RFID and Sensor Networks

From Sensor/Actuator to Business Application

Rolf Clauberg
IBM Research, Zurich Research Laboratory
8803 Rüschlikon, Switzerland

Introduction

In the past years a significant change for computer networks occurred: the rise of direct device-to-device or machine-to-machine communication within general computer networks. The most visible and publicly discussed example is the use of radio frequency identification (RFID) tags for improved supply chain management [1,2]. Several large retail chains as well as the US Department of Defense demand support of RFID technology from their suppliers within the next years. Delta Airlines together with the US Transportation Security Administration successfully performed a pilot project for using RFID tags for baggage handling at the airport of Jacksonville, Florida [3,4]. Another example, which reached some publicity, is the use of wireless sensor networks for environmental and military monitoring [5-7]. Started some years ago as the so-called “smart dust” project by the US Defense Advanced Research Projects Agency (DARPA), wireless sensor networks using large numbers of tiny sensor units are now used in first commercial applications for agriculture and vineyard monitoring [7]. Other examples of sensor networks are condition-based maintenance systems for early recognition of tank-refill or component-replacement requirements as well as real-time process-control systems for industrial automation. The latter systems existed already in the past, but were isolated from enterprise communication networks. Today’s demand for real-time status information about all business processes requires integration of these networks into the general enterprise computer network and, in connection with supply- or value-chain management, even access to the Internet through web services for crossing company boundaries [8]. This brief article will describe system requirements and a suitable architecture for scalable computer networks with integration of RFID and sensor networks. The main emphasis hereby is on the computer network architecture, the corresponding message-oriented middleware, and the integration of available standardized smart sensor devices and RFID readers into the network.

System Requirements and Network Architecture

A main issue for machine-to-machine communication is that the flow of information differs substantially from that in present-day computer networks. Instead of a large flow from central servers to clients at the edge of the network, the main data flow for RFID and sensor network systems is from many devices at the edge of the network towards a few central servers. This is especially true for condition-based maintenance-sensor networks and RFID networks at manufacturers and distribution centers. In both systems, sensors or RFID readers detect certain events and forward the corresponding information to some business application on a central server. The business application then responds to these inputs and arranges corresponding actions such as e.g.

replacing a fragile component before it fails or requesting the delivery of additional products before they are sold out. For both kind of networks this creates an imploding data stream from the edge to the center. To handle this kind of data streams for a large number of sensors or RFID readers, data or event filtering as well as data aggregation and abstraction are necessary at all suitable points from the edge towards the center of the network. Hence, certain parts of the business application are transferred from central servers to those at the edge of the network. To enable fast implementation of new applications, a flexible and automatic deployment of software on the edge servers is necessary. RFID systems at points of sale or access-control and sensor networks for real-time process control require actuators for automatic responses in addition to RFID readers and sensors. Depending on the acceptable response time, decisions on corresponding RFID reader or sensor data are made at central servers or directly at the closest edge server or sensor controller. If short response times are required, significant parts of the application must be running on the edge server or sensor/RFID controller, thereby shifting intelligence and responsibility from the network center to the network edge.

Other system requirements are remote device configuration, remote device software updates, system diagnostics (including sensor diagnostics), network reliability and security, and application access to data on a by-topic base instead of a per-device base. The requirements for remote configuration and software updates stem from the possibly very large number of edge devices and the fact that many of these devices will be installed far away from any information technology knowledgeable staff. Under these conditions, the total cost of ownership of RFID and sensor networks becomes unacceptable without remote system management. The requirement for real-time system diagnostics and overview is a simple consequence of the fact that these networks provide mission-critical inputs to business applications. Also, business applications usually need to know data according to specific topics, such as manufacturer information according to RFID electronic product code (EPC), temperature, pressure, or other parameters, and not according to which device measured the data. Hence, an intelligent network infrastructure should provide the corresponding data automatically in the way the applications need them. Depending on the applications, customers will require various degrees of network reliability and security. For most RFID and sensor networks, the important issue will be network reliability and data integrity, i.e. there should be no network breakdowns due to failing components or external denial-of-service attacks, and information received from the network should be reliable. Protection against failing components will require redundant designs of critical network elements, whereas data integrity and protection against denial-of-service attacks will require device and message authentication.

