

A Framework to Transmit Process Control Data over Commercial Wireless Networks

Mark Nixon, Rusty Shepard,
Bill Bennett, Deji Chen
Emerson Process Management
12301 Research Blvd. Building III,
Austin, TX 78759

Aloysius K. Mok
Dpt. of Computer Sciences
University of Texas at Austin
Austin, TX 78712

KEYWORDS

Wireless network, Process Data, Middleware

ABSTRACT

Wireless draws huge attention nowadays. People from the real-time and embedded community are also looking at introducing wireless into mission-critical environment. Lots of work has been done on the pros and cons of applying wireless technology to process control. In this paper we take a different approach. We assume that we have to use wireless and analyze its adoption in process control. In particular, we shall consider using commercial of the shelf (COTS) wireless products to transmit process data between two remote sites. We look at a common middleware approach that abstracts the support of the underlying wireless network. This separates the process control itself from the idiosyncrasy of wireless. As an example, we apply this approach to oil extractor system, a real world application.

1. INTRODUCTION

Wireless is all the rage. We do not need to repeat its benefit here. It suffices for us to say that those who ignore it risk missing the biggest opportunity.

In essence, wireless technology is about data transmission, and data transmission is a key part in process control. The benefit offered by wireless network is appealing for process control. An example is offshore oil drilling where systems on land communicate to systems on drilling platforms. In places where wire is difficult or expensive to deploy, satellite communication maybe the best choice. Using commercial wireless network makes possible process control that was previously unachievable.

Communication bandwidth, disturbances, and cost have been very real impediments to moving monitor and control techniques into a much wider set of problems. Oil exploration, production, and distribution are areas where improved control would provide considerable benefits.

However, we are not going to debate if we should or should not apply wireless technology to process control. Rather, we investigate how we could apply it. We first look at a typical process control system in Section 2. We formalize the role of wireless network at different communication levels of a process control system. In Section 3 we analyze a possible framework in which a middleware separates the idiosyncrasies of wireless and process control. This reduces the complexity of the problem. We shall illustrate this approach with a real world oil extractor example in the next section. The last section, Section 5, concludes the paper.

2. PROCESS CONTROL SYSTEM AND WIRELESS NETWORK

Figure 1 is a distributed process control system. There are three layers of networks, level one is between the smart devices and the controllers; level two is among the controllers and the workstations; and level three is from the workstations to the outside world. Level one runs the process. It has tight real-time requirement of high predictability and reliability. The network protocols are usually industry standards like HART, Foundation Fieldbus, DeviceNet, etc. Level two supports user interaction, including configuration, control, and monitoring. It has less timing requirement than level one, but still requires good reliability. The network protocol could be proprietary, or industry standards such as Ethernet. Level three could be considered a gateway of the control system to other corporate systems, like accounting, inventory, management decision systems, etc. It would be nice to have good support at this level, but conventional networks are deployed, such as office network, intranet, etc.

Wireless as a kind of network, it could be used to target any of the three network levels of a process control system. Level one requires reliable short-range data transmission. Data package sizes are small, usually less than one hundred bytes. There is much debate on whether wireless could be used at this level. Of different wireless technologies, ZigBee, Wi-Fi, BlueTooth, Ultra Wideband (UWB), etc. work well at short range. One could implement Fieldbus on top of them, or define new process control fieldbus standards for wireless communication. Self healing mesh technologies such as that developed by Echelon [6] are finding a place in this space. Level two networks have longer transmission ranges and bigger data package sizes. They have stricter requirements than COTS networks. People in the process control industry are already using wireless technology at this level. Satellite, Wi-Max, etc. supports long range wireless transmission; microwave and radio are used for shorter distances. In comparison, bringing wireless to the third level draws far less concern from people of process control. Any conventional methods to replace wireline with wireless in non process control world apply to this level.

