30th Annual International IEEE EMBS Conference
Vancouver, British Columbia, Canada, August 20-24, 2008

Networking and Plug-and-Play of Bedside Medical Instruments

Pongnarin Thongpithoonrat, *Student Member, IEEE,* Paul K. McKneely, Suman Gumudavelli, *Student Member, IEEE,* D. Gurkan, *Member, IEEE,* Frank M. Chapman

***Abstract*—Medical device manufacturers continuously improve instruments with more capabilities at the point of care such as the bedside, operating room, intensive care unit, or emergency room. The instruments are in turn becoming more sophisticated; however, the operation of an instrument is still expected to be locally done by authorized medical personnel. The measurements from these instruments are stored using archaic methods such as a patient history record on a paper. The access to these records is cumbersome and not available unless the medical personnel is present at the point of care. Unfortunately, each medical instrument has its unique proprietary API (application programming interface - if any) to provide automated and electronic access to monitoring data. Integration of these APIs requires an agreement with the manufacturers towards realization of interoperable health care networking. As long as the interoperability of instruments with a network is not possible, ubiquitous access to patient status is limited only to manual entry based systems. Manual entry is being implemented to create electronic health records, HL7, and similar initiatives. However, they do not address a bottom-up automation (i.e. from instrument and patient bed side up) to leverage the mature networking technologies in a health care setting. This paper demonstrates an attempt to realize an interoperable medical instrument interface for networking using MediCAN technology suite as an open standard. We will present the approach with a comparison study of a similar initiative led by ISO/IEEE 11073 standards.**

I. INTRODUCTION

MEDICAL professionals need to be present at the location of all medical instruments such as at the bedside of a patient in order to capture vital information. In many cases, medical professionals may have to manually control and adjust parameters of an instrument according to the measurements from a different instrument. Statistical research on personalized care is hard to execute when data is scattered and access is so limited. Capturing and entering patient data is still a tedious, time-consuming and error-prone process even though the information has to be real-time and potentially very sensitive for the wellbeing of the patient. The patient care itself is prone to errors because of error-prone manual recording of instrument measurements. There is a need for a common standard which allows for internetworking with the medical devices from different manufacturers.

In 2000, CEN, ISO and IEEE joined to build a single set of standards called ISO/IEEE 11073 for point-of care device communications to unify the interfaces of all medical devices [1-3]. Two of these five 11073 standards, ISO/IEEE 11073-30200 (cable-connected) and ISO/IEEE 11073-30300 (infrared-wireless), provide communication services and protocol definitions, consistent with IrDA specifications (Infrared Data Association) and adapted as appropriate for ISO/IEEE 11073 applications. However, ISO/IEEE 11073 has not been able to generate meaningful adoption by the industry. With advances in network communication technology, many researchers have been trying to connect isolated bedside medical instruments into a network. Many manufacturers have developed their own proprietary solutions failing to gain general acceptance.

MediCAN™ technology suite creates the interfacing hardware and related communication protocol in an open standard fashion for instruments to network in any healthcare environment. MediCAN system works towards a similar goal as ISO/IEEE 11073 (X73) [4, 5] in being a candidate to become an open standard. MediCAN addresses communication services and protocol definitions based on Control Area Network (CAN) communication. MediCAN™ uses instrument adaptors and networking equipment to connect instruments on a CAN bus and then to a network such as the Ethernet.

In this paper, a comparison of communication services and protocol between X73 and MCP (MediCAN Control Protocol), one of protocols of MedICAN providing communication between medical devices and a Gateway [6], will be presented.

II. IEEE 11073 WIRED AND WIRELESS

Both wired and wireless versions are intended to provide communication between medical devices and external computer systems with plug-and-play and interoperable interfaces. Wireless and wired physical/data link layer is based on IrDA specifications where wired runs on a RS232-compatible cable. Based on IrDA specifications, the connection link between Device Communications Controller (DCC) and Bedside Communications Controller (BCC) is a half-duplex, point-to-point communication.

A. Topology

BCC or primary node is a hub connecting to local or remote external site via LAN or WAN. DCC or secondary node connects each medical instrument to the communication network via BCC. Each device has a DCC and a BCC connecting to a gateway [1], Fig. 1.

Manuscript received April 7, 2008.

S. Gumudavelli, P. Thongpithoonrat, and D. Gurkan are with the Engineering Technology Dept., University of Houston, Houston, TX 77204-4020 USA (phone: 713-743-4037 e-mail: dgurkan@uh.edu).