A suitable federated architecture for RFID and sensor network backbones is shown in Figure 1. Here, SU means a sensor unit or RFID reader with a wired or wireless connection to a gateway GW, and AU means an actuator unit for automatic response actions. The gateway is a sensor or RFID controller which can connect to the normal enterprise network. It will usually be based on a 32-bit microprocessor, but, depending on the application, the capabilities of the gateway may still be limited by power-consumption constraints, e.g. there may be no local storage capability at the gateway. The first possible point for data filtering, aggregation and abstraction is at the gateway, except for the case where a mesh network of sensor units is connected to the gateway and some kind of data aggregation and filtering is already done within the mesh network [9].

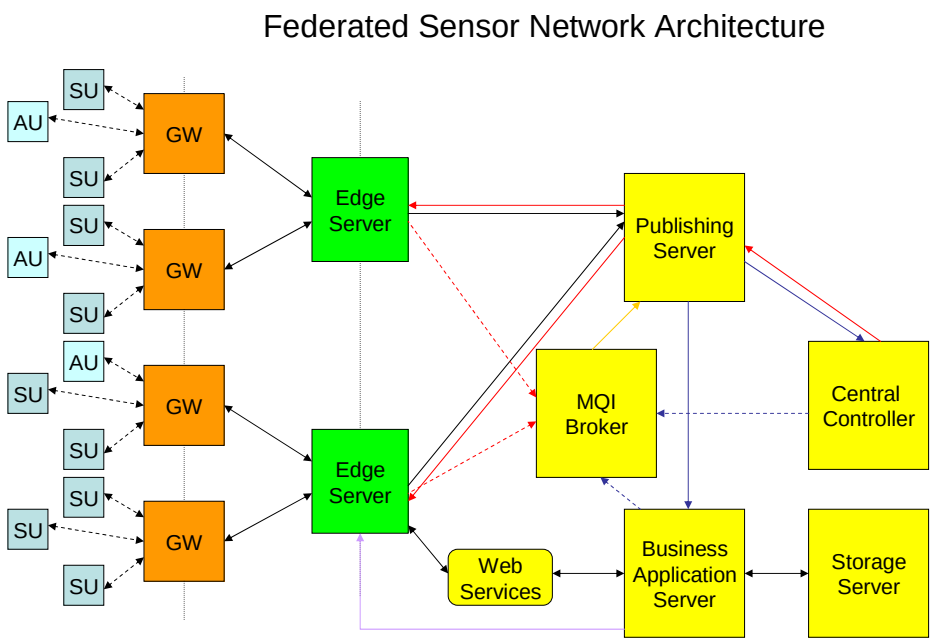

Figure 1

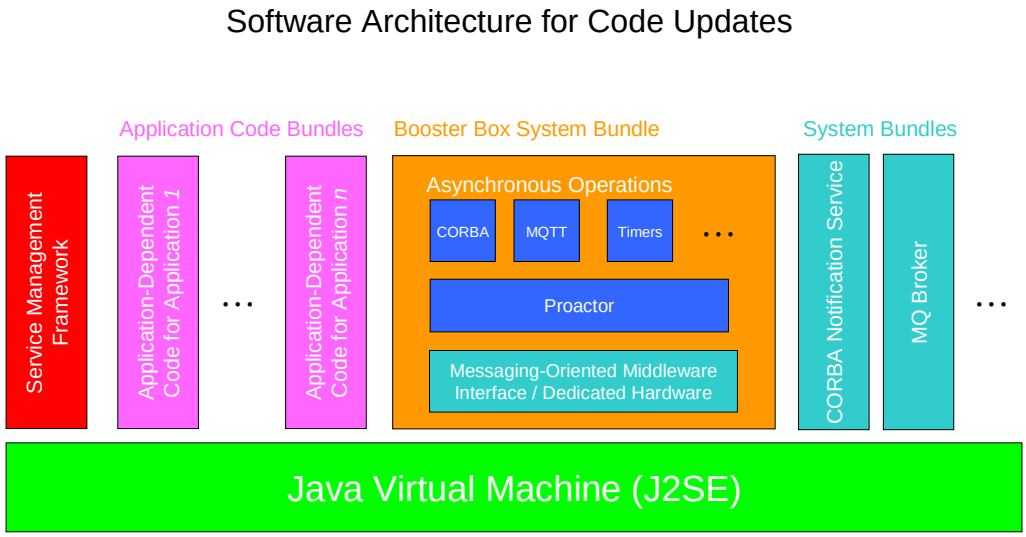

Figure 2

The next level of data aggregation and abstraction is performed at the edge servers. To enable flexible deployment of application codes from central servers to edge servers or even gateways, the software architecture of these devices uses IBM's Service Management Framework (SMF), which is based on the standards of the Open Services Gateway initiative (OSGi). Figure 2 shows a corresponding software architecture with a Java virtual machine as basis. SMF enables receiving of software code bundles from a central server and thereby updating application code and configuration information. This software architecture is suitable for edge servers, high-performance gateways and RFID controllers. Some gateways for remote sensor network applications with serious power consumption and related memory constraints may not be able to support the complete software architecture of Figure 2. However, all gateways should still be able to support a slimmed-down version based on the J9 Java virtual machine for embedded systems plus the Message Queue Telemetry Transport (MQTT) protocol for reliable message transport between gateways, edge servers, and central servers. Publish and subscribe functionality works by pushing data with MQTT to specific central servers for publishing data in specific formats to specific applications and servers based on subscription lists. These subscription lists are created by a central MQ integration (MQI) broker (see Figure 1) to which all applications send subscription requests defining what data they want to receive and in which format. The gateways on the other hand can subscribe to configuration updates concerning all sensor units or RFID readers connected to them. This enables efficient remote device configuration.
Access to web services for supply-chain optimization beyond single companies or EPC information access is possible from edge and central servers.