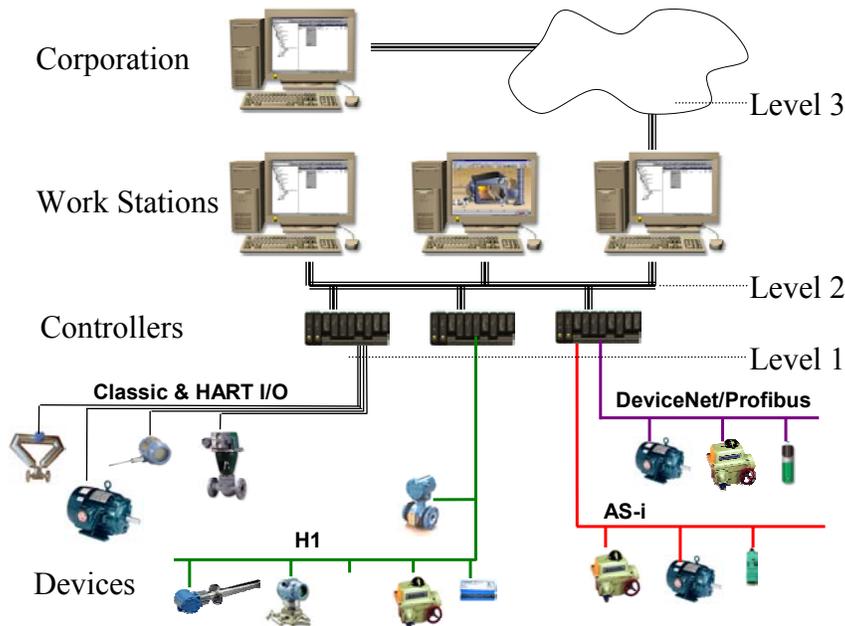


FIG 1 - A DISTRIBUTED PROCESS CONTROL SYSTEM

Wireless as a new type of technology could introduce new support for a process control system. Wireless has applications in display and handheld functions as well. For example, people talk about a handheld device that could retrieve data wirelessly from field devices.

Wireless has its own challenges compared with wireline, such as bandwidth, delay, noise, security, etc. Data communication in process control introduces further challenges like delivery guarantee, delay guarantee, continuous data request, asymmetric data traffic, etc. When wireless network is to do the process control work, even more challenges add up, such as reliability, cost structure, etc.

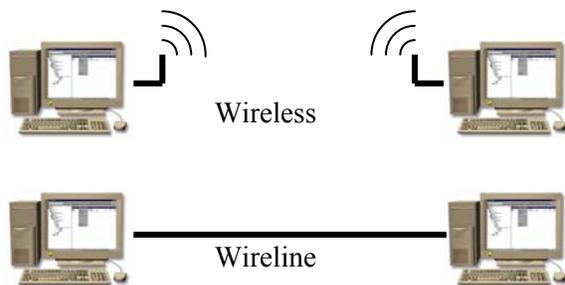


FIG 2 – WIRELESS VERSUS WIRELINE

The difference between wireline and wireless is big at the lower layer, but at a higher layer, the differences could be reduced to different parameter values, especially when both implement the same network protocols. Figure 2 shows the physical difference

between the two in terms of communication between two end points. For wireline, the connection is one piece. The connection has a set of properties like bandwidth, delay. For wireless, the connection contains two pieces, one at each end. In addition to normal wireline properties, it has additional properties like noise, error rate, etc. If a common network protocol is implemented at the top, like TCP/IP, the user of the connection will not see any difference other than the quality of the connection. This is the key point we are going to use in the next section.

3. A FRAMEWORK OVER COMMERCIAL WIRELESS NETWORKS

The approach we look at in this section is to replace the wireline network in a process control system with a wireless one, with minimum possible change to the existing control system. We shall look at replacing the level two networks. Figure 3 is the resulting level two network from Figure 1.

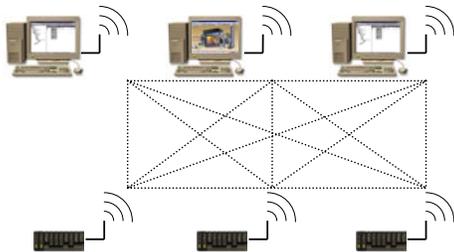


FIG 3 – WIRELESS NETWORK AT LEVEL TWO

Note we assume the existence of the control system with wireline networks. This paper is not about building a new control system from ground up. We also assume that the level two uses standard network protocol. If the wireline network uses proprietary protocol, the wireless version would have to implement the same protocol in order not to re-implement the control system. Assuming a standard protocol enables us to apply COTS wireless networks. Since almost all COTS wireless networks support TCP/IP, we further assume that, without loss of generality, the control system uses TCP/IP to communicate between the distributed workstations and controllers. In this way the control system runs but with flaws if we substitute the wireline network with the wireless counterpart.

