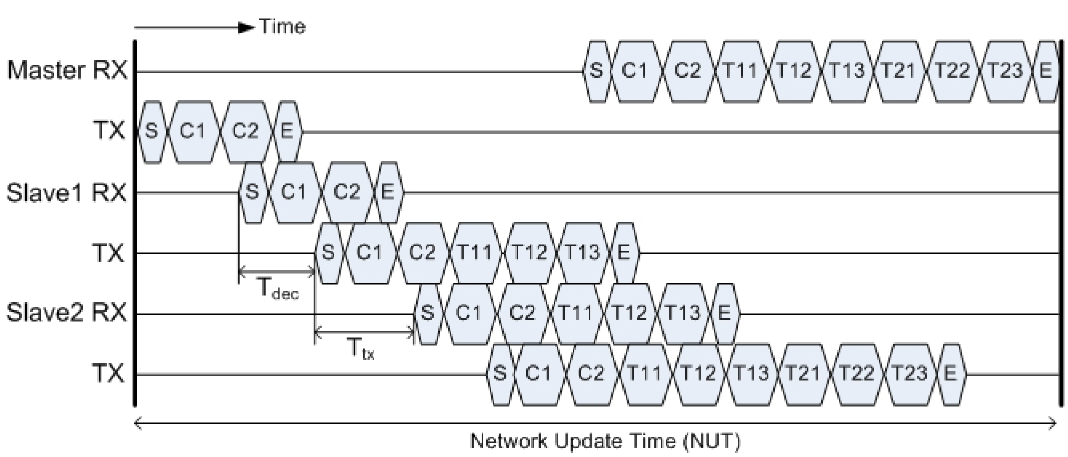

Fig. 5. Timing diagram during messages transfer in the ring-based NCS

update time (NUT), depends on constant parameters such as: the number of nodes, the decoding time of a first packet in a message, T_{dec}, and the propagation delay through a serial link, a transmitting transceiver and a receiving transceiver, T_{tx}, and the size of a returning augmented message to the master which is the function of the number of actuation data and sensor data of slave nodes. The propagation time through serial media, which is one of the factors in T_{tx}, may vary with the length of an interconnecting serial link. It is approximately 1 ns/ft in copper and 2 ns/ft in multi-mode optical cable [15]. Thus, the propagation delay through a few feet of a serial link is not significant compared with propagation delays through a transceiver. Consequently, T_{tx} is bounded and deterministic with the finite length of a link and so the network can be considered to be deterministic. Second, the network easily achieves the reconfiguration by adding, removing, or reorganizing slave nodes. This is possible due to the automatic assignment of IDs to slave nodes during initialization of the network. Another advantage is that fairness is guaranteed as long as the master provides at least one piece of actuation data for each slave nodes. And a message can contain any number of packets unless the NUT is within the network timeout. The overhead in a message is relatively small, especially at low traffic network since the overhead increases with the number of packets. An arbiter for priority arbitration is not needed since fairness is achieved. Next, one-shot of a message from the master enables distribution of actuation data and collection of sensor data. As actuation data circulate the network, they are applied to each individual node and sensor data are collected. Moreover, we reduce network delay by overlapping message transmission of nodes as shown in Fig. 5, so multi-node transmission is possible in this case due to the ring topology. Compared with mechanisms granting permission for one node to transmit at a time in common bus architectures, this scheme is beneficial, especially when the size of a message is large. Besides, independence between packets in a message implies decoupling data information so data in packets with errors can be discarded and data without errors can be stored at a receiver. It may then re-request only these discarded data to their source nodes, which is more convenient than requesting an entire data again. Finally, the echo of a message can also increase safety and integrity of the network by double-checking equality of a source message and its echo along with CRC error checking for each packet in a message.