P. K. McKneely and F. M. Chapman are with the MediCAN Systems Inc., Houston, TX 77079 USA (e-mail: fchapman@medicansystems.com).

B. Protocol Stack

Infrared Link Access Protocol (IrLAP) provides reliability, order of data, and device discovery from DCC to the BCC. Infrared Link Management Protocol (IrLMP) manages the Global Identifiers of DCCs and multiplexing of services on an IrLAP link. Tiny transport (TinyTP) provides flow control for IrLMP. Simple Network Time Protocol (SNTP) provides time synchronization service to BCC as a server and optionally to DCC as a client.

Data Link Layer: Infrared Link Access Protocol (IrLAP): BCC as a primary node would initiate a transaction such as device discovery or link negotiation. The secondary node (DCC) responds when spoken to by the primary. The medium is occupied by a primary-secondary device pair at any given time. IrLAP consists of four phases: Device Discovery, Link Negotiation and Connection Establishment, Information Exchange, and Disconnection.

Frame Format: There are four separate frame formats corresponding to the four types of data rates. Every frame has an address, a control field, and the payload. Address is 1 byte long: a 7-bit connection address generated by the primary during connection establishment phase; and a 1-bit Command/Response (C/R) value. Control field defines three types of frames: Unnumbered frames used during device discovery, negotiation, and connection establishment; supervisory frames used for link management such as flow control and rejection of malformed frames; and information frames used to contain actual data between devices. Payload length is multiple of 8 bits up to 2048 bytes.

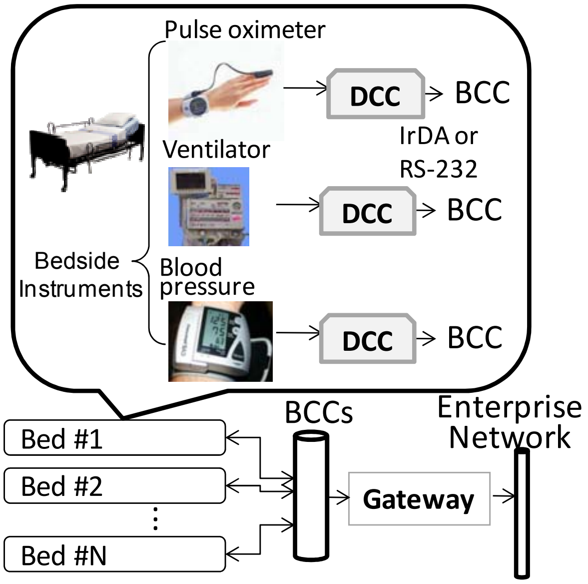

Figure 1. X73 provides interfaces of instruments to the network through DCC-BCC pairs. Implementations of this architecture has been limited to single PC demonstrations. The communication is based on IrDA and RS-232.

C. Communication

Device Discovery: Initially, any node sets its own parameters to default values. For example for wireless default values can be data rate = 9600 bps; window size = 1; packet size = 64 Bytes, min turnaround = 10 ms; XBOFs (additional beginning of frame for synchronization) = 10; max turnaround = 500 ms; link disconnect threshold = 40 sec. To prevent interruption of ongoing communication, the primary must wait at least 500 ms (max. roundtrip delay) and then sense the medium. If there is no transmission, the primary is free to initiate device discovery. The primary provides the number of slots (can be 1, 6, 8, or 16) to be used by sending an Exchange Identification Frame (XID). Secondary devices will randomly reply in one of the provided time slots. In cable-connected approach, only 1 slot is used. If multiple devices are found (typically in wireless infrared link), options are alert the user to choose the correct one or look at a hint bit and automatically connect to that correct hint bit device. After primary established a connection with a particular secondary node, the secondary node sends a XID response frame with: (i) Global Identifier address (64-bit device address); (ii) Device nickname (a character string to help provide device details); (iii) Hint bits (one or more bits identifying desired device).