If the wireless network provides identical support as the wireline network, our job is done and the paper could end here. Unfortunately, we have to continue to address the difference and its effect on the performance of the original system of wireline network. In the following we shall define the framework to mitigate the flaws.

The first subsection defines the middleware that captures the specifics of the wireless network. Subsection 3.2 discusses how the specifics are maintained during runtime. Subsection 3.3 talks about necessary changes required for the control system. We think it is not possible without changes to the control system. And finally Subsection 3.4 talks about things to observe when configuring a control system with wireless network.

3.1 The middleware

We assume a basic commercial wireless network offering, on top of which we build a middleware for process control data communication. The middleware provides an application programming interface (API) to the process control system. The control system sends data to the middleware with transmission constraint requirements and little knowledge of the underlying wireless network. Figure 4 shows the layering of the middleware.

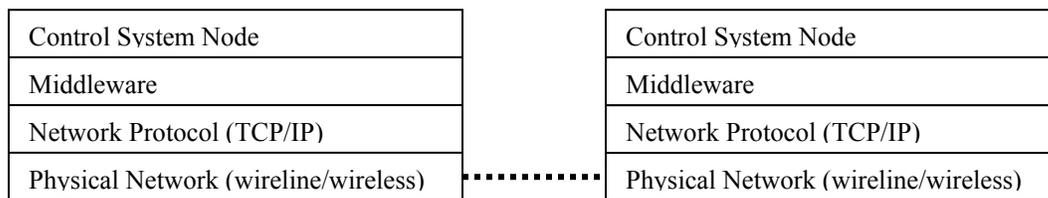


FIG 4 - THE PLACE OF THE MIDDLEWARE

The API provided by the middleware for the control system supports regular data communication, such as initialize, open, close, send, receive, acknowledge, cancel, etc. The middleware executes these commands by calling the corresponding functions at the lower layer. In addition, the middleware will monitor the performance of these calls. This is captured in a set of parameters in the middleware. Since we use commercial wireless network, no extra support could be demanded from it. So the difference of parameter values tells the difference of the underlying wireline and wireless networks. The parameters will keep track of the following, some of which are configurable:

- Flag indicating if the node is on the network
- The type of network connection
- Message timeout value
- Message roundtrip delay
- Number of message retries
- Flag indicating if message data is encrypted, and the type of encryption
- Flag indicating if the node has a redundant network connection
- Auto message timing indicating whether the low level automatically calculates the message timeout and re-tries based on round trip timing or use the configured values
- Flag indicating if there is failed communications or bad integrity

- License information
- Security information

3.2 Execution of the middleware

During execution, the middleware carries out the commands from the control system and maintains the parameters current. Ideally the middleware would be able to automatically adjust itself by taking into account the response time, bandwidth, number of packets, number of unacknowledged packets, delay times, and cost - perhaps even choosing between transmission media based on time-of-day, etc.

One major work the middleware should perform is to maintain network connections. Wireless network could be intermittent. The middleware should handle this gracefully. Depending on the configuration, the connection could be permanent between two nodes, or could be ad-hoc whenever connection is available. In the case of redundant connection, the middleware should switch between primary and standby connections without the notice of the control applications. The middleware could add variable retry and timeout times for each connection to account for propagation delays. Or it could allow multiple outstanding messages for more efficient use of bandwidth.

During active data transmission period, the middleware should perform a suite of housekeeping tasks. These include:

- Round trip delay timing. The initial round trip delay is based on the time between the synchronous request and the synchronous response during connection establishment. The round trip delay should be updated over time by timing the delay between sending a message and the receipt of an acknowledgment. Round trip delay value is added to the message header to communicate the initial value to the passive end of the connection and to keep both ends of the connection in agreement of the current value.
- Send message processing. Remote connections need to have the ability to support multiple outstanding sent messages. We could add a window parameter that defines the limit on how many messages are to be outstanding at a time, change the message sending function to send all messages on the pending message queue up to the window limit, request an acknowledgment only on the last message sent, and add a timer to trigger sending messages that have queued while waiting for an acknowledgment on a previous message.
- Receive message processing. When acknowledging a message, copy the time sent value from the received message into the acknowledgment message. This will be used to calculate round trip times and will provide a mechanism for validating that the acknowledgment is associated with the acknowledged message. When an out of order message is received, return an acknowledgment for messages that have been accepted to this point. This will prevent messages that have already been received but not yet acknowledged from being needlessly retransmitted. When an acknowledgment is received, the time sent value is checked against the

time sent value in the message being acknowledged. If these times match, the round trip time will be averaged into the round trip time value.