Table 1. Typical System Properties For control networks

	EtherNet	Control Net	DeviceNet	ExoNet
Data Rate (Mb/s)	10	5	0.5	50~200
Max. Length (m)	2500	1000	100	unlimited[a]
Max. Data size (bytes)	1500	504	8	unlimited[b]
Min. Message size[c](bytes)	72	7	47/8[d]	2[e]
Max. number of nodes	>1000	99	64	8[f]
Advantages	•Simple algorithm •Almost no delay at low network nodes •Good for data transfer	•Deterministic •Reconfigurable •Good at high network loads •Fairness by token rotation	•Deterministic •Optimized for a short message •Fairness by message priority	•Deterministic •Reconfigurable •Fairness by actuation data rotation •Any size of a message •Small overhead at low network load •No arbitrator •One-shot distribution / collection [g] •Multi-node transmission •Independence between packets [h] •Echo of a message
Disadvantages	•Non-deterministic •Unfairness •Message collision •Large overhead	•Most time is spent to pass a token at light network loads when many nodes exist in a ring	•Arbitrator •Slow data rate •Bit synchronization •Support for only 8 bytes data transfer at a time	•Single master only •Increasing delay as the number of nodes increases •Increasing overhead per the number of data •Complex algorithm in a master node

a: there is no intrinsic distance limit in HOTLink, but it is determined by an interconnected media [16]
b: usually limited by allowable network delay and determined by the number of system data
c: zero data size
d: 47bit overhead in DeviceNet
e: one-byte SOM and EOM
f: if the size of ID bits is 3 bit
g: master distributes actuation data and collects sensor data over the network by sending only one message
h: corruption of a packet doesn't result in the loss of other packets because each packet includes independent overheads and decoded independently

As a downside of the network, the number of slave nodes is limited to 8 with 3-bit IDs, and there must be a single master node. The network delay increases as the number of nodes because of a message rotation time and the overhead size also increases with the number of packets or data in a message. The algorithm in the master node is relatively more complex than in slave nodes. The master node has to retain all network information such as the number and the order of actuation and sensor data for each slave node.

The comparison of network properties is described in Table 1. More details about the timing will be discussed in the following section.

Protocol Analysis

Timing Analysis

Timing analysis of control networks such as Ethernet, ControlNet and DeviceNet was discussed [1]. Using the terminologies defined, we can characterize the timing properties of the protocol that are significant in a distributed control system.

Total network update time

As described in the previous section, the total NUT can be explicitly expressed by the equation:

$$NUT = \underbrace{(N_{node}-1)\times T_{dec} + N_{node}\times T_{tx}}_{MRT} + \underbrace{\left(\sum_{i=1}^{N_{slave}}\left(N_{act}(i)+N_{dat}(i)\right)\times N_{bpp} + N_{del}\right)\times T_{clk}}_{MDT} \quad (1)$$

where,

- N_{node} : number of nodes, $N_{slave}+N_{master}$
- N_{master} : number of master nodes, $N_{master}=1$
- N_{slave} : number of slave nodes
- $N_{act}(i)$: number of actuation data for slave(i) node
- $N_{dat}(i)$: number of sensor data for slave(i) node
- N_{bpp} : number of bytes per packet
- N_{del} : number of bytes of the delimiters (SOM and EOM) of a message
- T_{dec} : decoding time of a first packet
- T_{tx} : all propagation delay from a transmitter to a receiver
- T_{clk} : clock period

The NUT can be represented with the message rotation time (MRT) through the network and the returning message decoding time (MDT). T_{dec} is a fixed value once a packet is defined and T_{tx} is a constant since it is determined by the transmission speed of the transceiver and the propagation delay of a serial link, which are network properties. Hence, the MRT ultimately depends on the N_{slave} on the ring. As slave nodes are added or removed in the network, the MRT will increase or decrease, respectively. The MDT is the function of the sum of the number of actuation and sensor data of slave nodes, since N_{bpp} and N_{del} are determined by the protocol, and T_{clk} is a fixed network property.

