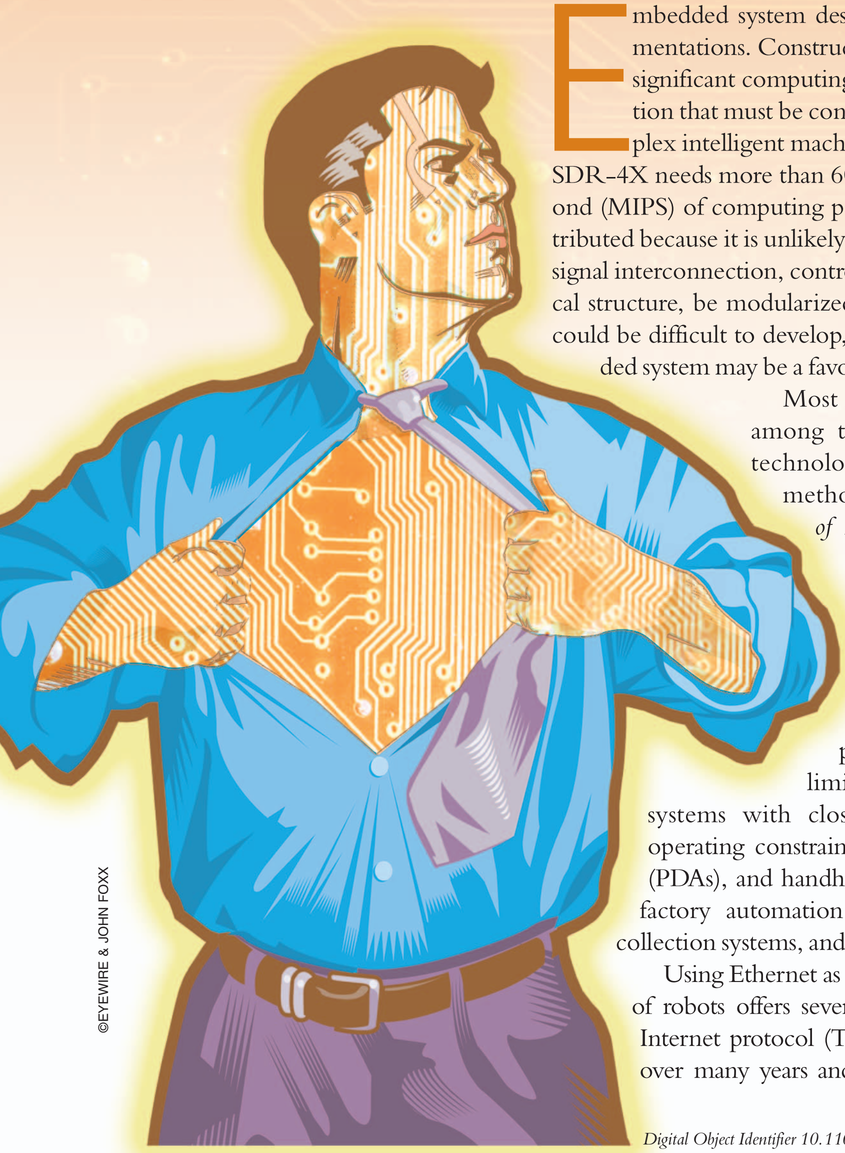
Embedded System Design for Robots

Design Concept, System Architecture, and Implementation

BY LI-WEI WU AND
JWU-SHENG HU



©EYEWIRE & JOHN FOXX

Embedded system design is essential for successful intelligent robotic implementations. Constructing a robot that can perform complicated tasks requires significant computing power and system integration effort [1]–[3]. The question that must be considered is, "What embedded system is needed for a complex intelligent machine such as a humanoid robot?" for example, the SONY SDR-4X needs more than 60 processors and over 2,260 million instructions per second (MIPS) of computing power [3]. The computing architecture is inherently distributed because it is unlikely to dump all the raw information into a single CPU. The signal interconnection, control, and information processing should, like the mechanical structure, be modularized. Without careful design, the entire embedded system could be difficult to develop, maintain, and extend. Therefore, a distributed embedded system may be a favorable choice.

Most distributed systems have a certain network topology among their processing units. Based on currently available technology, what would be the appropriate networking method for an intelligent machine? In May 1998, the *Journal of Internet Computing* ran a special issue on embedded Ethernet technologies and highlighted recent developments and industrial applications of embedded Internet technology [4]. This technology supports devices and operating environments outside the traditional desktop PC envelope, where onboard memory, CPU power and speed, display capability, persistent storage, and costs are usually severely limited. At the soft end of the spectrum are embedded systems with close-to-desktop-PC resources and no real-time (RT) operating constraints, including cellular phones, personal digital assistants (PDAs), and handheld terminals. At the hard or deeply embedded end are factory automation and machine controllers, instrumentation and data collection systems, and telecommunication equipment [5], [6].

Using Ethernet as the communications backbone for the embedded systems of robots offers several advantages. First, the transmission control protocol/Internet protocol (TCP/IP) de facto standard has been proven to be robust over many years and is open to the technical community. Second, when a

Digital Object Identifier 10.1109/MRA.2008.917885

1070-9932/08/$25.00©2008 IEEE

robot is interconnected with the access network (the public network) and the home network, the designer must heavily emphasize the addressing system and related security issues. Historically, the IP community has carefully considered these problems, and therefore, IP technology is competent for addressing these issues. The abundance of applications and tools make Ethernet highly effective for unifying the system interface and reducing system complexity. For example, it can be used as the communication backbone in parallel processing [7]–[9]. The latest technology has pushed Ethernet bandwidth to the Gb/s/Tb/s range [10], leading to proposals to incorporate it into the system-on-a-chip (SoC) design in both the academic and industrial sectors [11]–[13]. In terms of bandwidth and cost, Ethernet offers a very competitive solution for embedded applications.

However, Ethernet devices require synchronization for purposes such as motor control and vision. Conventional protocols, e.g., usere datagram protocol (UDP) and TCP/IP, are inadequate for synchronization because the 1-persistent carrier sense multiple access with collision detection (CSMA/CD) protocol has unpredictable delay characteristics. CSMA/CD employs an exponential back-off mechanism to circumvent collisions, making the network nondeterministic and thus non-RT. When both RT and non-RT packets are transported over an ordinary Ethernet, RT packets may experience a large delay because of 1) the contention with non-RT packets in the originating node and 2) the collision with RT and non-RT packets from other nodes.

Numerous studies have been undertaken to support RT bandwidth guarantees over Ethernet hardware such as 100VG AnyLAN [14], RETHER [15], EtheReal [17], [18], RTnet [19], RtP [20], and TTP/C [21]. However, most of them have a concentrated management structure, requiring an effective and programmable switch router or hub to run the RT schedule algorithm. Such a router is usually very large and superfluous for small robot systems.

In 2003, the International Electrotechnical Commission in Geneva developed the standardization process for RT Ethernet (RTE) to use Ethernet as a fieldbus alternative, encouraging the further establishment of the standards while maintaining the highest possible levels of flexibility [22]. Currently, industrial Ethernet is also vigorously becoming established in automation technology, and several promising approaches for RTE are forcing their way into industry, such as PROFINET, Ethernet for control automation technology (EtherCAT), and Powerlink. The various RTE approaches differ significantly and are not compatible with one another. Therefore, they are associated with some defects in embedded systems of small mobile robots. Each issue is explained in the following sections.

PROFINET

PROFINET is the industrial Ethernet solution supported by PROFIBUS International User Group. PROFINET's background is distributed automation: objects can be easily described, reused, and connected to one another. It supports two protocols: standard distributed component object model (DCOM) over TCP/IP for non-RT communication and RT class I protocol for medium-performance real time. The well-known DCOM [23] can be seamlessly integrated with process automation based on object linking and embedding for process control. Notably, however, DCOM is a high-overhead protocol and thus unsuitable for embedded, low-cost systems or small robot systems [24]. For further information of PROFINET, please see [22], [24]–[28].

EtherCAT

EtherCAT is an open RT solution that was developed by Beckhoff and is supported by the EtherCAT Technology Group. It is tailored for centralized automation only. It does not support distributed control systems. EtherCAT can be regarded as a new fieldbus with Ethernet cables. Notably, EtherCAT is limited to the use of Beckhoff proprietary application-specific integrated circuits (ASICs) and so cannot be widely applied in any other embedded Ethernet development solution (standard Ethernet chips) [29]. This issue will limit the development of robot systems. Please see [22], [30], and [31] for more detail information.

Ethernet Powerlink

Ethernet Powerlink [32]–[35] was defined by Bernecker & Rainer, and it is now supported by the Ethernet Powerlink Standardization Group (www.ethernetpowerlink.org). Ethernet Powerlink is mainly based on a principle called slot communication network management (SCNM). It uses individual slots for isochronous data and shared slots for asynchronous data. The SCNM method guarantees collision-free data transfer between master and slave. All nodes within a segment are synchronized by the reception of the start-of-cycle Ethernet packet, which the master sends at the beginning of each cycle. The master polls each slave using a poll-request frame. A slave is only allowed to send an Ethernet packet after it has received such a frame. However, SCNM method does not make the best of current network switch performance. Since a current switch can functionally be considered to be a multiport bridge (see "Network Switch"), in practice, a switch is much more powerful than a traditional bridge primarily because of its ASIC-based hardware architecture and its ultrarapid simultaneous multiple access memory. Additionally, a switch can have an IP address and as many media access control (MAC) addresses as ports, facilitating its configuration. Restated, an Ethernet switch can provide one transmitting collision domain per port (dedicated bandwidth segment), allowing the collisions to be completely eliminated if the port is in full duplex [36] and only one station is connected to it [37], [38]. In fact, in such a configuration, the CSMA/CD protocol is kept only to ensure compatibility with the classic shared Ethernet since no transmitting collision is possible. However, the SCNM only allows an Ethernet packet to be sent after the reception of the poll-request frame, so Powerlink has not fully utilized the characteristics of the current network switch. By using the current switch technique, the improvement in the RTE protocol can allow different collision domains to send Ethernet packets in each individual slot.