Device and message authentication between queue managers is an integral part of IBM's MQ middleware, and creates the basis for end-to-end system security in RFID and sensor networks based on the architecture shown in Figure 1.
For clarification it should be mentioned that the network architecture of Figure 1 can be extended to a very large number of gateways and sensor/RFID reader units or even collapsed into a single physical device. Especially combinations of sensor/RFID units with gateways/RFID controllers or gateways with edge servers into single devices will be common for specific applications.

Smart Sensor Devices and their Integration into the Network

Looking at the different RFID reader and sensor devices, there are three main classes of devices. The first class is that of wired devices with no serious power constraints. These devices will usually include physical sensors or RFID readers, plus a 32-bit microprocessor for local data processing and a network connection. They are a combination of sensor unit and gateway or RFID reader and RFID controller, respectively. Main applications are fixed installed RFID readers or wired sensor units for real-time process control in industrial automation. The second class is that of PDA-like battery-driven mobile devices as RFID readers or smart sensor units. They are nearly identical to the wired devices but use wireless connections to the backbone network. Their main applications are RFID-based inventory control, personal smart sensor systems for e.g. medical control, and remote condition-based maintenance systems that are switched on just once or twice per day. Battery lifetimes for the RFID and personal smart sensor systems will be comparable to those of mobile phones. Acceptable battery lifetimes of about 10,000 hours for the condition-based

maintenance systems are achieved through extremely low duty cycles for these systems. The third class is that of battery-driven very-low-power, low-performance smart sensor units. These devices include physical sensors, plus a low-power (usually 8-bit) microcontroller, very little memory, and a low-power, small-range wireless radio connection. Battery lifetimes of 10,000 to 15,000 hours are achieved with duty cycles of about 0.01%, i.e. these systems are in sleep mode about 99.9% of their time. These units need a gateway to connect to a usual computer network. At least for now there seems to be no counterpart in RFID systems for this class of devices.
Devices of the first and second class are easily integrated into standard computer networks because they can use standard embedded software solutions such as e.g. the one shown in Figure 3. The software architecture of Figure 3 supports MQ-based connection to the backbone network as well as access to the Internet through web services. It is therefore completely compatible with the architecture given in Figure 1. The real-time signal-processing application runs directly on embedded Linux instead of on the Java virtual machine to enable very fast feedback.