In the example in Fig. 5, where N_{slave}=2, $N_{act}(i)$=1, $N_{dat}(i)$=3, N_{bpp}=3, N_{del}=2, $T_{dec}=5T_{clk}$, $T_{tx}=20T_{clk}$ and T_{clk}=50 ns, we can compute the NUT to be 4.8 μs for 16 bytes system data transfer over a single master and two slave nodes at 200 Mb/s transfer rate.

Blocking Time

The blocking time is defined as the duration that a message must wait once a node is ready to send it. It is a major factor in the determinism and the performance of a control network [1]. Like the token policy of the ControlNet, a slave node must wait to send a message in system data transfer until it receives its own actuation data in an incoming message from the logically previous node. It also needs to retransmit received packets before it begins to send sensor data. Therefore, the blocking time, T_{block}, can be expressed by T_{tx}, T_{dec} and the time to retransmit a received message at the current node. The T_{block} at the ith slave node is

$$T_{block}(i) = i\times(T_{tx}+T_{dec}) + \left(\begin{array}{l} N_{bpp}\times(\sum_{j=1}^{N_{slave}} N_{act}(j) + \\ N_{Tdat}(i)) + N_{SOM}\end{array}\right)\times T_{clk} \quad (2)$$

$$N_{Tdat}(i) = \begin{cases} 0 & ,i=1 \\ \sum_{j=1}^{i-1} N_{dat}(j) & ,i>1 \end{cases}$$

where the N_{SOM} is the number of bytes of a SOM code and the N_{Tdat}(i) is the number of sensor data that previous slave nodes have transmitted to the current ith slave node. The blocking time of the master node is zero since it can transmit a message without delay at the beginning of communication.

Frame Time

The frame time is the time needed to transmit a message frame. It can be expressed by the size of the data, the overhead, any padding, and the bit time [1]. In our protocol, the frame time T_{frame} at the ith slave node is represented by

$$T_{frame}(i) = \left(N_{bpp}\times(\sum_{j=1}^{N_{slave}} N_{act}(j) + \sum_{j=1}^{i} N_{dat}(j)) + N_{del}\right)\times T_{clk} \quad (3)$$

One of the properties of the ExoNet that differs from the other general protocols is that the size of the overhead increase as the amount of data increases since an overhead is added to each packet while those of the Ethernet and DeviceNet are fixed but relatively long. They are described in Table.1.

Propagation Time

The propagation time is the time needed for data to pass through a serial medium [1]. It is generally dependent on the distance between a source and a destination node and the physical properties of a serial medium. The propagation time according to link media is described earlier and referred to in [15].

The HOTLink has no intrinsic distance limit but the maximum distance is determined by the choice of interconnect media and the jitter while the data is in transit over the media.

The fiber-optic interconnection guarantees maximum distances with the lowest interference, and the selection of wire transmission lines depends on frequency-dependent attenuation as a function of the data rate and the media length [16]. However, a suitable attenuation compensation filter can lessen the detrimental effect of jitter and help to build the long-distance network.

Network efficiency

The network efficiency is defined as the ratio of the total transmission time to the time needed to send a message [1]. Since the network experiences no collision between messages and the releasing policy of a message is the same as the scheduled one as shown in Fig. 5, a node can start to transmit a message when it receives its actuation data, and the network efficiency of the NCS is 100%. This means all the time delay arises only from message transmission rather than message collision, and that performance of the network is good.

Network utilization

The network utilization, P_{util}, is the ratio of the time sending a message compared to the total running time [1]. However, in a ring-based structure, the definition must be changed. Here, we define it as the total time spent to send a message to the total running time of all nodes. Therefore, when all nodes run during the NUT, P_{util} can be expressed as:

$$P_{util} = \frac{\sum_{i=0}^{N_{slave}} T_{frame}(i)}{N_{node} \times NUT} \tag{4}$$

$T_{frame}(0)$ is the frame time for the master node and is equal to the equation (3) where $N_{dat}(j) = 0$. For the example in Fig. 5, it is 17.7%, which is better than those of Ethernet, ControlNet, but worse than one of DeviceNet in the scheduled releasing policy [1].