To effectively address the issues raised earlier, this study implements a hardware RT protocol (HRTP) for an embedded Ethernet robot system. An HRTP-task time wave structure based on the packet traffic control approach is proposed to enable management to reduce each receive collision domain and form microsegments that are separated by network switch. Furthermore, the HRTP is a distributed RT protocol with a small footprint and is very suitable for the proposed application (especially for low-end standard embedded Ethernet solutions). Embedded networking was achieved with a distributed embedded system using Ethernet to validate HRTP. The system comprises microcontrollers for motoring and sensing and a host platform that uses SoC. The microcontrollers interface with the network interface controller (NIC) chips where both HRTP and a lean TCP/IP protocol stack are implemented. An RT UDP-to-serial packet converter was designed and implemented to handle HRTP Ethernet packets. The host platform runs embedded Linux and has two network interfaces, a 10/100 Mb/s LAN and a wireless LAN. The network protocol stack of the embedded system was modified to accommodate HRTP. A multiaxis robot platform was also constructed to demonstrate the capability of the embedded system. The platform has 16 degrees of freedom (DoF) (16 motors) and the number of sensors can be expanded. The robot takes the form of a quadruped walking machine. Each foot has four motors and is capable of a complex motion profile. HRTP's transparency enables a developer to manipulate the robot easily using socket programming on a desktop computer or mobile device (such as PDA). The proposed architecture provides a system design tutorial for highly intelligent and extremely complex robot systems.

The rest of this article is organized as follows. The "HRTP" section briefly describes the HRTP model and its simple remote clock synchronization methodology. This section also outlines HRTP's control packet structure. The "Implementation" section explains the application of distributed architecture in a robot electronic platform, and this platform is applied to a complex quadruped to demonstrate its effectiveness in the conclusion. Network switch, OSI model, and TDMA-E are discussed later.

Network Switch

Switches occupy the same place in the network as hubs. Unlike hubs, switches examine each packet and process it accordingly rather than simply repeating the signal to all ports. Switches map the Ethernet addresses of the nodes residing on each network segment and then allow only the necessary traffic to pass through the switch. When a packet is received by the switch, the switch examines the destination and source hardware addresses and compares them to a table of network segments and addresses. If the segments are the same, then the packet is dropped (filtered); if the segments are different, then the packet is forwarded to the proper segment. Additionally, switches prevent bad or misaligned packets from spreading by not forwarding them.

The filtering of packets and the regeneration of forwarded packets enable-switching technology to split a network into separate collision domains. The regeneration of packets allows for greater distances and more nodes to be used in the overall network design and dramatically lowers the overall collision rates. In switched networks, each segment is an independent collision domain. In shared networks, all nodes reside in one, large shared collision domain.

Most switches are easy to install and self-learning. They determine the Ethernet addresses in use on each segment, building a table as packets are passed through the switch. This plug and play element makes switches an attractive alternative to hubs [37], [38].

HRTP

This section elucidates the basic concepts and fundamental elements of HRTP. First, the control network topology for robots is proposed and related issues considered. The new open system interconnection (OSI) model, incorporating packet traffic concept in the HRTP architecture, is described, and several state machines are defined to explain how each mechanism controls the state in HRTP. Furthermore, the clock synchronization issue is addressed. Eventually, the simple HRTP control packet is presented. All conceptual elements are considered later.

Proposed Control Network Topology

Figure 1 schematically depicts the proposed model. The robot system consists of three layers: the Internet layer, the gateway layer, and the control and sensing layer. Clearly, this structure mimics the topology of today's network infrastructure, in which the control and sensing layer is the so-called Intranet. However, the difference is that RT messaging must be enforced for RT control (as in motor synchronization). The proposed HRTP is implemented to provide RT messaging. Significantly, by maintaining TCP/IP compatibility, the robot system can easily expand within this layer (such as by adding another processor or sensor). Additionally, robot internal communications between the Internet and the control and sensing layers become transparent (such as a simple bridge function in the gateway layer). Many communications technologies can also be implemented over TCP/IP.

However, the CSMA/CD of Ethernet employs an exponential back-off mechanism to prevent collisions that make the network nondeterministic and thus non-RT Specifically, the Ethernet packets are gathered into a gate, and collisions occur in the receiver node (Figure 2). If an embedded Ethernet receiver cannot handle these packets, then the receiver node also breaks and affect the stability of the robot system. Furthermore, the proposed HRTP is a distributed RT protocol, which is

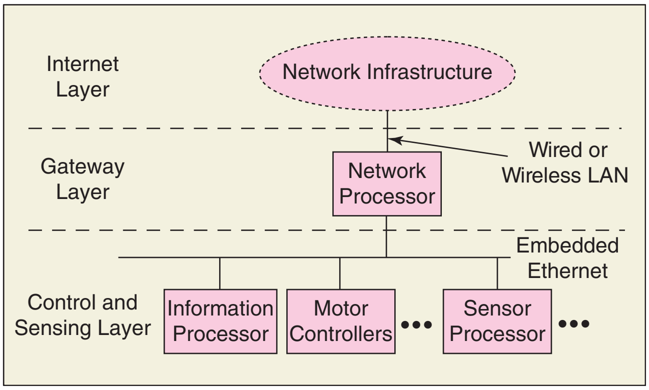


Figure 1. Robot architecture platform model.

targeted for embedded RTE environments. The HRTP model concept is explained later.

New OSI Model for HRTP

Figure 3 presents a model of the HRTP system in a robot internal Ethernet network. The new OSI model (see "OSI Model") and the conventional model differ mainly in that the proposed model inserts toggle traffic switch, called the HRTP-transport switch, between the transport and data link layers. This alteration does not affect original TCP/IP because HRTP integrates into the network's interface driver. The HRTP network driver has the following advantages:

- high-speed switching
- adaptation to original TCP/IP protocol stacks
- efficient utilization of operating system resources
- increased driver por-tability.

The traffic switch (HRTP-transport switch) intercepts all sending out packets and divides time slices according to the HRTP schedule and the common base clock. Figure 4 describes HRTP's time mechanism. The following list presents the states of the HRTP state machine.

- *Initialization:* Prepares the intranet system (local embedded Ethernet nodes in robot system).
- *Remote clock synchronization:* Synchronizes the clock to ensure accurate time events, as explained later.
- *Explore intranet:* Exports the robot's locally embedded Ethernet network state to determine the client's IP address and the devices.
- *Post HRTP schedule:* Sends out HRTP schedule to all devices.
- *Start clock:* Starts running common base clock.
- *Traffic control:* Begins HRTP traffic control.
- *Idle:* System enters sleep mode.

The state machine is controlled by HRTP command packets, as presented in the "Content of HRTP Control Packet" section. Collision occurs in received nodes according to the switch property described in the introductory paragraph. Notably, HRTP applies the distributed conception to lower the workload of the conventional centralized RT management structure and simplify the RT

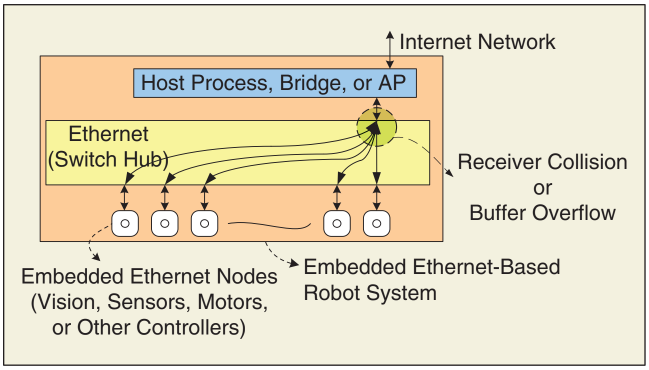


Figure 2. *Receiver collision model.*

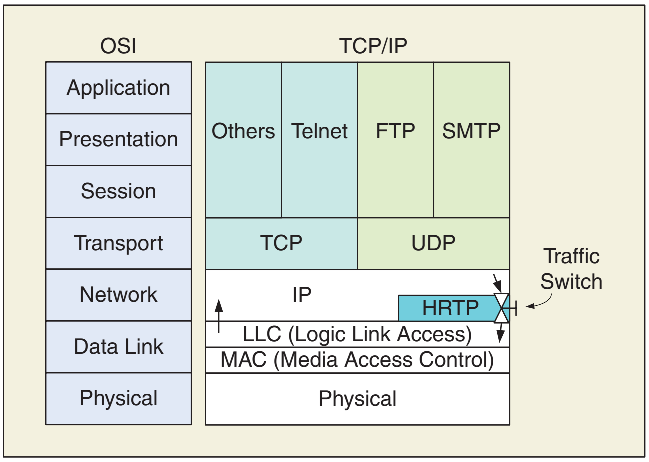


Figure 3. *A new OSI model describing the service of HRTP system.*

OSI Model

The OSI model [37] is a reference model developed by the International Organization for Standardization in 1984 as a conceptual framework of standards for communication in a network that links various equipment and applications from different vendors. It is now considered to be the primary architectural model for intercomputing and internetworking communications. The structures of most of the network communication protocols used today are based on the OSI model. The OSI model defines the communication process as involving seven layers and divides the tasks involved with moving information between networked computers into seven smaller, more manageable task groups. A task or group of tasks is then assigned to each of the seven OSI layers. Each layer is reasonably self-contained so that the tasks assigned to each layer can be implemented independently. This enables the solutions offered by one layer to be updated without adversely affecting the other layers.