Link Negotiation and Connection Establishment: The primary issues a Set Normal Response Mode (SNRM) frame with its preferred link parameters in order to invite the secondary device to establish a connection. The negotiated parameters are: Baud Rate (up to 115.2 kbps for cable-connected standard and up to 16.0 Mbps for infrared-wireless), Maximum Turnaround (MTA) Time (max 500 ms), Data size (up to 2048 Bytes, depends on MTA time), Window size (always 1 for slow rates whereas with higher rates 7 is available, window size of 128 is available for 16 Mbps), Additional BOFs (the maximum can be 48 XBOFs), Minimum Turnaround time (as low as 0.01 ms), Link Disconnect/Threshold Time (up to 40 sec). The secondary node can either reject the invitation by sending back a Disconnected Mode (DM) frame or accept the invitation by sending back an Unnumbered Acknowledgment (UA) frame with its preferred link parameters. Then, both sides apply the same algorithm to find their best common attributes, and finally reach the same conclusion. If the secondary node accepts the invitation, the primary node sends a Receive Ready (RR) frame at the negotiated rate and the secondary responds at the new data rate with its own RR frame.

Information Exchange: Nr (next frame number) and Ns (current frame number) fields are used to order the packets. Automatic Repeat Request (ARQ) is used for flow control. Receiver requests the dropped packet: the sender resends all packets counted from the packet requested up to the current packet. An RR frame is used for acknowledgment.

Disconnection: Negotiated Link Disconnect/Threshold time expires, or, one of both sides can disconnect by issuing a Disconnect (DISC) frame, which is acknowledged by a UA frame. All parameters are reset to default.

III. MEDICAN CONTROL PROTOCOL

MediCAN Control Protocol version 1.0 (MCP) is the protocol layered over on ISO 11898:1995 CAN 2.0B

(Control Area Network) defining physical layer and data link layer used for integrating medical instruments into the MediCAN System. MCP provides plug-and-play capability.

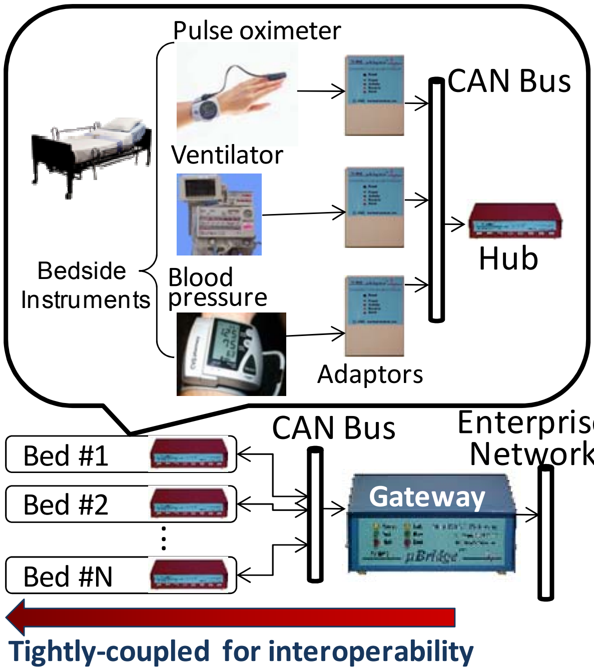

Figure 2. MediCAN system is composed of instrument adaptors at the bedside connected to a hub and then with other beds to a gateway to be accessed by users of the enterprise network.

A. *Topology*

MCP allows direct communication only between a primary node called the Gateway (GNode) and one or more Device Nodes (DNodes) using a shared broadcast CAN bus as shown in Fig. 2. A DNode is the local access point for one or more medical devices. The maximum number of DNodes that are connected to a single GNode is limited to number of ports on GNode. According to CAN 2.0B, the maximum data rate of a 40-meter-long bus is 1 Mbps. GNode is a system providing an access point between a medical instrument network (using MCP) and an Ethernet network (using the MediCAN Gateway Protocol - MGP) over UDP.

B. *Protocol Stack*

Physical layer defines how signals are actually transmitted and deals with the description of Bit Timing, Bit Encoding, and Synchronization. CAN is a carrier-sense multiple-access protocol with collision detection and arbitration on message priority (CSMA/CD+AMP). With CSMA, each node on a bus must wait for a prescribed period of inactivity before attempting to send a message, and with CD+AMP, collisions are resolved through a bit-wise arbitration based on priority of each message in the identifier field of a message: the higher priority always wins the bus access. Receiver nodes check the consistency of a message using the CRC field, acknowledge a consistent message or flag an inconsistent message using the last 2 bits in a CAN frame.

Frame Formats: Unlike a traditional network such as Ethernet, CAN does not send large blocks of data, rather it sends many short messages broadcast to the shared bus. Because MCP frame is mapped onto the CAN 2.0B frame, MCP frame format follows the same structure with modifications to accommodate the prioritization scheme of medical instrument data transfer.