Network bandwidth

The network bandwidth (NBW) in the NCS can be defined as the ratio of the total bytes of the transferred system data to the total running time. It implies the number of bytes of system data transferred per unit time. Therefore, it can be represented by

$$NBW = \frac{N_{bpd} \times \sum_{i=1}^{N_{slave}} \left(N_{act}(i) + N_{dat}(i) \right)}{NUT} \tag{5}$$

where, N_{bpd} is the number of bytes per data; for instance, 2 bytes for 16-bit data. If it is applied to the example in Fig. 5, the NBW is 3.33 Mbytes/s.

Case Study: Exoskeleton

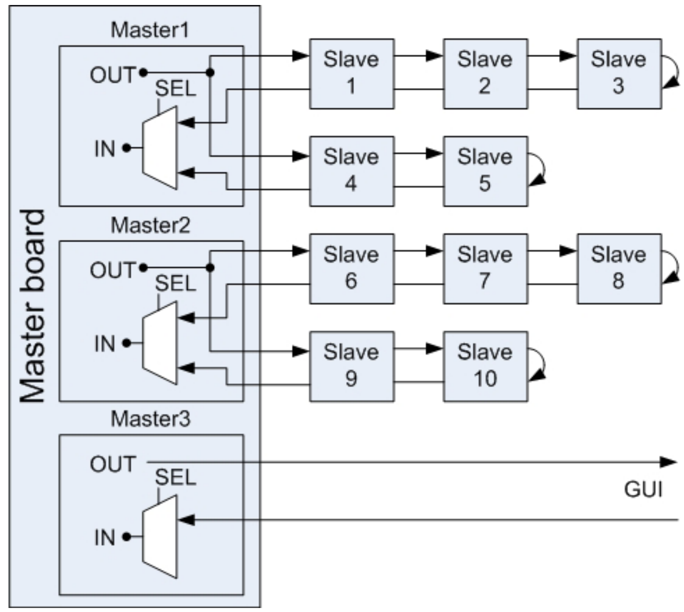

Fig. 6. Network of the Exoskeleton

The protocol has been applied to the NCS of the Exoskeleton robot, as shown in Fig. 6. The system is composed of a main controller that contains 3 transceivers as a master node and 10 slave nodes all over the robot, such as near joints or on the back. Among these 3 master nodes, two are dedicated to transferring system data, and the remaining one is used to communicate with the Graphical user interface (GUI) system for debugging purposes. Each of two master nodes includes two ring networks, one with 3 slave nodes and the other with 2 slave nodes, so each has 5 slave nodes on its two ring networks. We distributed nodes in order to reduce the volume of wiring from the main controller. Although the master can supervise two ring networks, it can communicate with only one network at a time. To switch the network, a master must switch the serial input to that network and wait for a receive section of a transceiver to reframe data stream. It generally takes a few clock cycles. Each slave node is assigned with one actuation data, $N_{act}(i)=1$, and six sensor data, $N_{dat}(i)=6$. T_{tx} is 20 clock periods and T_{dec} is 8 clock periods which increase from 5 clock periods in the previous example. It is prolonged to give a delay before reading data from the Receive FIFO in a transceiver to prevent its underflow. In addition, a time penalty to switch a network is 20 clock periods in our design. The NUT in the system where a master includes 5 slave nodes over two ring networks is 16.05μs to transfer 70 system data bytes, and its NBW is 4.36Mbytes/s. In an NCW with two master nodes, the total NBW is doubled, and is therefore 8.72Mbytes/s for

140 system data bytes. This network delay takes up only 3.21% of 500μs of the control loop period.