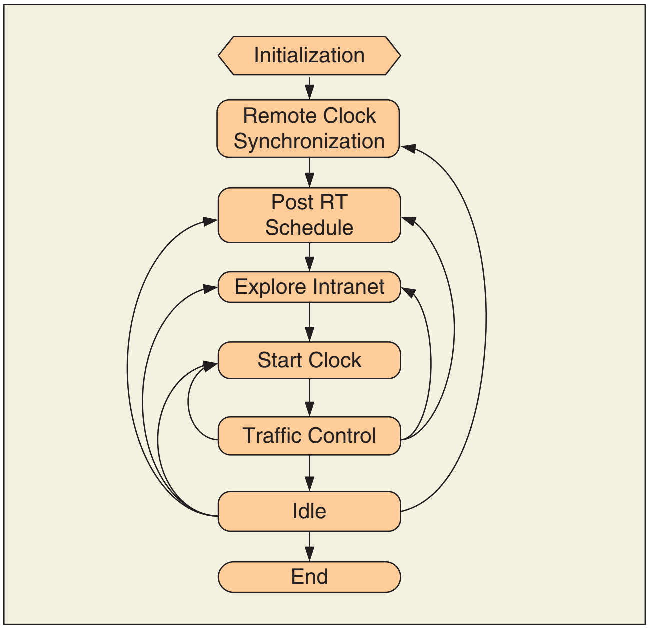


Figure 4. *State machine of HRTP.*

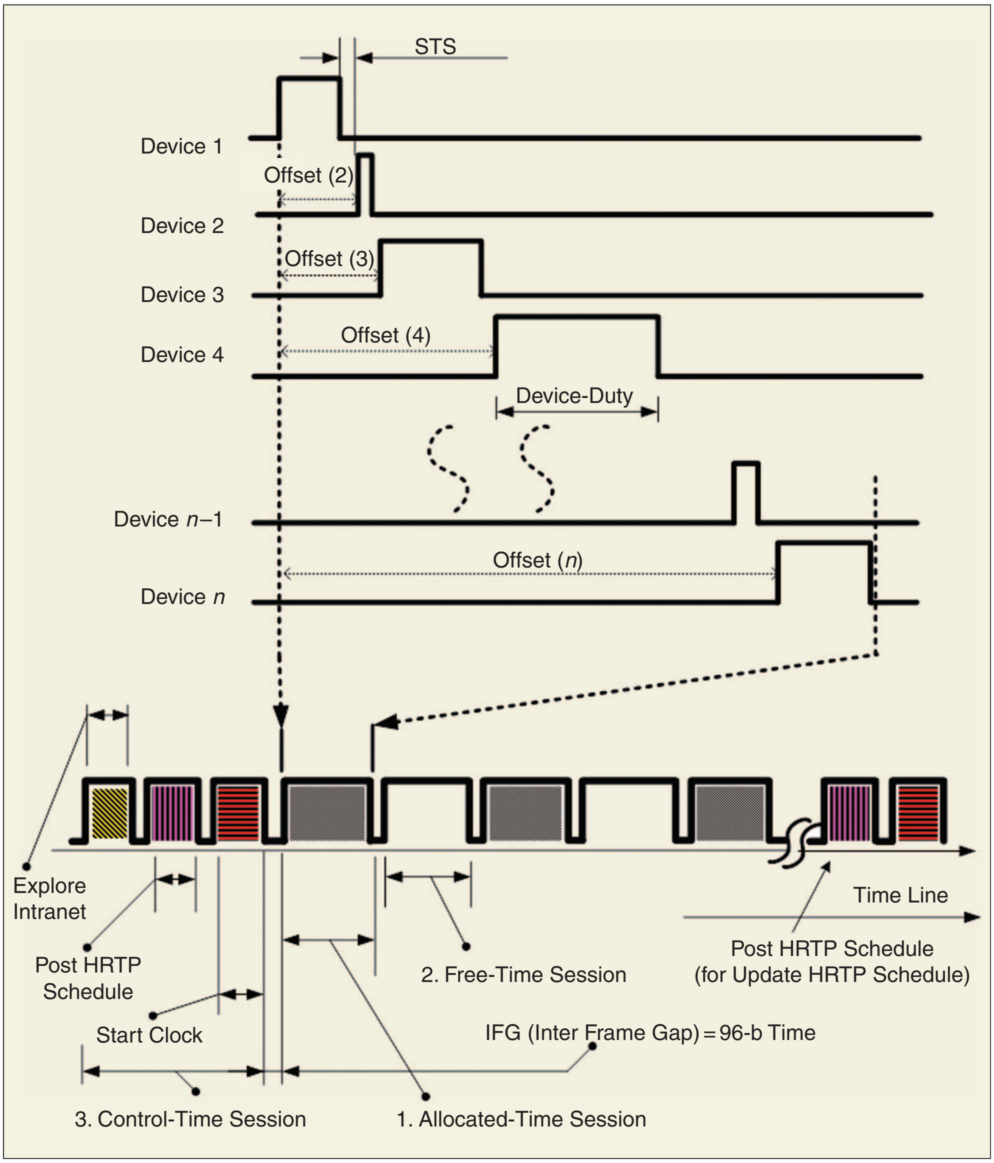


Figure 5. *HRTP time wave structure based on packet traffic control technique approach.*

schedule. Hence, the right of management is dispatched to each received node. Further, the rights of management contain governing the events trigger of overall state machine, generating the packet content of RT schedule and posting the packet of RT schedule.

TDMA-E

TDMA-E is Ethernet transmission technology that allows a number of RTE sending nodes to access the respective Ethernet receive nodes without interference by allocating unique time slots to each RT node within each collision domain. Since the current Ethernet switch can provide one transmitting collision domain per port, allowing complete elimination of the collisions if the port is in full duplex (IEEE 802.3x) [36], and is connected to only one station, the TDMA-E transmission scheme multiplexes packets over different collision domains in a network switch (see ''Network Switch''). Figure 23 shows an example of TDMA-E in one network switch.

HRTP Time Wave Structure

The HRTP-task time wave structure is presented to describe further the time control mechanism of HRTP. The HRTP time wave structure is based on packet traffic control technique. A timeline is divided into three sessions using time-triggered architecture (Figure 5). These are 1) allocated (fixed)-time session, 2) free-time session, and 3) control-time session. Each session is explained later.

Allocated (Fixed)-Time Session

The allocated-time session mainly allocates to each sending node a time slot in which to transmit RT data (device 1 ~ device n, as illustrated in Figure 5). Consecutive sessions are separated by safety-time sections (STS), defined as

$$\mathrm{STS} = T_{\mathrm{IFG}} + T_{\mathrm{BTMT}}, \quad (1)$$

where T_{IFG} represents the interframe gap (IFG) and T_{BTMT} denotes the NIC's buffer transfer maxima time (NIC_BTMT). The IFG is 96 b in length [36]. The NIC_BTMT is defined by the MAC ring buffer. NIC_BTMT is related to the MAC device and the supporting driver and reserves a short span of time for transmitting packets under the MAC ring buffer. Therefore, the STS can circumvent the ring buffer overrun problem. The wake-up of the device duty is managed by the variable offset (n), as indicated in Figure 5.

Furthermore, the allocated-time session is mainly based on the time division multiple access Ethernet (TDMA-E) model concept (see "TDMA-E") to simplify and govern the distributed embedded Ethernet packet traffic.

Free-Time Session

The free-time session operates as in the standard IEEE 802.3 [36] and allows every device to send packets unrestricted by the CSMA/CD protocol. The principal service transmits non-RT packets. In the free-time session, all devices compete for time slots to transmit using the conventional CSMA/CD protocol. Its objective is to preserve the original CSMA/CD flexibility on HRTP.

Control-Time Session

The control-time session is defined to detect and control the device nodes in intranet, the command of post HRTP schedule and start clock (Figures 4 and 5), and occurs at the beginning of each Ethernet traffic period. This controlling and detecting of behavior is regarded as explorer intranet (Figure 4). The transmission is governed by the HRTP time schedule and driven by time events. The traffic control model is elucidated next.

HRTP Traffic Control Model

The previous sections described the concepts that underlie HRTP. This section introduces the operation of HRTP in real hardware. The hardware implementation of HRTP is associated with a traffic control model and is called the hardware RT traffic control model (HRTP-TCM). Figure 6 illustrates the HRTP-TCM, which consists of six main modules.

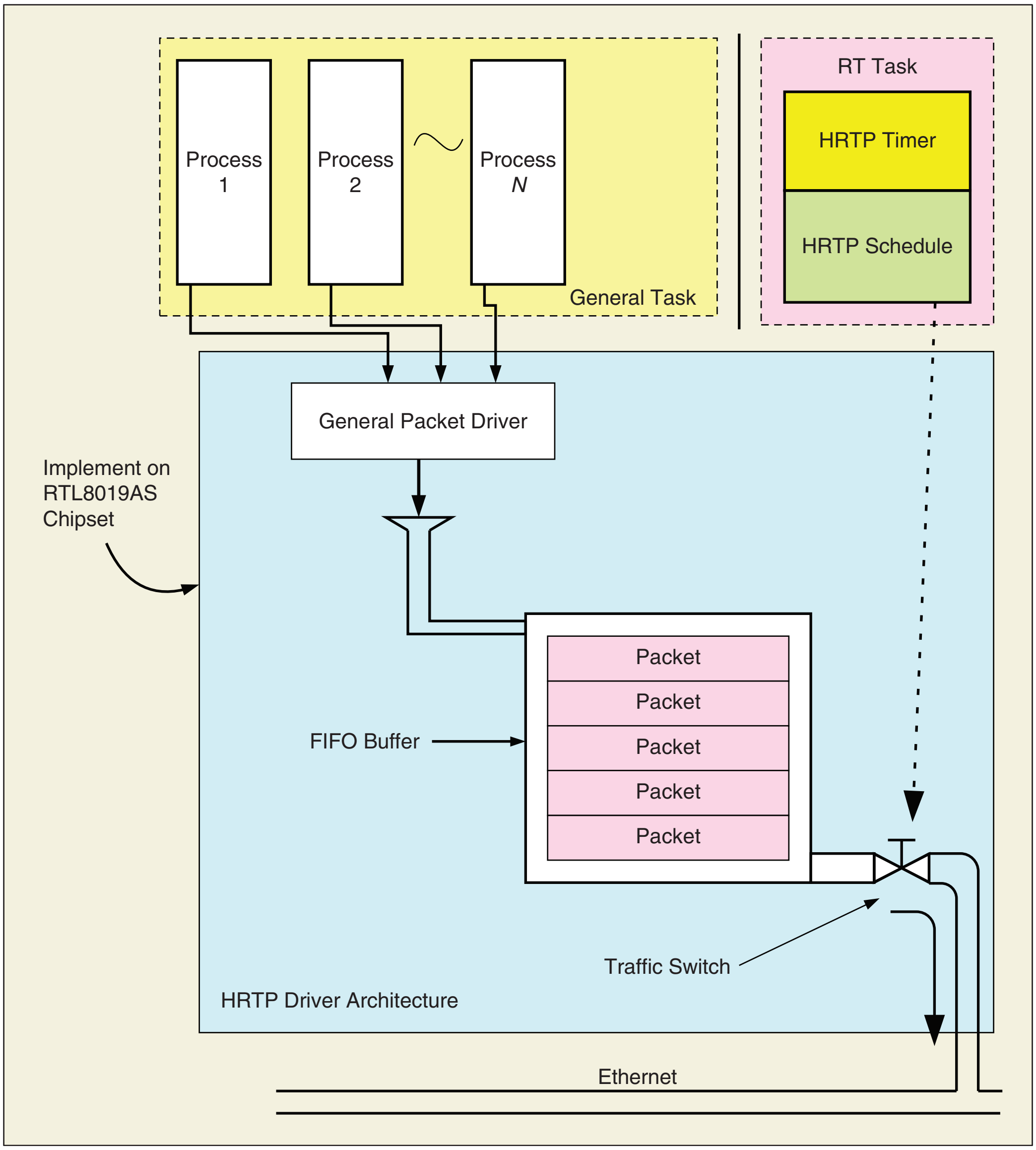


Figure 6. RT traffic control model of HRTP.

- *HRTP timer:* It is the RT timer in each embedded Ethernet device.
- *HRTP schedule:* It stores and governs the schedule of HRTP.
- *Traffic switch:* It controls the traffic flow. According to HRTP timer and HRTP schedule, traffic switch controls the passing through and blocking of Ethernet packets.
- *FIFO buffer:* FIFO stands for first in/first out and refers to the way in which the NIC processes data. A FIFO buffer is a memory device that allows flow from the CPU to the network controller and vice versa to be controlled and ensures packet integrity.
- *General-packet driver:* It includes TCP/UDP/IP packet driver.
- *HRTP-communications interface:* It receives HRTP command packets during control-time session.