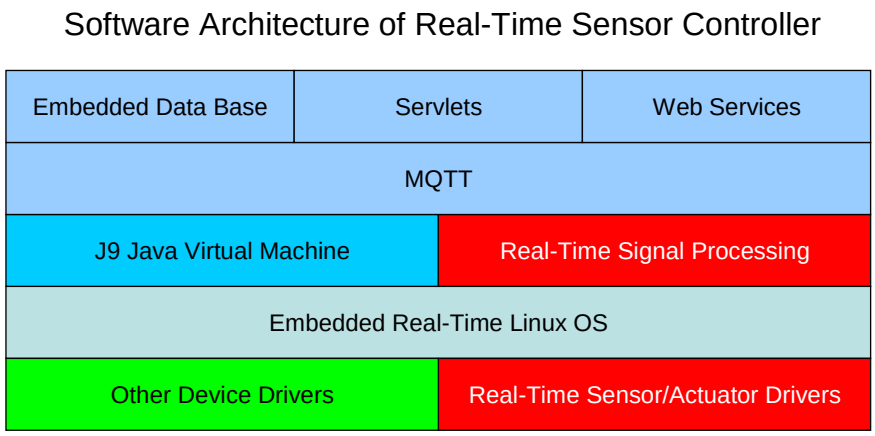

Figure 3

Devices of the third class pose more difficulties for integration into a complete system solution with end-to-end security and service guaranties because they are based on highly application-specific software usually running on 8-bit microcontrollers. Examples of operating systems are TinyOS from the University of California at Berkeley [10] and the IEEE 802.15.4 [11] protocol stack extended by the ZigBee industry alliance recommendations [12]. The complete protocol stack for TinyOS is about 3.5 KB [10] and that of the IEEE 802.15.4/ZigBee standard about 4 KB for simple sensor network nodes. Nevertheless, at least end-to-end system security should be feasible as the IEEE 802.15.4 standard supports symmetric key encryption and

authentication. Integration of these devices into the network architecture of Figure 1 is currently one of the activities of the sensor network group in Rüschlikon.

Summary

System requirements for end-to-end RFID and sensor network solutions were discussed, and a suitable network architecture for integrating RFID and sensor systems into computer networks was presented.

Acknowledgment:

I want to thank my colleagues P. Scotton and C.C. Clauss for input and discussions to the topics described above.

References:

1. M. Levinson, “The RFID Imperative”, CIO Magazine, Dec. 1, 2003
2. K.J. Delaney, “Technology for Tracking Goods Gets Boost from Microsoft, IBM”, The Wall Street Journal, Jan. 26, 2004
3. B. Brewin, “Delta has Success in RFID Baggage Tag Test”, Computerworld, Dec. 18, 2003
4. B.J. Feder, “Delta to Invest in Radio Tags for Luggage at Airports”, The New York Times, July 1, 2004
5. “In Dust we Trust”, The Economist, Jun. 12, 2004
6. H. Green, “Sensor Revolution: Bugging the World; Soon, Sensor Networks will Track Everything from Weather to Inventory”, Business Week, Aug. 25, 2003
7. B.J. Feder, “Psst. This Is Your Sensor. Your Grapes Are Thirsty”, The New York Times, July 26, 2004.
8. Peter Fingar and Ronald Aronica, *The Death of 'e' and the Birth of the Real New Economy: Business Models, Technologies and Strategies for the 21st Century*, Megan-Kiffer Press, Tampa, FL, USA, 2001.
9. M.D. Yarvis, W.S. Conner, L. Krishnamurty, J. Chhabra, B. Elliott, and A. Mainwaring, “Real-World Experiences with an Interactive Ad Hoc Sensor Network”, in *Proceedings of the International Conference on Parallel Processing Workshops (ICPPW’02)*, Aug. 18-21, 2002, pp. 143-151.
10. J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, “System Architecture Directions for Networked Sensors”, in *Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-IX)*, Cambridge, MA, USA, Nov. 12-15, 2000, published in Operating Systems Review vol. 34, no. 5 , Dec. 2000, pp. 93-104.
11. IEEE Standard 802.15.4: “Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)”, IEEE 2003.
12. ZigBee industry alliance : www.zigbee.org