Conclusion

In this paper, we proposed a newly designed ring-based protocol for the HOTLink transceiver, CY7C924ADX, trademarked by Cypress. We discussed how the NCS is structured to establish a ring network between a single master node and slave nodes. Its architecture is developed to reduce volume of wiring and easy installation. The packet format that contains one piece of data information and the message frame format consisting of packets are generally defined in the protocol design for the ring topology. We introduced the communication procedures and presented message transfer using a message frame during system data transfer. It is an efficient system for distribution of actuation data and collection of sensor data at fast speed over the NCS. It reduces time delay caused by serial communication and guarantees determinism, collision-free and fairness in the network. Several characteristics of the network were presented and the comparison with the other common protocols was also described. Then we discussed its application on the Exoskeleton robot. The ExoNet protocol is found to be competitive, effective, and capable of superior performance in the distributed control system with a single master node and several slave nodes regardless of the amount of data that each node contains.

Our ongoing developments include making use of the recently released high speed transceiver, Cypress CYP15G series, whose maximum transfer rate is 1.5Gbps. We are also redesigning the protocol in order to remove redundancy of the overhead by allocating the whole fixed overhead instead of separate overhead per packet, and we also plan to increase the maximum number of nodes to 256 with 8-bit ID field.

REFERENCES

[1] F.-L. Lian, J.R. Moyne, and D.M. Tibury, “Performance evaluation of control networks: Ethernet, ControlNet, and DeviceNet”, IEEE Control Systems Magazine, vol. 21, no. 1, pp. 66-83, Feb. 2001.

[2] S. Biegacki and D. VanGompel, “The application of DeviceNet in process control”, ISA Trans, vol. 35, no.2 pp. 169-176, 1996.

[3] L.D. Gibson, “Autonomous control with peer-to-peer I/O networks”, Sensors, vol. 12, no. 9. pp. 83-90, Sept. 1995.

[4] A. Ray, “Introduction to networking for integrated control systems”, IEEE Contr. Syst. Mag., vol.9, pp. 76-79, Jan. 1989.

[5] G. Schickhuber and O. McCarthy, “Distributed fieldbus and control network systems”, Computing Contr. Eng., vol. 8, no. 1, pp. 21-32, Feb. 1997.

[6] D. Song, T. Divoux, F. Lepage, “Design of the distributed architecture of a machine-tool using FIP fieldbus”, in Proc. IEEE Int. Conf. Application Specific Systems, Architectures, Processors, Los Alamitos, CA, 1996, pp. 250-260.

[7] G. Cena, C. Demartini, and A. Valenzano, “On the performances of two popular fieldbuses,” in Proc. IEEE Int.Workshop Factory Commun. Syst., Barcelona, Spain, Oct. 1997, pp. 177-186.

[8] J.D. Wheelis, “Process control communications: Token Bus, CSMA/CD, or Token Ring?” ISA Trans., vol. 32, no. 2, pp. 193-198, July 1993.

[9] S. Saad-Bouzefrane and F. Cottet, “A performance analysis of distributed hard real-time applications,” in Proc. IEEE Int. Workshop Factory Commun.Syst., Barcelona, Spain, Oct. 1997, pp. 167-176.

[10] F.-L. Lian, J. Moyne, and D. Tilbury. Network design consideration for distributed control systems. IEEE Transactions on Control Systems Technology, March, 2002.

[11] F.-L. Lian, J.R. Moyne, and D.M. Tilbury. Time delay modeling and sample time selection for networked control systems. In Proceedings of the ASME Dynamic Systems and Control Division, Nov. 2001

[12] P. Otanez, J. Moyne, and D. Tilbury. Using deadbands to reduce communication in networked control systems. Proceedgins of the 2002 American Control Conference, May, 2002.

[13] Gang, Y. Cai, X. He, and W. Zhang. A new realtime Ethernet MAC protocol for time-critical applications. Journal of Central South University of Technology (English Edition), March, 2002.

[14] CYPRESS CY7C924ADX Datasheet (http://www.cypress.com/cfuploads/img/products/CY7C924ADX.pdf)

[15] HOTLink Transmitter/Receiver (http://www.cypress.com/cfuploads/img/products/38-02017.pdf)

[16] HOTLink Copper Interconnect-Maximum Length vs. Frequency (http://www.cypress.com/cfuploads/support/app_notes/copdist.pdf)