- **Retry and timeouts.** To support multiple outstanding messages, each message must have a timeout value, not one time for the connection. As messages are acknowledged, they are removed from the acknowledgment timeout queue leaving remaining messages to be timed out at the proper time. An acknowledgment may acknowledge multiple messages. All messages within the window that precede the sequence number being acknowledged are also acknowledged and must be taken off the retransmit queue. Acknowledgments for messages with sequence numbers that are not in the window are ignored. Management of the retransmit queue must handle timing out multiple sets of messages sent at different times with each message given the same fixed amount of time to remain on the retransmit queue before being considered timed out. The retry value should be based on the configured timeout or the round trip delay. If the round trip delay is used, the retry value should be computed based on a minimum plus the round trip delay multiplied by two. The timeout value for a link is based on the number of retries for a link. Slower, perhaps less reliable, links may use a larger count.
- **Message packing and unpacking.** To better utilize the given bandwidth on a remote connection, it may be desirable to pack as much information as possible into each packet. This mechanism is only valuable if it is common for several small messages to be queued waiting for a message to be acknowledged or for the window to open up. To accomplish this, the following enhancements may be needed: As messages are removed from the pending message queue to be transmitted, if two or more messages will fit in a single message buffer, a large message will be allocated and all available messages that will fit are copied into the large buffer. As messages are received that contain more than one message as a result of being packed by the remote peer, new receive messages are allocated and data is copied from the packed message into individual message buffers to be processed. All messages contained in a single message will be acknowledged by a single acknowledgment of the last sequence number on the packed messages.
- **Other optimizations.** The extra middleware layer enables many kinds of optimizations. If several applications request the same data, we would get multiple requests for the same data sent over the wireless link. For remote network connections using satellites, modems or other slow or delayed transmission media, we could collect up the runtime data on the remote side of the network over the satellite (or other slow connection) and distribute it out to the other side. This reduces the amount of message traffic sent over the slow network. The remote applications would then get their runtime data from the local middleware instead of requesting it over the satellite link.

Finally the middleware should handle link integrity and security. Since we are allowing remote connections that may be connected via Satellite or Microwave type connections, we will need to provide security measures to prevent someone from being able to sniff

the network traffic and break into the system. The option of encrypting messages on the remote network connections should be provided.

3.3 Change to the control system

The middleware is a new component in the system. It has to be developed. This might be considered extra work for the control system.

It is impossible to run the wireline version of the control system without any change on this framework of wireless. Even if the middleware supports the same set of APIs that the control system invokes, the control system still needs to be enhanced to access the set of parameters in the middleware. Of course, if the control system and the middleware were developed together for the wireline version, switching to wireless could be much easier. The goal of the approach is to mitigate the difficulties introducing wireless to process control. For wireless network, the control system should configure and monitor the middleware parameters. The control system could rely on the middleware for wireless communication, but the more monitoring by the control system itself, the better control could be achieved.

3.4 Configure a system with wireless networks

Given a wireless control system, how to configure it makes a lot of difference. We have talked about how to develop a wireless control system. In this last subsection we discuss issues involved in deploying a wireless control system.

In Figure 3 all communications among the distributed nodes are wireless. Figure 5 depicts the same set of nodes, but only one wireless connection between a workstation and a controller. The workstations are connected with wireline. So are the controllers. This reduces the wireless exposure and should result in better data quality. Many real applications, such as the one we shall discuss in the next section, can only afford wireless communication between two dedicated remote nodes, which act as the gateway for all their local nodes.