However, the HRTP protocol can only be optimized by synchronizing all network devices. When the clocks are not synchronized, HRTP merely provides a new, reasonable, and effective scheme to plan the use of intranet transmission resources. Figure 7 describes packet transmission with reference to the RT traffic control model when the HRTP schedule is changed.

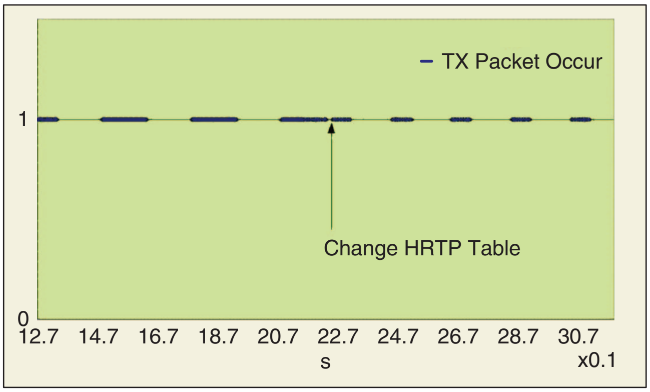


Figure 7. Packets transmitted using HRTP-TCM and instantaneous changes to the HRTP schedule.

Clock Synchronization

TCP/IP and CSMA/CD are the two factors that most strongly influence timing control. Timing precision depends on the application. In an HRTP system, the RT traffic controller modes are designed to be distributed. The remote clocks are not always synchronized. Clock synchronization is extremely important in ensuring that a traffic event is performed correctly. Minimizing the communications delay is generally a critical design consideration. However, in control applications, the deterministic response time requirement is frequently more important than the minimization of delay. The network time protocol (NTP) is used to synchronize the time of a computer client or server with another server or reference time source, such as a radio or satellite receiver or modem. Typical NTP configurations utilize multiple redundant servers and diverse network paths to obtain high accuracy and reliability [39]–[41]. However, NTP alone is not intended for high-reliability RT applications because it does ensure the precision of the global time base [20]. For the proposed application, a differential clock methodology, similar to the IEEE 1588 Standard [42], was adopted. The internal clock is synchronized by adjusting the local clock of every participating node to the local clock of a specific master node. This method is called the master-worker scheme [20]. This protocol is quite simple and easy to implement. Figure 8 presents this synchronization mechanism within HRTP and synchronization steps. On the

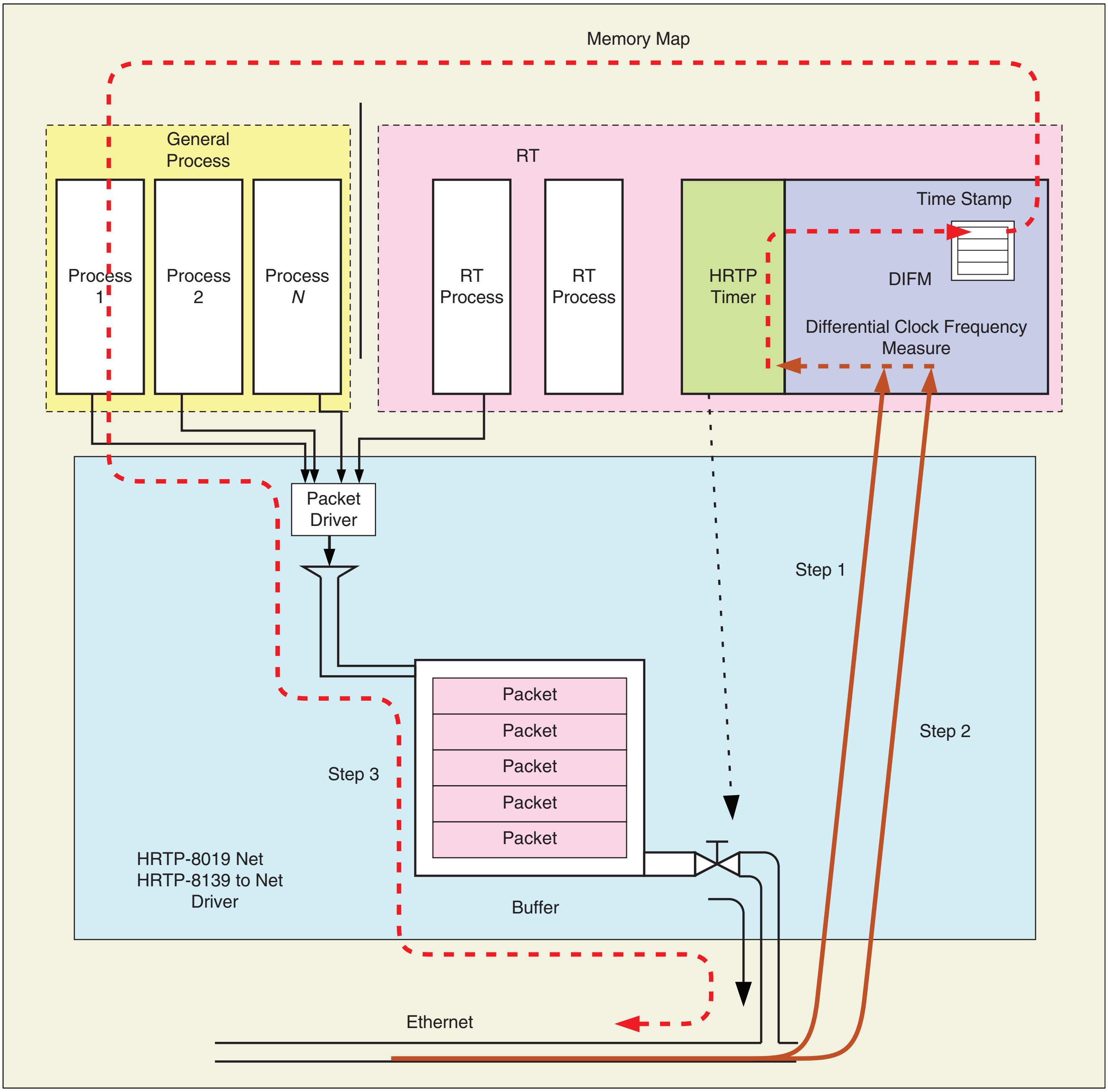


Figure 8. *Clock synchronization model in HRTP. (The differential time is calculated in Step 1 and Step 2, and the result corrects the HRTP timer. Step 3 reports the clock ticks to the network.)*

platform designed in this work, the resulting clock synchronization error is approximately ±0.5 ms (see the "Implementation" section).

Content of HRTP Control Packet

Generally, HRTP defines every receive node as a host node to manage its client nodes. Thus, HRTP allows the embedded Ethernet robot system to consist of several receive nodes in a network switch. Accordingly, HRTP also provides three packets to manage each subsystem and the timing control.

Request Report IP Data Command

Figure 9 proposes the request report IP data command packet, which is the first command packet that initializes the HRTP-embedded Ethernet systems. The hosts (defined as the receive nodes in the network switch) initiate the transmission of request report IP data command packets. This packet requests the client's IP address and numbers of the devices from a management host LAN device and sends this information to every client. Every client device that receives such a packet replies with the same packet to the management host LAN device. The host management LAN device records these IP addresses to access the LAN status.

HRTP Posting Schedule Command

The main function of the HRTP schedule command packet is to describe event-driven traffic switches. This packet fully describes the sliced segment of the HRTP-task time wave structure and has a variable packet length (Figure 10). Free-time and allocated-time sessions are both 32 b long. The duty cycle of the device is defined by

$$\text{Device duty} = \frac{T_{\text{RT}}}{T_{\text{ALT}} + T_{\text{FT}}}, \tag{2}$$

where T_{RT} denotes the length of the allocated-time session for each node and T_{ALT} and T_{FT} are the lengths of the entire allocated-time and free-time sessions, respectively. In the HRTP posting schedule command content, Device (n) Duty is defined as (2). Additionally, Figure 5 defines offset (n) as the offset time value of device n, whose traffic switch is turned on during an allocated-time session.

As stated in the introductory paragraphs, a current network switch can provide a separate transmitting collision domain for each port. Thus, the HRTP exploits this characteristic of network switch and TDMA-E to govern the RT schedule (see "TDMA-E"). In the process, each received node generates the schedule and manages the Ethernet traffic of all sending nodes. However, a network switch generally has some receive nodes and some sending nodes. And the low-end embedded Ethernet solutions usually have limits in handling the Ethernet packets and information processing. Hence, the Ethernet packet schedule should satisfy the following two constraints:

$$\text{DR} \le \text{MGS}(i) \cdot \text{TSL}, \tag{3}$$

where $i = 1, 2, \ldots, m$, and

$$\text{MHR} \ge \sum_{i-1}^{n} \text{MGS}(i), \tag{4}$$

where the parameters of (3) and (4) are defined as follows.

1) *DR:* Overall information processing Demand Rates of each receive node in its respective collision domain (in B/s)
2) *TSL:* HRTP Time Slot Length (traffic on) of each node in one second
3) *MGS(i):* Maximum Generating rates of information of the Sending node (i) (in B/s)
4) *MHR:* Maximum information Handling Rates of receive node.

Equation (3) claims that the sending node's (i) generation of information satisfies process demands. Equation (4) claims that the packet flow of receive node (n) may be smooth and capable of being processed. Each receive nodes in a switch will generate the respective HRTP schedule command packet and send it to its client nodes. Accordingly, the HRTP mechanism can schedule and guarantee that the entire packet transmission process of embedded Ethernet devices in each switch collision domain is in real time.

