Dependability-driven Embedded Systems Networking

Miroslav Sveda and Radimir Vrba
Brno University of Technology
Bozetechova 2, CZ-612 66 Brno, Czech Republic
Phone: +420 54114 1288; Fax: +420 54114 1270; E-Mail: sveda@fit.vutbr.cz

Abstract
This paper deals with embedded system networking support, which is aimed at applications based on distributed components interconnected by wired Internet and wireless sensor networks. It presents a safety and security-driven approach to embedded system networking that offers reusable patterns for designs aiming at various application domains. After introducing terminology dealing with functionality and dependability, the article copes with industrial, sensor network and Internet based architectures. It discusses an integrated networking framework stemming from the IEEE 1451.1 smart transducer interface standard, which is an object-based networking model supporting client-server and publish-subscribe communication patterns in group messaging, and from the IP multicast communication, mediating safe and secure access to wireless sensor networks through Internet. The case study demonstrates how clients can access groups of wireless smart pressure and temperature sensors and valves, which monitor and control gas pipelines effectively through Internet using developed networking architecture that respects prescribed requirements for application dependent safety and security.

1 Introduction

Internet technologies complemented by wireless access networks are rapidly becoming the preferred choice for building next generation distributed measurement and control systems. The framework, which can provide for current trends, stems from the IEEE 1451.1 standard specifying smart transducer interface architecture that enables to unify interconnecting smart sensors and sensor-based embedded systems with various wireless networks, and also their direct coupling to recent Ethernet-based Intranets. Supporting techniques included in the framework, namely publish-subscribe messaging and IP multicasting respecting security requirements, offer scalable and traffic-saving solution important from viewpoint of the contemporary Internet. The schemes discussed can properly interplay with each other and can supply suitable support for design of networked, sensor-based embedded system applications.

2 Functionality and Dependability

The design of current networked embedded system applications should consider namely functionality and dependability measures [6]. Functionality means services delivery in the form and time fitting requirement specifications, where the service specification is an agreed description of the expected service. Functionality properties should be realized efficiently and cost-effectively, so reachable performance and maintainability belong to the checked properties. Dependability is that property of a system that allows reliance to be justifiably placed on the service it delivers. A failure occurs when the delivered service deviates from the specified service. Dependability measures consist namely of reliability, availability, security, safety and survivability. Availability is the ability to deliver service under given conditions for a given time, which means namely elimination of denial-of-service vulnerabilities. Security is the ability to deliver service under given conditions without unauthorized disclosure or alteration of sensitive information. It includes privacy as assurances about disclosure and authenticity of senders and recipients. Security attributes add requirements to detect and avoid intentional faults. Safety is the ability to deliver service under given conditions without catastrophic affects. Safety attributes add requirements to detect and avoid catastrophic failures.

A failure occurs when the delivered service deviates from the specified service. The failure occurred because the system was erroneous: an error is that part of the system state which is liable to lead to failure. The cause of an error is a fault. Failures can be classified according to...
consequences upon the environment of the system. While for benign failures the consequences are of the same order of magnitude (e.g. cost) as those of the service delivered in the absence of failure, for malign or catastrophic failures the consequences are not comparable.

A fail-safe system attempts to limit the amount of damage caused by a failure [5]. No attempt is made to satisfy the functional specifications except where necessary to ensure safety. A mishap is an unplanned event (e.g. failure or deliberate violation of maintenance procedures) or series of events that results in damage to or loss of property or equipment. A hazard is a set of conditions within a state from which there is a path to a mishap.

A fail-stop system never performs an erroneous state transformation due to a fault [8]. Instead, the system halts and its state is irretrievably lost. The fail stop model, originally developed for theoretical purposes, appears as a simple and useful conception supporting the implementation of some kinds of fail-safe systems. Since any real solution can only approximate the fail-stop behavior and, moreover, the halted system offers no services for its environment, some fault-avoidance techniques must support all such implementations.

3 Embedded System Networking

Embedded system networking concepts stem from hierarchically interconnected networks of various kinds, principally Internet, local area wired and wireless networks, and wireless sensor networks. The logical hierarchy is usually based on tiered architectures [11] that bring cost-effectiveness, longevity and scalability, and allow adapting straightforwardly to various application requirements. The last subsection provides a brief outline of the IEEE 1451 communication architecture aimed at smart sensors and fitting embedded system networks of which smart sensor networks constitute essential subset.