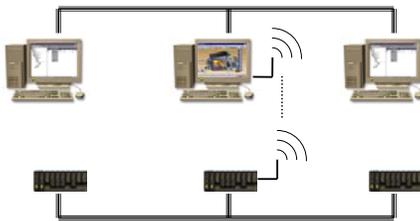


FIG 5 - DEDICATED WIRELESS NODES

Another important issue is the types of data transmitted across the wireless networks. As wireless is less reliable than wireline, a system needs to be configured such that only necessary data passes through wireless links. For example, configuration database could be placed where most of the download happens; historian data is better archived where the data source is; diagnostic data could be retrieved on demand; large quantity of non time critical data could be transmitted between production phases, etc.

4. THE OIL EXTRACTOR CASE

In this section we apply the idea in above section to analyze an oil extractor control system. Figure 6 [complements of Crudex, Inc, Sherman, TX, and Heritage Oil, Georgetown, TX] is an oil extractor and Figure 7 is the architecture.



FIG 6 - AN OIL EXTRACTOR

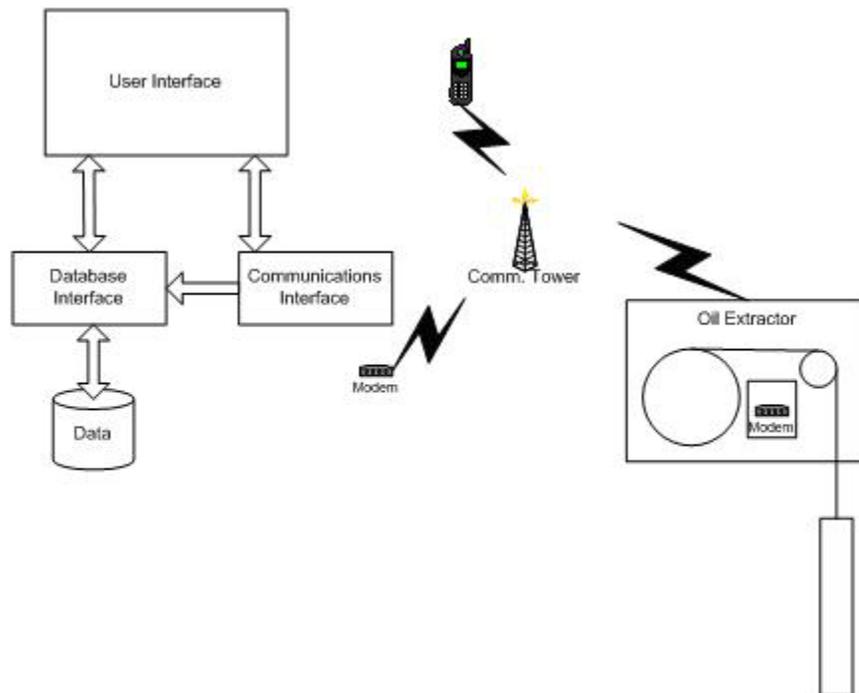


FIG 7 - THE OIL EXTRACTOR AND ITS COMMAND & CONTROL CENTER

The canister is lowered into a well, after some time oil fills up the canister, which is then pulled up to the assembly. The collected oil is then metered and pumped to a storage tank. The whole procedure is automated. Radio wireless communication will be used to communicate to a master communicator which then employs cellular communications to remotely communicate with operators.

Some of the system requirements interesting to this paper are:

- Allow users to remotely command the oil extractor unit.
- Allow users to locally operate the oil extractor with a hand held unit.
- Detect abnormal situation like leak and plugged line, salt-water, etc.
- Auto calibration, metering.
- Upload measurements and events to host through cellular links.
- Security to prevent an unauthorized user from tampering with metering information.

The wireless network used by the oil extractor provides performance statistics and cost structure. For example, we found the communication over Global System for Mobil Communication (GSM) using Short Message Service (SMS) messages is significantly effected between 7-9pm Monday-Friday. The pricing structure of Satellite tends to be for bandwidth; with cellular it's a combination of packets and bandwidth. Using cellular has many advantages. One advantage is the ability to send abnormal situation events to individuals or response centers.

The middleware dynamically adjusts to the network performance to send data with the best result. Some of the middleware's considerations are.