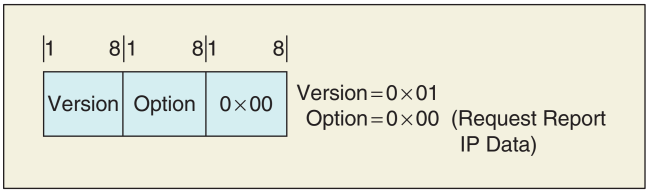


Figure 9. *Content of request report IP data.*

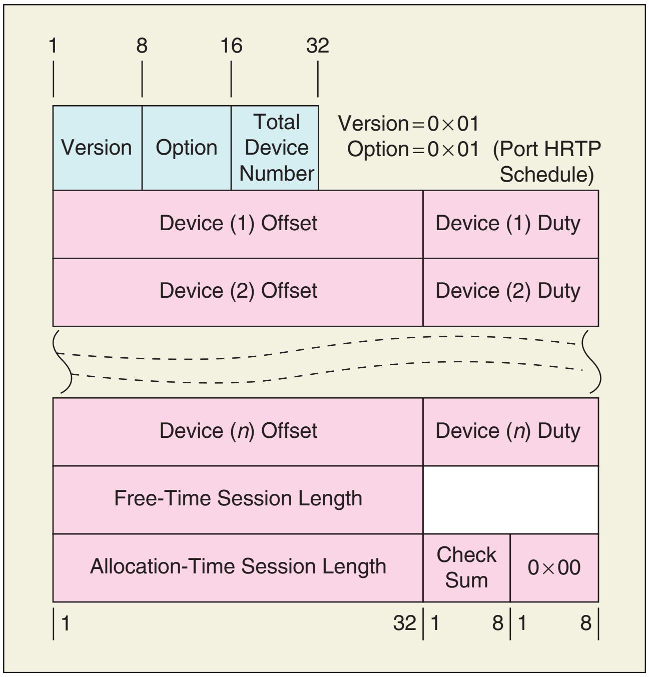


Figure 10. *Content of HRTP posting schedule command.*

Start Clock Command

The primary function of this packet is to fire up all of the HRTP client timers. Figure 11 shows the content of start clock command.

In summary, the HRTP utilizes the self-splitting separate collision domains property of network switch technology and elaborately controls the traffic of packets of every embedded Ethernet device to ensure RT operation. The distributed tiny Ethernet RT protocol architecture for low-end embedded Ethernet devices and small robot systems is presented earlier. The following section will present the implementation in small robot systems.

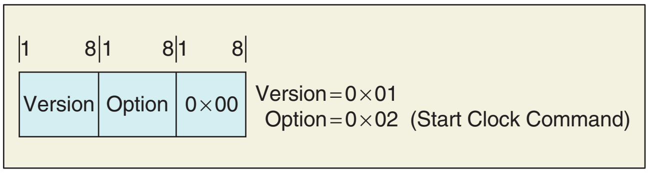


Figure 11. *Content of counter start command.*

Implementation

Overview

The proposed RT protocol is implemented on a small quadruped machine to enable transparent machine interconnections with the embedded RTE network environmental infrastructure (HRTP).

Nature provides very good design samples. The cerebrum, cerebellum, and nerves are metaphors for the computing power and database of computers on the Internet (Figure 12). Since the proposed novel distributed platform provides a wireless LAN interface, this platform concept is easy to integrate with other Internet devices. Figure 13 displays a simple schematic diagram of the robot platform. In Figure 13, A is the host processor, running embedded RT Linux and serving as the central processing unit and a gateway or bridge to the outer world. B represents a hub for network expansion. Both C and D are microcontrollers, which handle motor control and sensing. E represents a robot in which the systems (A–D) are installed.

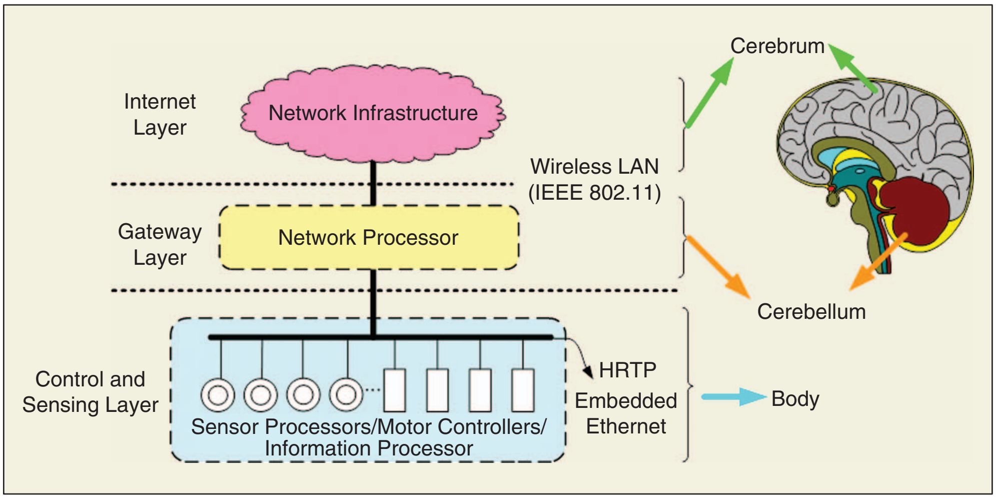


Figure 12. *Novel distributed embedded platform using Ethernet as the communications backbone and based on a three-layer concept: 1) control and sensing layer, 2) gateway layer, and 3) Internet layer. This proposal regards the cerebrum, cerebellum, and nerve as abstract concepts. An RT protocol named HRTP is proposed in the control and sensing layer.*

Electronic

Figure 14 presents the design diagram of an embedded HRTP Ethernet controller and multimotor controller module microcontrollers (parts C and D in Figure 13). The embedded HRTP Ethernet controller, which consists of an 8-b micro control unit (MCU) (PIC-18F452), is based mainly on the sensor network protocol (I_2C bus) and a serial motion control network. Each multi-microcontroller comprises an 8-b MCU (PIC-18F452), which controls eight motors and is integrated with numerous analog and digital interfaces, such as the pulse width modulation interface and the I_2C EEPROM (Microchip

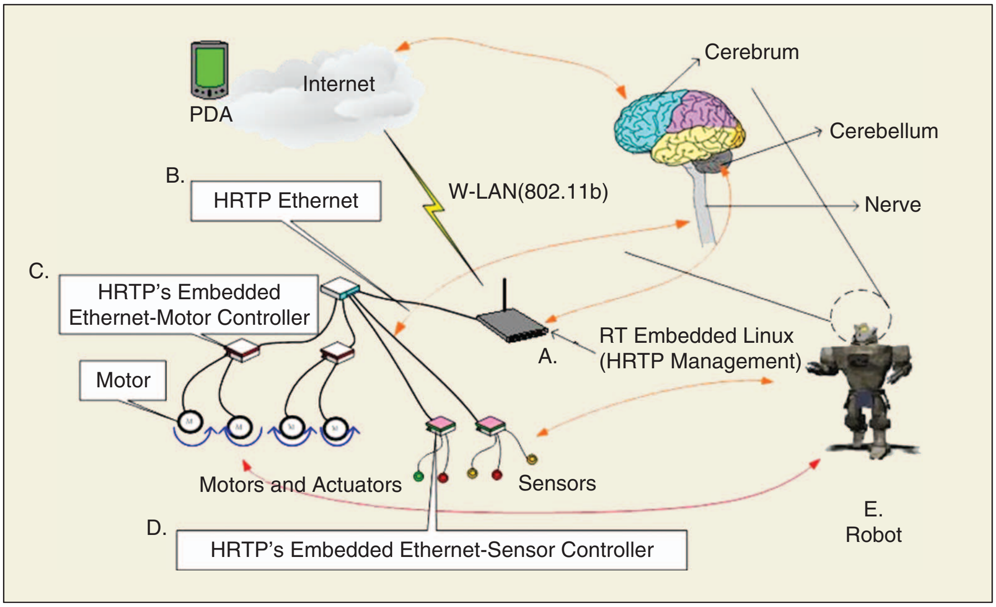

PDA
Internet
Cerebrum
Cerebellum
Nerve
B.
HRTP Ethernet
W-LAN(802.11b)
C.
HRTP's Embedded Ethernet-Motor Controller
Motor
A.
RT Embedded Linux (HRTP Management)
Motors and Actuators
Sensors
D.
HRTP's Embedded Ethernet-Sensor Controller
E. Robot

Figure 13. *Conception of the platform and materialization.*

24LC256) file system. Figure 15 shows this module. Each embedded HRTP Ethernet controller is interfaced with a 10-Mb/s Ethernet NIC (RTL-8019AS). A Media-GX (Geode CS5530A) Pentium-class SoC [43] is applied as the gateway-level network processor (Figure 1, Internet layer). Major peripherals that are connected to the SoC include wireless LAN, 10/100-Mb/s Ethernet NIC (RTL-8139), and the peripheral component interconnect bus.

Robot Mechanical Design and System Integration

Figure 16 presents the computer-aided design (CAD) diagram of the four-footed machine, which we called O-Di robot (O-Di); each foot has 4 DoF. Each joint motor has one potential meter and one proportional-integral differential (PID) controller to control the position. Embedded HRTP Ethernet is applied to update and reload sensor feedback to control O-Di's posture. Figure 17 schematically depicts the integration of the mechanical system with the electronic system. Figure 18 displays a photograph of the fully assembled machine.

Software

The following software or firmware modules were installed to demonstrate the proposed system:

1) firmware on the microcontroller, including the motor control, the NIC driver, and a lean TCP/IP protocol stack [44] with HRTP modification
2) an embedded RT Linux system, running on the Pentium-class SOC
3) a modified Linux TCP/UDP/IP protocol stack to provide HRTP capability
4) a program to integrate robot network bridging and motion control functions
5) a software package for motion control and running on a PC.

The design architecture allows motion control software to be implemented easily on a PC and contains 16 slider bars to manage each individual motor directly. Additionally, it can also generate a database to record the state of motion and playback afterwards in a synchronized fashion. Figure 19 displays the overall system, including the remote control software. Although the present features are rather primitive, the platform clearly has significant potential for future expansion and software developments.