3.1 Wired and wireless networks

Actually, networking appears a basic feature of promising embedded systems architectures. Internet access to individual components of distributed embedded systems can be based on both wired and wireless LAN technologies, predominantly on IEEE 802.3 and related Ethernet standards, and on IEEE 802.11b WiFi and associated wireless LAN protocols. Embedded systems and their components can be attached directly to Ethernet with TCP/IP protocol stack, but also indirectly or exclusively through various wired Fieldbuses or wireless technologies such as IEEE 802.11b WiFi, IEEE 802.15.1 Bluetooth, and IEEE 802.15.4 with related ZigBee. Sensor networks bring an important pattern with single base station connected to a wired network on one side and wirelessly to smart sensors on the other side. When sensors are clustered, the base station communicates to cluster heads and through them to individual sensors. Moreover, the applications can use also ad-hoc wireless network architectures that enable to extend wireless part of the system network over physical limits and bring new dimension to fulfill application requirements.

Application dedicated architectures can be developed based on IEEE 1451.0, 1, .2, .3, .4 and .5 standards [9, 12]. In this case, application layer can benefit from object-oriented abstractions introduced by IEEE 1451.1 Network Capable Application Processor (NCAP) model [2], see the following section.

3.2 IEEE 1451 architecture

The IEEE 1451, see Figure 1., consists of the family of standards for a networked smart transducer interface that include namely (i) a smart transducer software architecture, 1451.1 [2], targeting software-based, network independent, transducer applications, and (ii) a standard digital interface and communication protocol, 1451.2 [4], for accessing the transducer or the group of transducers via a microprocessor modeled by the 1451.1. The next three standard proposals extend the original hard-wired parallel interface 1451.2 to serial multi-drop 1451.3, mixed-mode (i.e. both digital and analog) 1451.4, and wireless 1451.5 interfaces. The document 1451.0 complements the above standard set defining the structure of Transducer Electronic Data Sheets (TEDS), and associations between 1451.1 on one side and 1451.2/3/4/5 on the other side with message exchange protocols and command set for transducers.

![Figure 1: IEEE 1451 configuration example](image-url)
4 Selected Techniques

The following subsections introduce techniques important for embedded system networks in industrial applications. Publish-subscribe pattern provides simple approach to data distribution in application layer group communication. Next subsection deals with multicast communication, namely with network layer multicast in the Internet protocol stack. The third subsection discusses security-related techniques in multicast context.

4.1 Remote procedure call and publish-subscribe

The majority of communication protocols provide the client-server style of communication. In case of sensor communications, the client-server pattern covers both the configuration of transducers and initialization actions. If the client wants to call some function on server side, it uses a command execute. On server side, this request is decoded and used by the function perform. That function evaluates the requested function with the given arguments and, after that, it returns the resulting values to the client.

The client-server pattern corresponds to remote procedure call (RPC), which is the remote invocation of operations in a distributed context [1]. To be more precise, the RPC interaction considered in this paper provides a synchronous client-server communication, i.e. the client is waiting for the server’s response before completion RPC actions related to the current call. Evidently, the client-server communication style relates to point-to-point message passing called as unicast.

The subscriber-publisher style of communication [1] can provide the efficient distribution of measured data. All clients, wishing to receive messages from a transducer, register themselves to the group of its subscribers using the function subscribe. After that, when this transducer generates a message using the function publish, this message is effectively delivered to all members of its subscribing group. Transducers in the role of publishers have also the ability to advertise the nature of their future events through an advertise function.

The interaction publish-subscribe relates to point-to-multipoint or multipoint-to-multipoint asynchronous message passing. Of course, it can be implemented using multiple unicast communication transactions. On the other hand, to satisfy the requirement of efficiency, it is necessary to utilize elaborate multicast techniques encompassing multicast addressing and, namely, multicast routing. The basic principles of the network layer multicast in the Internet environment are discussed in the following section.