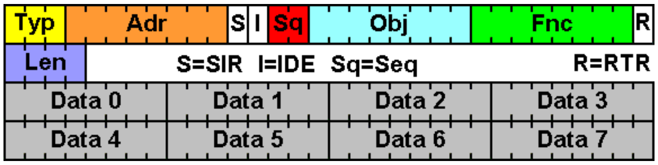

Figure 3. MediCAN Control Protocol frame format.

Typ field provides the direction of packet GNode to/from DNode. More detail on MCP is presented in [6].

C. *Communication*

Connect/Grant: When a DNode is first connected to an MCP network, it has no address and it takes the first action. The DNode starts with a request for connection to the GNode by issuing a ConnectionRequest (ConReq) frame, containing its registered Global Unique Identification (GUID) in Data field and 16-bit CRC of the GUID in Obj and Fnc fields. If GNode grants a specific DNode, it sends a reply (ConGnt) frame returning the GUID in the Data field and an 8-bit DNode address in Obj and Fnc fields.

Function call (polling mode): A function call consists of two parts: a Request from the GNode to one of the DNodes and a Response from DNode back to GNode. Each of these messages consists of one or more frames. GNode is only allowed to have one Function call at a time, but it can have many Function calls in progress for more than one DNode.

Request Phase: GNode initiates a Request Data frame (FReqDat) to a particular DNode, specifying which object and function to be called. The GNode sets its timer to wait for a reply within 250 msec. DNode sends an acknowledge frame (FReqAck) back to GNode. A request message can consist of more than one frame by using Next Request Data frame (NReqDat). Acknowledgement of the Final Request frame ends the Request phase of the Function call. The GNode then sets its timer and expects DNode to begin the Response phase by sending an actual Data frame. If a timeout occurs, the GNode does not send any more frames for this Function call, and it aborts the transaction.

Response Phase: DNode sends a Response frame (FRspDat) to the GNode with maximum of eight data bytes in the Data field. The DNode then set a timer to wait for the first acknowledgement frame (FRspAck) from GNode within 250 ms. A timeout aborts the transaction. If the Response message has more than one frame, Next Response frames will be used until the final (NRsqDat). Each response frame requires an acknowledgment frame (NRsqAck).

Report (pushing mode): Report transaction type is a series of frames going in only from a DNode to GNode. Reporting consists of one or more frame(s) with data followed by a frame containing a time stamp for the final frame. There are three types of variables: Parameter, Event and Stream [5]. Parameter is used to report numerical value of variables. Event reports string characters of states of a variable. Since

Parameter and Event variables only report changes, they only require a single value and therefore only a single frame is sufficient. Stream variable continuously sends frames of data such as a waveform from a Pulse Oximeter. No handshaking acknowledge frames are needed in Report transactions of MCP. However, the CAN controller acknowledges receipt of frames using built-in acknowledgement mechanism in CAN 2.0 B frame.

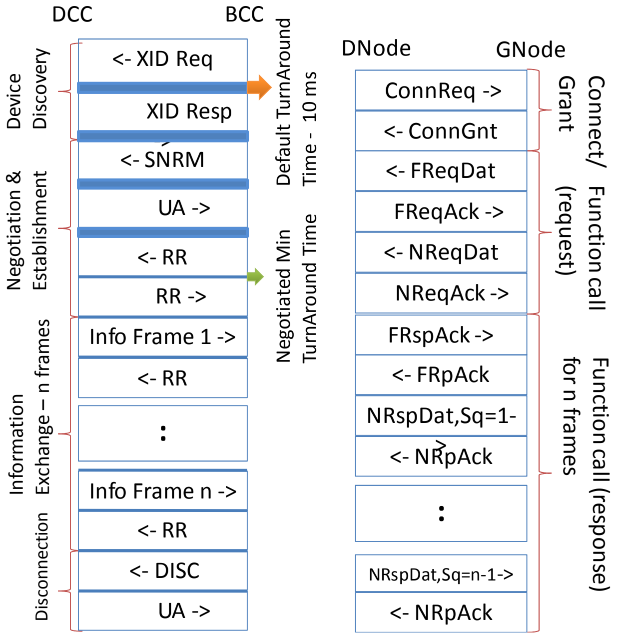

Figure 4. (a) MCP (b) X73 Communication.