As mentioned earlier, the embedded Ethernet platform and HRTP provide the system communication, integration, power saving, and low-cost advantage. Hence, the proposed

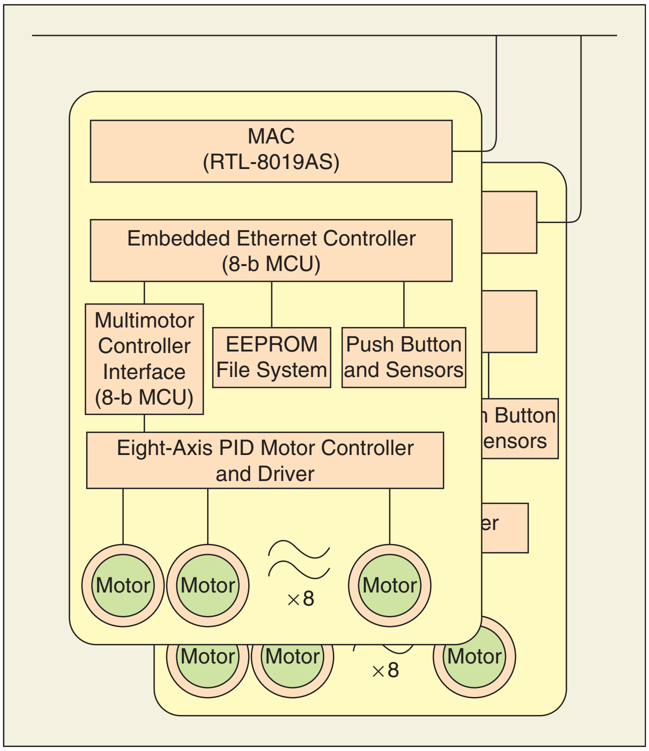


Figure 14. *Schematic diagram of electronics.*

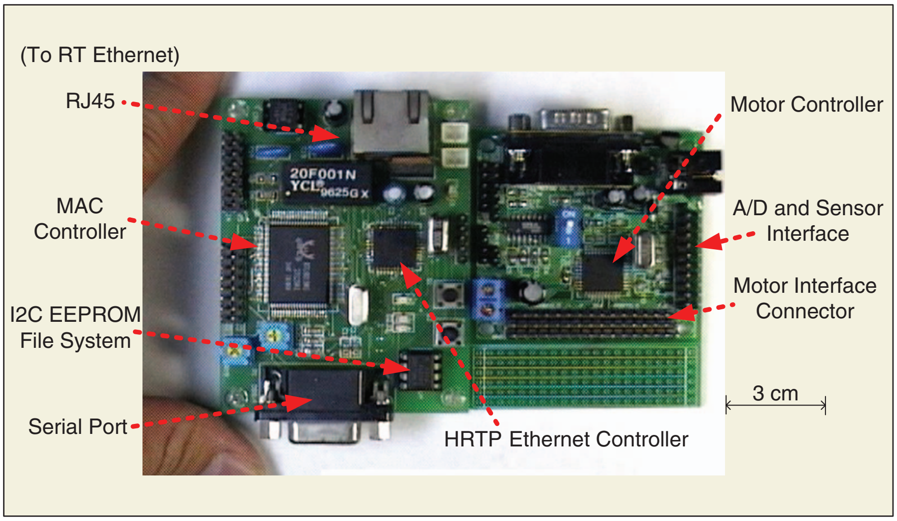


Figure 15. *A prototype board with embedded HRTP Ethernet controller and motor controller module.*

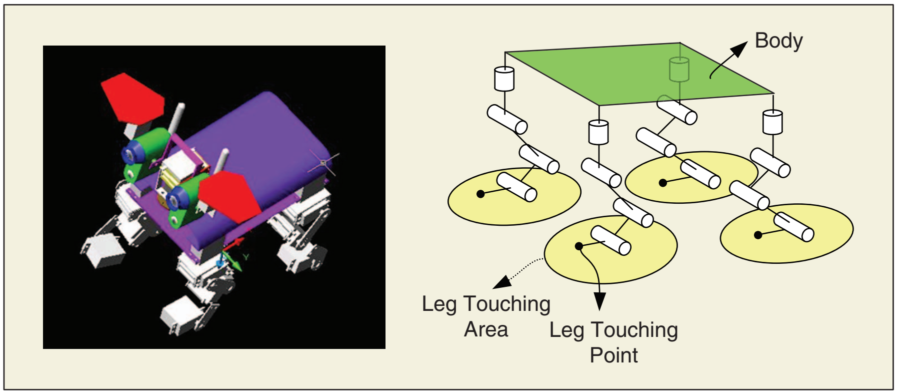


Figure 16. *O-Di's CAD design and DoF.*

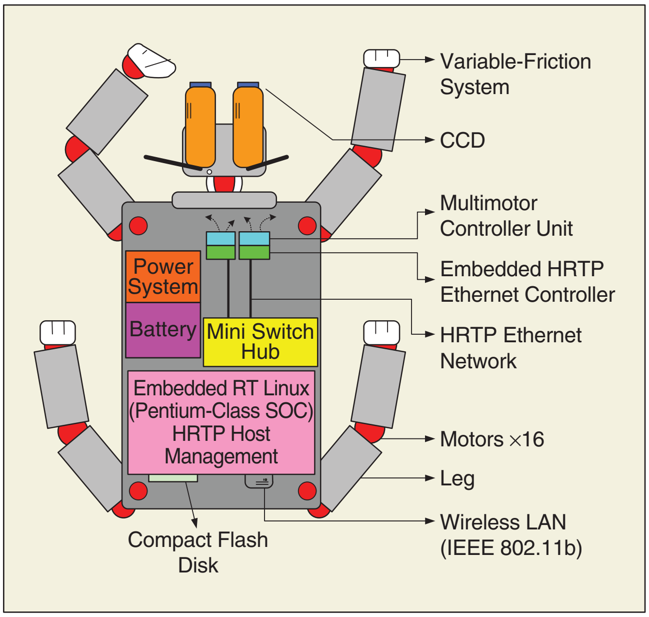

Variable-Friction System
CCD
Multimotor Controller Unit
Embedded HRTP Ethernet Controller
HRTP Ethernet Network
Motors ×16
Leg
Wireless LAN (IEEE 802.11b)
Compact Flash Disk
Power System
Battery
Mini Switch Hub
Embedded RT Linux (Pentium-Class SOC) HRTP Host Management