4.2 Multicast

Traditional network computing paradigm involves communication between two network nodes. However, emerging Internet applications require simultaneous group communication based on multipoint configuration.
propped by multicast IP, which saves bandwidth by forcing the network to replicate packets only when necessary. Multicast improves the efficiency of multipoint data distribution by building distribution tree from a sender to a set of receivers [7]. The Internet multicast concept converts the mesh Internet or Intranet into a shared resource for senders to send messages to multiple participants or groups. To make this group communication flexible, both for senders and for the routing functions, the system should be independent of the particular recipients.

The functions that provide the Standard Internet Multicast Service can be separated into host and network components. The interface between these components is provided by IP multicast addressing and Internet Group Management Protocol (IGMP) group membership functions, as well as standard IP packet transmission and reception. The network functions are principally concerned with multicast routing, while host functions can also include higher-layer tasks such as the addition of reliability facilities in a transport-layer protocol.

IP multicasting is the transmission of an IP datagram to a host group, a set of hosts identified by the single IP destination address of class D. Multicast groups are maintained by IGMP (IETF RFC 1112, RFC 2236). Multicast routing considers multicasting routers equipped with multicasting routing protocols such as DVMRP (RFC 1075), MOSPF (RFC 1584), CBT (RFC 2189), PIM-DM (RFC 2117), PIM-SM (RFC 2362), or MBGP (RFC 2283). For Ethernet-based Intranets, the Address Resolution Protocol provides the last-hop routing by mapping class D addresses on multicast Ethernet addresses.

4.3 Securing multicast

Multicast security should provide assurances about disclosure (privacy) and authenticity of sender/recipient. The key exchange protocols used between unicast hosts do not scale well to groups. Re-keying is required to maintain confidentiality as group membership changes. Evidently, any efficient authentication transforms used between two unicast hosts cannot protect traffic between mutually distrustful members of a group. The IETF Multicast Security and IRTF Group Security working groups developed a building block approach to solve the problem. The blocks encompass data security transforms, group key management and security association, and group policy management. Any application may use different blocks together to create a protocol that meets its specific requirements.

A data security transforms block provides confidentiality and authentication services for data being transported among group members. Confidentiality can be provided using standard encryption algorithms. Authentication appears more difficult, because the secret used to authenticate the traffic must be shared between all sending and receiving parties. Public-key signatures would solve this problem, but they are an order of magnitude slower than symmetric authentication algorithms and, hence, especially unsuitable for real-time traffic. Instead, blocks such as the Timed Efficient Stream Loss-tolerant Authentication Protocol (TESLA) trade off small amounts of functionality to retain the efficiency benefits of symmetric algorithms. TESLA senders use a hash chain of keys to sign data. They release each key in the chain a short interval after the data the key has signed. As long as other group members received the data during that interval, they can be confident that the signature was made by the sender. If keys are lost during transmission, receivers can re-compute any key earlier in the sequence simply by repeatedly applying the hash function used to any later key received. Finally, they can be sure that keys are coming from the sender because the first key in the sequence is digitally signed, while only the sender can know the later keys in the sequence.

The Group Key Management architecture provides a unified model for key management blocks. A central Group Controller/Key Server provides Traffic Encrypting Keys or Key Encrypting Keys to new group members after authenticating them with a unicast protocol. Three key management building blocks include Group Domain of Interpretation that builds on the Internet Security Association Key Management Protocol to allow the creation and management of security associations for network or application layer protocols, Group Secure Association Key Management Protocol, and GSAKMP-Light profile.

The final building block defines policies such as which roles various entities may play in the group; who may hold group information such as cryptographic keys; the cryptographic algorithms used to protect group data; and proof that the creator of a given policy is authorized to do so.

Multicast traffic in sensor networks should profit from special application architectures. The design can simplify security arrangements and, concurrently, enables to prop robustness and reliability.

5 Case Study

This section describes a case study that demonstrates utilization of the introduced framework. Application deals with pressure and temperature measurement and safety and security management along gas pipes. The related
implementation stems from the IEEE 1451.1 model with Internet and the IEEE 1451.5 wireless communication based on ZigBee running over the IEEE 802.15.4.