- The effective bandwidth of a GSM or Satellite network may be significantly reduced due to over-loading, noise, etc.
- In the case of Satellite communications noise is a significant issue, while in the case of a cellular network traffic is very real issue.
- Since the delay times with Satellite communications is significant, we could send many packets out and acknowledge in bulk.
- With cellular, at 2-5 cents per packet it makes sense to keep the packets as large as possible by combining several data packages into one.

The middleware API for the oil extractor is designed as follows:

- Configuring and Initializing the Modem port:
readIniFile()
bool initializeModemPort(HANDLE fd)
bool openComPort(LPCTSTR ComPort)

- Reading and writing to/from the Modem Port:
serialWrite(HANDLE fd, char *buffer)
bool checkIncoming(HANDLE fd, bool* bFound)
- Parsing SMS messages and storing them into the database:
void extractNumbers(int startPos, int endPos, CString& inputSMSMessage, CString& extractedString)
void parseCycleData(CString& inputSMSMessage, CString& commaDelimintedMessage)
void parseRecData(CString& inputSMSMessage, CString& commaDelimintedMessage)
int openFiles()
void closeFiles()
- Sending messages. The following methods are used to read SMS messages from the client and write them through the modem to the destination port. If the write succeeds an 'ACK' message is returned; otherwise a 'NAK' message is returned.
bool readBufferedMessages(HANDLE fd)
int sendMessage(HANDLE fd, char* phoneNumber, char* messageBuffer)
void instanceThread(LPVOID lpvParam)
int writeMsgThread(char *MyID)
void pPRFunc(void)

Table 1 is some of the logged runtime data and some production data.

TABLE 1 - OIL EXTRACTOR RUNTIME DATA

RUNTIME DATA		PRODUCTION DATA	
Current Depth (Qft)	386	Well Identification	15127754026
Min Depth Setting (ft)	1001	Cycle Trips	1349
Max Depth Setting (ft)	1003	DD-MMM	1-Feb
Current Depth Setting (Qft)	4012	HH:MM	8:36
Size of Canister (ounces)	542	Current Amt	4.09
Metering Pulse Count (Last Dump)	0	This Day Total	4.1
Total Production Today	0	Prev Day Total	0.0
User Entered Minimum Value	1001	This Month Total	4.0
Last Dump Time	1158	Prev Month Total	280.1
Last Dump Date	3012	Meter Counter	6
Calibration Time for Full Load	4		
Latched Depth Count (Qft)	4015		
Calibration Time for Empty Load	0		
Last Dump in Ounces	0		
Total Number of Cycles	326		
Increase in Time by Autotuner	600		

Note in our analysis wireless is considered as the replacement of wireline. By design, the two ends of the wireless connection are considered part of one single control system.

This furthers the concept of a single system. Normally a wireline control system is deployed in one physical plant. In our oil extractor system, the command and control center may be located thousands of miles away.

5. CONCLUSIONS

In this paper we looked at process control system and wireless technology. Without debating whether wireless is right for process control, we analyzed the issue faced by wireless and studied a possible middleware layer that separates the difficulties of wireless and process control. We consider process control applications where remote physical sites communicate via purchased service from commercial wireless networks. We discussed the design and execution of the middleware, the necessary change to the control system, and system configuration for wireless. We also presented an oil extractor implementation that makes heavy use of wireless communication. The paper only touched the surface of the wireless challenges for process control. We look forward to continued study on this topic.

REFERENCES

1. Moss, G., "Wireless ain't wireless!" *InTech Protocol*, April 2004.
2. Cutler, T., "Wireless solves legacy systems" *InTech*, January 2004.
3. Caro, D., "ZigBee short on power by design" *InTech*, May 2004.
4. Steven J. Vaughan-Nichols, "Wireless Middleware: Glue for the Mobile Infrastructure," *Computer Magazine*, Vol. 37, No. 5, pp.18-20, May 2004.
5. Steven J. Vaughan-Nichols, "Achieving Wireless Broadband with WiMax," *Computer Magazine*, Vol. 37, No. 6, pp.10-13, June 2004.
6. Hieb B., "Developing a Small Wireless Control Network", Master's Thesis, the University of Texas at Austin, December 2003.
7. <http://www.echelon.com>
8. <http://standards.ieee.org/wireless/>
9. <http://easydeltav.com/keytechnologies/index.asp>