Figure 17. *Schematic diagram of the O-Di.*

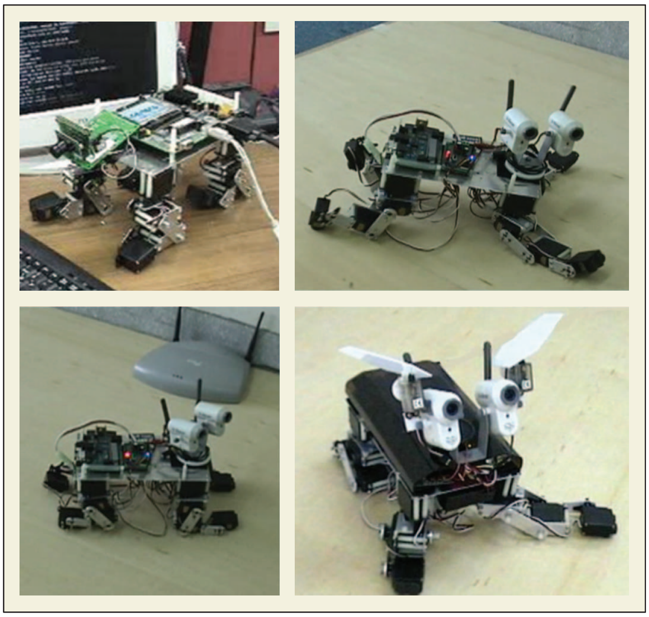


Figure 18. *Fully assembled quadruped machine (O-Di).*

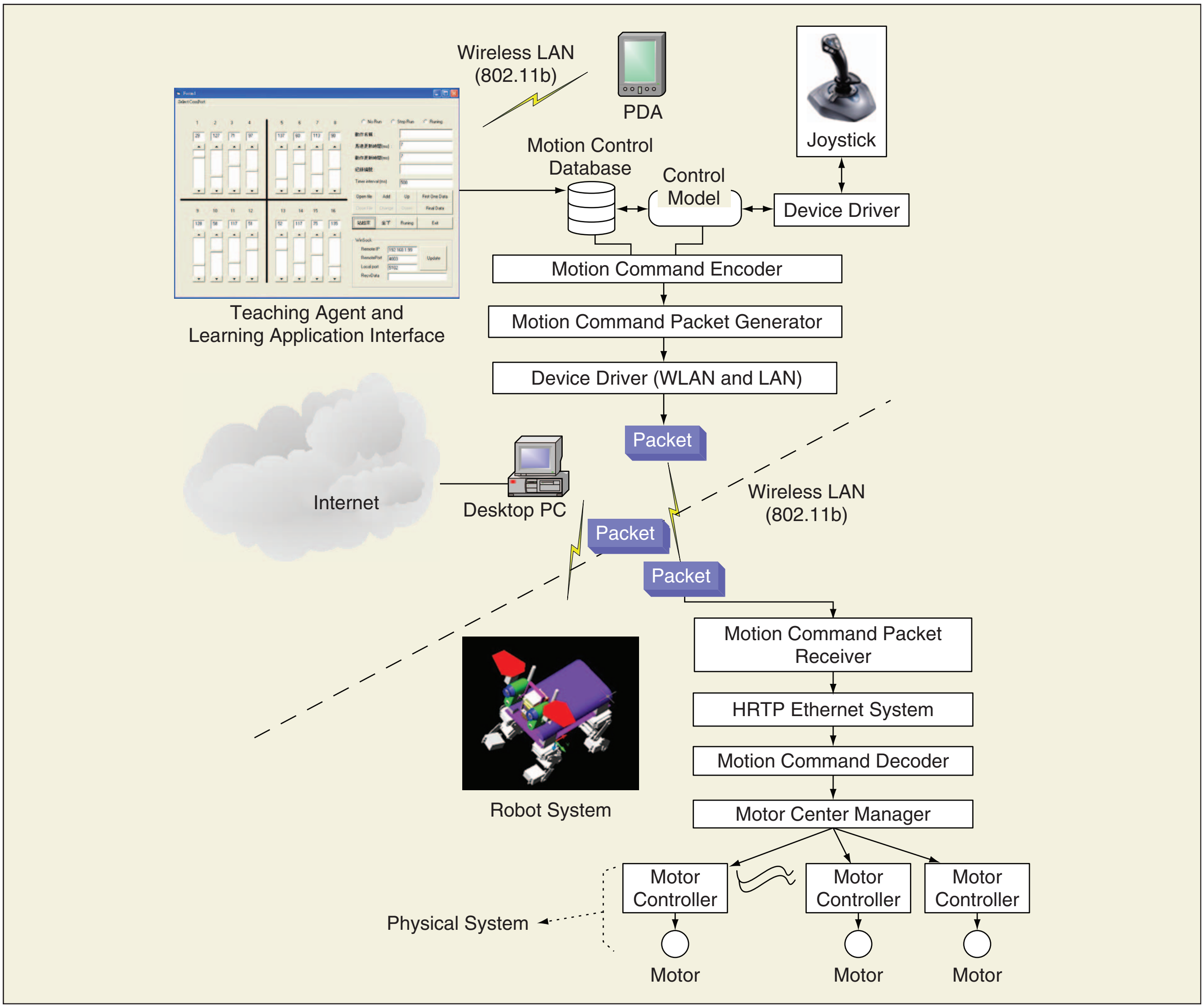


Figure 19. *Overall system including PC-side remote control.*

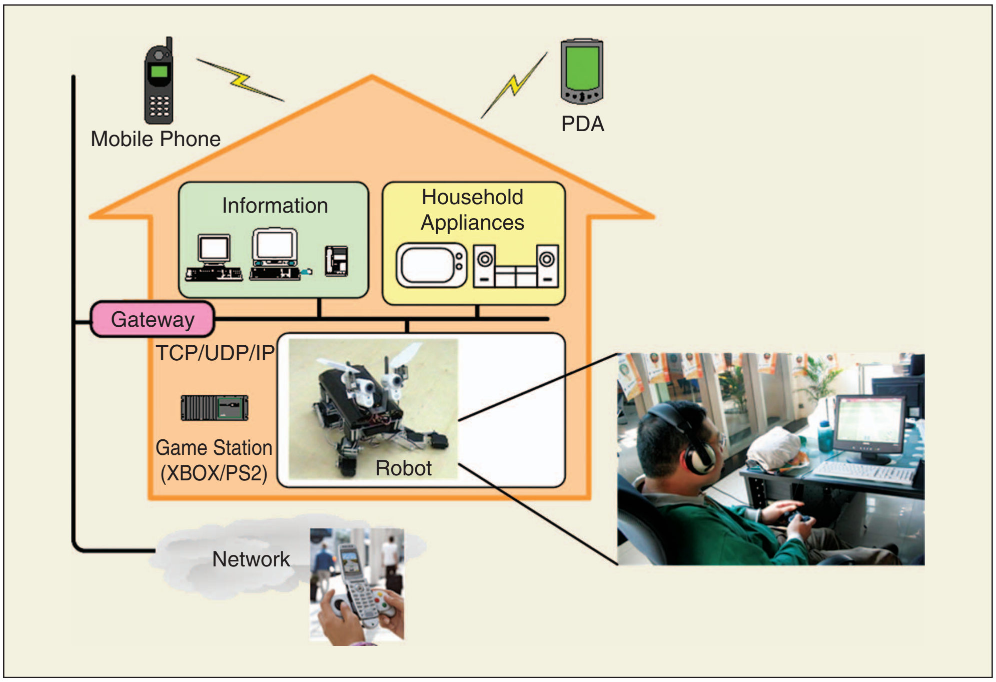


Figure 20. Appling propose robot network platform integrated with today's home network. Various network technologies can be interconnected in home network environments using TCP/UDP/IP.

architecture has potential applications in the design architecture of home network-based robots. Furthermore, this robot platform can be easily integrated with today's home network. Various network technologies can be interconnected in home network environments using TCP/IP (Figure 20).

Example of Motion and System Integration

The clock synchronization on the proposed platform has an accuracy of ±0.5 ms for every 100 ms time-triggered protocol (TTP) cycle. The clock is resynchronized every 1 s, and the maximum error is 5 ms. The motion commands are issued every 50 ms. Restated, a 10% error in motion synchronization results from the timing mismatch. Because the system is low-end CPU, the result of performance is reasonable. For motion profiles, which do not need a high precision, this error is acceptable. The better performance for future can improve by improving CPU performance and raising clock frequency. Figure 21 displays a sample of the feedback control (using the PID algorithm to control a one-axial dc servo motor) in the proposed network system. The HRTP can serve as a low-collision RT network environment. A two-wheel mode controller algorithm [45], [46] was implemented to demonstrate the proposed system's effectiveness. Figure 22 shows a sequence of pictures of the robot in motion. Full robot demo videos can be seen in the Web site http://xlab.cn.nctu.edu.tw/Liwei/eRobot.htm [47].

Conclusions

This study presents a novel distributed embedded platform using Ethernet as the communications backbone, with three layers: 1) control and sensing layer, 2) gateway layer, and 3) Internet layer. This proposal regards the cerebrum, cerebellum, and

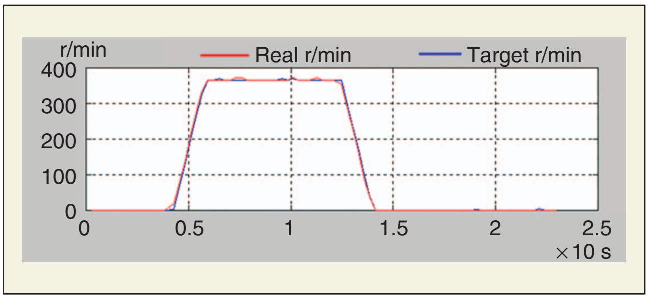


Figure 21. A sample result of motor feedback control (using PID algorithm to control one-axial dc servo motor).

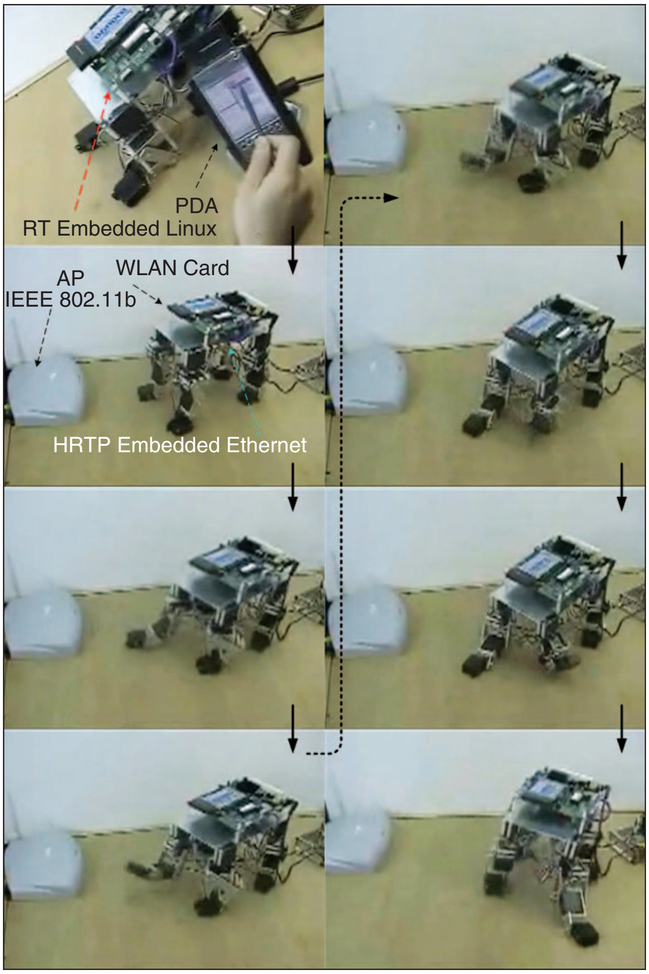


Figure 22. Robot walk around and PDA control robot via WLAN.

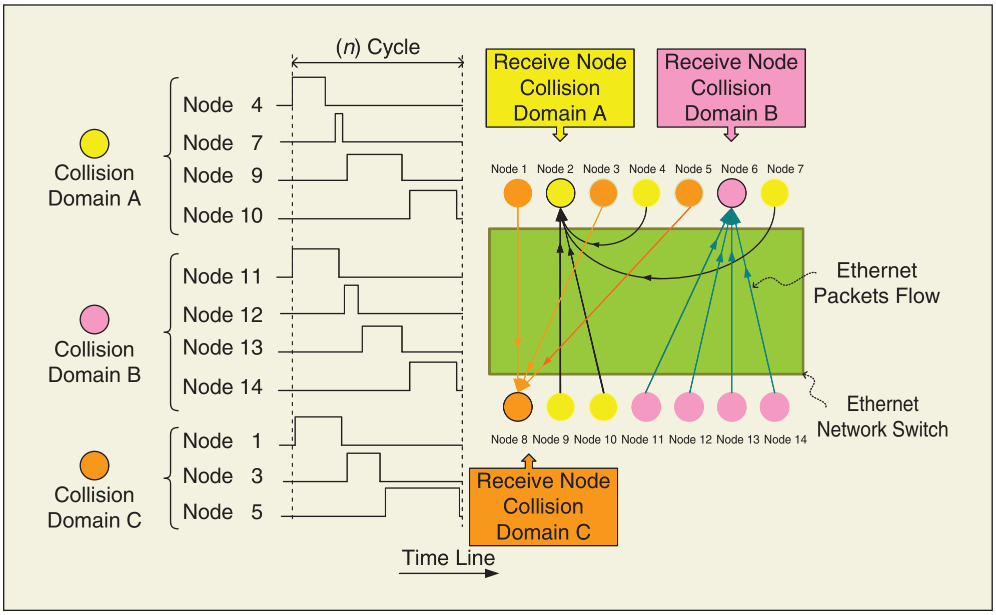


Figure 23. *The three collision domains A, B, and C. The packet flows in an Ethernet switch. The current Ethernet switch technique can provide one transmitting collision domain per port (dedicated bandwidth segment), allowing the complete elimination of collisions if the port is in full duplex (IEEE 802.3x) and only one station is connected to it. The TDMA-E is transmission technology that allows a number of RT nodes to access the Ethernet node without interference, by allocating unique time slots to each RT node within each collision domain. However, it allows only one RT node transmission to access the respective Ethernet receive node in each collision domain. TDMA-E is most responsible for regularizing packet traffic through the group nodes in independent collision domains (first part).*

nerve as abstract concepts. An RT protocol named HRTP is proposed in the control and sensing layer. The HRTP is a distributed RT protocol with a small footprint, which is especially suitable for the embedded and distributed network-based robot control application. HRTP is the first distributed RTE protocol stack specifically for an embedded Ethernet network. Notably, traditional RTE technologies have centralized structures that are large and superfluous for small robot systems. The proposed HRTP, in contrast to a distributed RT protocol, is for embedded RTE environments and robot network development.

A quadruped walking machine (O-Di) and an electronic system were designed to implement the proposed platform. Software modules were installed to demonstrate the overall system. On the basis of the three-layer concept, this model can be imitated to design more complex robot systems. Future research will involve designs of more complex machines to demonstrate the capability of the distributed embedded platform.

Acknowledgments

Part of this work was presented at the 2006 IEEE International Conference on Robotics and Automation (15–19 May 2006) and the IEEE International Conference on Mechatronics (10–12 July 2005). This work led to Taiwan patent no. I238622, Taiwan patent no. I257214, U.S. patent application no. 11/296,217, and U.S. patent application no. 11/370,929. The authors thank the National Science Council of the Republic of China, Taiwan (contract no. NSC92-2213-E-009-004 and NSC94-2218-E-009-064), the Ministry of Education of the Republic of China, Taiwan (contract no. 91-E-FA06-4-4), and Aiming for the Top University and Elite Research Center Development Plan program under the account no. 95W803E for financially supporting this research. This article has author-provided supplementary downloadable material available at http://xlab.cn.nctu.edu.tw/Liwei/eRobot.htm.

Keywords

Network robot, embedded Ethernet, real-time network, distributed control, HRTP, O-Di robot, quadruped robot, real-time Ethernet, home robot.

References

[1] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura, "The intelligent ASIMO: System overview and integration," *Intell. Robots Syst.*, vol. 3, pp. 2478–2483, 2002.