5.1 Functionality and dependability issues

The application architecture comprises several groups of wireless pressure and temperature sensors with safety valve controllers as base stations connected to wired intranets that dedicated clients can access effectively through Internet, see Figure 2. The WWW server supports each sensor group by an active web page with Java applets that, after downloading, provide clients with transparent and efficient access to pressure and temperature measurement services through controllers. Controllers provide clients not only with secure access to measurement services over systems of gas pipes, but also communicate to each other and cooperate so that the system can resolve safety and security-critical situations by shutting off some of the valves.

Each wireless sensor group is supported by its controller providing Internet-based clients with secure and efficient access to application-related services over the associated part of gas pipes. In this case, clients communicate to controllers using a messaging protocol based on client-server and subscriber-publisher patterns employing 1451.1 Network Block functions. A typical configuration includes a set of sensors generating pressure and temperature values for the related controller that computes profiles and checks limits for users of those or derived values. When a limit is reached, the safety procedure, which is derived from the fail-stop model discussed in section 2, closes valves in charge depending on safety service specifications.

Security configurations in this case can follow the tiered architecture discussed above. To keep the system maintenance simple, all wireless communication uses standard ZigBee hop-by-hop encryption based on single network-wide key because separate pressure and/or temperature values, which can be even-dropped, appear useless without the overall context.

Security in frame of Intranet subnets stems from current virtual private network concepts. The discussed application utilizes ciphered channels based on tunneling between a client and a group of safety valve controllers. The tunnels are created with the support of associated authentications of each client.

5.2. Structure of the 1451.1 implementation

In the transducer’s 1451.1 object model, basic Network Block functions initialize communication between a client, which passed an authentication procedure, and the controller identified by a unique unicast IP address. The client-server style communication, which in this application covers both the configurations of controllers and initialization actions, is provided by two basic Network Block functions: execute and perform. The standard defines a unique ID for every function and data item of each class. If the client wants to call some function on server side, it uses command execute with appropriate parameters. On server side, this request is decoded and used by the function perform. That function evaluates the requested function with the given arguments and, in addition, it returns the resulting values to the client. Those data are delivered by requested variables in execute arguments.

Each controller communicates wirelessly with its sensors through 1451.5 interfaces by proper communication protocol. In the discussed case the proposed P1451.5-ZigBee, which means ZigBee over IEEE 802.15.4, protocol was selected because it fits application requirements, namely those dealing with power consumption, response timing, and management [3, 12]. The subscriber-publisher style of communication, which in this application covers primarily distribution of measured data, but

![Figure 2: Network configuration example](image-url)
also distribution of group configuration commands, employs IP multicasting. All regular clients wishing to receive messages from a controller, which is joined with an IP multicast address of class D, register themselves to this group using IGMP. After that, when this controller generates a message by Block function publish, this message is delivered to all members of this class D group, without unnecessary replications.

The 1451.1 network model provides an application interaction mechanism supporting both client-server and publisher-subscriber paradigms for event and message generation and distribution. Controllers play the role of clients or subscribers for the wireless part of the system network, and the role of servers or publishers for the wired part. Moreover, they compute temperature and pressure profiles, check the limit values and handle the safety valves.

6 Conclusions

This contribution deals with industrial, embedded systems networking support that is aimed at applications with distributed components interconnected by wired Internet compatible intranets and wireless sensor networks. The paper presents an approach to embedded system networking that offers a reusable design pattern for the considered class of Internet-based applications. It brings an integrated networking framework stemming from the IEEE 1451.1 smart transducer interface standard and from the IP multicast communication, which mediates safe and secure access to sensors and actuators in sensor networks through Internet. The case study demonstrates how clients on Internet can access groups of wireless smart pressure and temperature sensors effectively respecting prescribed application requirements including safety and security.

Original contribution of this paper consists in explicit formulation of the integrated networking framework focused on embedded systems. While there are also other papers presenting comparable approach to networking applications in that domain (e.g. [10]) the authors are not aware of similar generalization offering such application framework.

Acknowledgements

The research has been supported by the Czech Ministry of Education in the frame of Research Intention MSM 0021630528: Security-Oriented Research in Information Technology, and by the Grant Agency of the Czech Republic through the grants GACR 102/05/0723: A Framework for Formal Specifications and Prototyping of Information System’s Network Applications and GACR 102/05/0467: Architectures of Embedded Systems Networks.

References