IV. PERFORMANCE COMPARISON

We compare transmission times used to transmit actual data of 1024 bytes for two cases of X73 standards and MCP with some assumptions for a simple calculation, Fig. 4. For X73, after information exchange phase, we choose data rates of 9.6 kbps and 4 Mbps with window side of one, and minimum turnaround time of 0.01 ms. We did not include disconnection phase. For MCP, the shared bus operates at 1 Mbps and to be comparable with X73, we assume there is only one DNode communicating with one GNode although a gateway can potentially handle 8 or more ports with 8 or more instruments on each port. There is no processing time, propagation delay, and packet lost in the calculations. Since we calculate only MCP frames, we included only IrLAP payload in case of X73. Only one function call is assumed for 1024 actual bytes data and GNode transmits two frames in request phase.

To establish the link between a pair of GNode and DNode, MCP the time spent for Connect/Grant is approximately 0.1 ms. Since each MCP frame can contain 8 bytes of data, we need 128 frames to transmit in response phase. However, for each frame sent in any direction, it is necessary to have a corresponding acknowledgement frame from receiver. The time spent during the function is approximately 17.16 ms. If we combine transmission time of Connect/Grant and Function Call, the total is approximately 17.26 ms.

For X73, we assume that information field in IrLAP frame is not used unless frames carry information; therefore, only Address and Control fields are used. Before Connection Establishment phase of both cases of 9.6 kbps and 4 Mbps, the default parameter values are used. Therefore, both spend the same amount of time for Device Discovery and Negotiation totaling approximately 49.16 ms.

During Information Exchange phase, we use maximum Packet size of 256 and 2048 bytes for 9.6 kbps and 4 Mbps, respectively. Therefore, 9.6 kbps transmission needs four frames and 4 Mbps only one frame. In addition, because window size is one, each information frame from DCC requires acknowledgement frame from BCC.

Together with transmission time during Connection Establishment, 9.6 kbps spends approximately 870 ms, and 4 Mbps approximately 0.3 ms. As a result, the total time from the beginning to the end for 9.6 kbps and 4 Mbps approximately 919.16 and 43.46 ms, respectively.

From above calculations, it is obvious that X73 spends a lot of fixed time for link establishment before actual data can be transmitted. Moreover, Min Turnaround time also a major factor of X73 latency. Nevertheless, we can realize that if the size of data is much more than 1024 bytes, 4.0 Mbps can operate with less latency. In addition, window size of seven can be implemented according to IrDA standards, which would allow a better performance. MCP, on the other hand, spends less time for link establishment (Connect/Grant), only 0.1 ms. However, this can vary based on the number of DNodes on the shared bus. It is worth noting that MCP provides multiple access. During information exchange, MCP spends less time compared with that of 9.6 kbps, but more with that of 4 Mbps.

From the previous section, MCP provides Report phase for streaming data without need for request frames from GNode and, more importantly, acknowledgement frames due to using built-in CAN acknowledgement mechanism.

Finally, from user's point of view, the flexibility of setup might be a trade-off between these two approaches. As of now, MCP has been successfully implemented and tested on a prototype.

REFERENCES

[1] M. Gallaraga, et.al., "Proposal of an ISO/IEEE 11073 platform for healthcare telemonitoring: plug-and-play solution with new use cases," 29th IEEE EMBS Annual International Conference, 2007.
[2] S. Warren, J. Yao, R. Schmitz, and J. Lebak, " Reconfigurable point-of-care systems designed with interoperability standards," IEEE EMBC , 2004. Vol. 26, pp. 3270 – 3273.
[3] IEEE Standards Association page: http://standards.ieee.org/
[4] R. J. Kennelly and R. M. Gardner, "Perspectives on development of IEEE 1073: the medical information bus (MIB) standard," Int. Journ. Clin. Monit. Comput., vol. 14, pp. 143-149, 1997
[5] Suman Gumudavelli, Paul K. McKneely, Pongnarin Thongpithoonrat, D. Gurkan, Frank M. Chapman, " Medical Instrument Data Exchange," submitted to the IEEE EMBC 2008, April 2008.
[6] P. K. McKneely, F. Chapman, and D. Gurkan, "Plug-and-Play and Network-Capable Medical Instrumentation and Database with a Complete Healthcare Technology Suite: MediCAN," 2007 HCMDSS and Medical Device Plug-and-Play Interoperability.