[2] T. Ishida, Y. Kuroki, J. Yamaguchi, M. Fujita, and T. T. Doi, "Motion entertainment by a small humanoid robot based on OPEN-R," in *Proc. 2001 Conf. Intelligent Robots and Systems*, vol. 2, pp. 1079–1086.

[3] T. Makimoto and T. T. Doi, "Chip technologies for entertainment robots—Present and future," in *Tech. Dig. Electron Devices Meeting, IEDM'02*, 2002, pp. 9–16.

[4] B. H. Lee, "Embedded Internet systems: Poised for takeoff," *IEEE Internet Comput.*, vol. 2, pp. 24–29, 1998.

[5] R. E. Filman, "Embedded Internet systems come home," *IEEE Internet Comput.*, vol. 5, no. 1, pp. 52–53, 2001.

[6] J. Axelson, *Embedded Ethernet and Internet Complete: Designing and Programming Small Devices for Networking.* Madison, WI: Lakeview Research, 2003.

[7] D. L. Sancho-Pradel, S. R. Jones, and R. M. Goodall, "System on programmable chip for real-time control implementations," in *Proc. 2002 IEEE Int. Conf. Field-Programmable Technology*, pp. 276–283.

[8] S. Perez and J. Vila, "Building distributed embedded systems with RTLinux-GPL," in *Proc. 2003 IEEE Conf. Emerging Technologies and Factory Automation*, 2003, vol. 1, pp. 161–168.

[9] J. S. Jwo, S. S. Chang, Y. C. Chen, and D. F. Hsu, "A distributed environment for hypercube computing," in *Proc. 2nd Aizu Int. Symp. Parallel Algorithms/Architecture Synthesis*, 1997, pp. 256–263.

[10] J. Wang, (2002). Soft real-time switched ethernet: best-effort packet scheduling algorithm, implementation, and feasibility analysis, M.S. thesis. Dept. Electr. Comput. Eng., Faculty of the Virginia Polytechnic Institute and State Univ. [Online]. Available: http://scholar.lib.vt.edu/theses/available/etd-10032002-152839/

[11] K. Schrodinger, J. Stimma, and M. Mauthe, "A fully integrated CMOS receiver front-end for optic gigabit Ethernet," *IEEE J. Solid State Circ.*, vol. 37, no. 7, pp. 874–880u, 2002.

[12] B. Blaner, D. Czenkusch, R. Devins, and S. Stever, "An embedded PowerPCTM SOC for test and measurement applications," in *Proc. 13th Annu. IEEE Int. ASIC/SOC Conf.*, 2000, pp. 204–208.

[13] J. Bilek and I. P. Ruzicka, "Evolutionary trends of embedded systems," in *Proc. 2003 IEEE Int. Conf. Industrial Technology*, vol. 2. pp. 901–905.

[14] M. Molle, "100Base-T/IEEE802.12/packet switching," *IEEE Commun. Mag.*, vol. 15, no. 10, pp. 64–73, 1996.

[15] C. Venkatramani, "Implementation and evaluation of RETHER: A real-time Ethernet protocol," Ph.D. dissertation, State Univ. New York at Stony Brook, 1996.

[16] C. Venkatramani and T. S. Chiueh, "Supporting real-time traffic on Ethernet," in *Proc. 15th IEEE Real-Time Systems Symp*, 1994, pp. 282–286.

[17] S. Varadarajan and T. Chiueh, "Ethereal: A host-transparent real-time fast Ethernet switch," in *Proc. IEEE Int. Conf. Network Protocols*, 1998, pp. 12–21.

[18] T. C. Chiueh and C. Venkatramani, "Supporting real-time traffic on Ethernet," in *Proc. IEEE Real-Time Symp.*, 1994, pp. 282–286.

[19] H. Scholten, P. G. Jansen, F. Hanssen, and T. Hattink, "RTnet: A new approach to in-home real-time multimedia communication," in *Proc. 12th Workshop on Local and Metropolitan Area Networks*, 2002, pp. 15–19.

[20] K. H. K. Kim, C. Im, and P. Athreya, "Realization of a distributed OS component for internal clock synchronization in a LAN environment," in *Proc. IEEE Object-Oriented Real-Time Distributed Computing*, 2002, pp. 263–270.

[21] M. Schwarz, "Implementation of a TTP/C cluster based on commercial gigabit Ethernet components," M.S. thesis, Institut fur Technische Informatik, Vienna Univ. Technology, Austria, 2002.

[22] M. Felser, "Real-time Ethernet—Industry prospective," *Proc. IEEE*, vol. 93, no. 6, pp. 1118–1129, 2005.

[23] N. Brown and C. Kindel (1996). Distributed component object model protocol—DCOM/1.0, Internet Draft. [Online]. Available: http://www.microsoft.com/oledev/olecom/draft-brown-dcom-v1-spec-01.txt

[24] P. Ferrari, A. Flammini, D. Marioli, and A. Taroni, "Experimental evaluation of PROFINET performance," in *Proc. 2004 IEEE Int. Workshop on Factory Communication Systems*, pp. 331–334.

[25] J. Jasperneite and J. Feld, "PROFINET: An integration platform for heterogeneous industrial communication systems," in *Proc. 10th IEEE Conf. Emerging Technologies and Factory Automation (ETFA 2005)*, 2005, vol. 1. pp. 815–822.

[26] J. Feld, "PROFINET-scalable factory communication for all applications," in *5th IEEE Int. Workshop on Factory Communication Systems (WFCS'2004)*, Vienna, Sept. 2004, pp. 33–38.

[27] K. Schneider, "Intelligent field devices in factory automation-Modular structures into manufacturing cells," in *Proc. Emerging Technologies and Factory Automation (ETFA '03)*, 2003, vol. 1, pp. 101–103.

[28] M. Popp and P. Wenzel, "PROFInet-linking worlds," in *Proc. 2001 8th IEEE Int. Conf. Emerging Technologies and Factory Automation*, vol. 2, pp. 519–522.

[29] F. Hansen (2006). *Real-time Ethernet is reaching the field for industrial control* [Online]. Available: http://www.rtcmagazine.com/home/article.php?id=100489

[30] D. Jansen and H. Buttner, "Real-time Ethernet the EtherCAT solution," *IEE Comput. Contr. Eng. J.*, vol. 15, no. 1, pp. 16–21, 2004.

[31] EtherCAT Technology Group. (2003). *EtherCAT—the Ethernet fieldbus*, [Online]. Available: http://www.ethercat.org/pdf/pcc_ethercat_e.pdf

[32] Bernecker & Rainer (B&R). (2003). *ETHERNET powerlink user's manual* [Online]. Available: http://www.br-automation.com/cps/rde/xchg/br-productcatalogue/hs.xsl/services_66597_ENG_HTML.htm

[33] Bernecker & Rainer (B&R). (2004). *ETHERNET powerlink* [Online]. Available: http://www.br-automation.com/cps/rde/xchg/br-automation_com/hs.xsl/service_5129_ENG_HTML.htm

[34] *Real-Time Ethernet EPL (ETHERNET Powerlink), Proposal for a Publicly Available Specification for Real-Time Ethernet*, document IEC, 65C/356a/NP, 2004.

[35] C. Schlegel (2005). *ETHERNET powerlink implementation strategies* [Online]. Available: http://www.ethernet-powerlink.org/index.php?id=35

[36] *Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specification[S].* IEEE Standard 802.3x2002.

[37] G. Held, *Ethernet Networks: Design, Implementation, Operation, Management.* Hoboken, NJ: Wiley, 2003.

[38] Lantronix. (2004). *Networking tutorials—Network switching* [Online]. Available: http://www.lantronix.cn/learning/tutorials/switching.html

[39] S. Johannessen, "Time synchronization in a local area network," *IEEE Control Sys. Mag.*, vol. 24, no. 2, pp. 61–69, 2004.

[40] H. Melvin and L. Murphy, "Time synchronization for VOIP quality of service," *IEEE Internet Comput.*, vol. 6, no. 3, pp. 57–63, 2002.

[41] D. L. Mills, "Internet time synchronization: The network time protocol," *IEEE Trans. Commun.*, vol. 39, no. 10, p. 1482, 1991.

[42] *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*, IEEE Standard 1588–2004, Nov. 8, 2002.

[43] Advanced Micro Devices Inc. (2004). *AMD Geode(R) GX processors data book* [Online]. Available: http://www.amd.com/files/connectivitysolutions/geode/geode_gx/31505E_gx_databook.zip

[44] J. Bentham, *TCP/IP Lean: Web Servers for Embedded Systems.* 2nd ed., Lawrence, KS: CMP Books, 2002.

[45] D. Golubovic and H. Hu, "A hybrid evolutionary algorithm for gait generation of Sony legged robots," in *Proc. IECON 2002*, pp. 2593–2598.

[46] C. Asghari, D. Golubovic, B. Li, and H. Hu, "A hybrid software platform for Sony AIBO robots," in *7th Int. Symp. RobotCup*, Padua, Italy, 2003.

[47] L.-W. Wu and J.-S. Hu. *Demo video: A distributed embedded control platform for robots using real-time Ethernet* [Online]. Available: http://xlab.cn.nctu.edu.tw/Liwei/eRobot.htm

Li-Wei Wu received his B.S. degree in mechanical engineering from Chinese Culture University, Taiwan, Republic of China, in 1998 and his M.S. degree in mechanical and aerospace engineering from Chung Hua University, Taiwan, Republic of China, in 2000. He received his Ph.D. degree from the Department of Electrical and Control Engineering at the National Chiao-Tung University in 2006. He received the Honorable Mention of the Chinese Taiwan Ministry of Education (championship) in the 2003 Embedded Software Design Contest, which was sponsored by the Ministry of Education Advisory Office and the SoC Consortium. He also received the Honorable Mention of the Chinese Taiwan Ministry of Education in the 2005 Intelligent Robot Contest. His research interests include robotics, RT network protocol, embedded Ethernet, RT systems, and embedded system design and implementation. Currently, he is a senior engineer at Compal Communications.

Jwu-Sheng Hu received the B.S. degree from the Department of Mechanical Engineering, National Taiwan University, Taiwan, in 1984 and the M.S. and Ph.D. degrees from the Department of Mechanical Engineering, University of California at Berkeley, in 1988 and 1990, respectively. He is currently a professor in the Department of Electrical and Control Engineering, National Chiao-Tung University, Taiwan, Republic of China. His current research interests include microphone array signal processing, active noise control, embedded system design, and robotics.

Address for Correspondence: Li-Wei Wu, Department of Electrical and Control Engineering, National Chiao-Tung University, 1001 Ta-Hsueh Road, Hsinchu 30010, Taiwan. Phone: +3 571 2121. Fax: +3 571 5998. E-mail: liwei0503@gmail.com, liwei.ece89g@nctu.edu.tw.