The ITB Journal

Volume 4 | Issue 2 Article 8

2003

Embedded Systems in a Mobile Distributed IP Network

Conor Gildea

Declan Barber

Embedded Systems in a Mobile Distributed IP Network

Conor Gildea and Declan Barber

Institute of Technology Blanchardstown

conor.gildea@itb.ie declan.barber@itb.ie

Abstract

This paper describes an approach to the internetworking of mobile IP sensor nodes in a converged network environment. This preliminary investigation of sensory network models is driven by a joint applied research project between ITB and Teagasc (Grange) which seeks to establish the feasibility of the real-time remote monitoring of animal welfare while in transit between Ireland, Europe and the Middle East. The paper examines traditional system architectures, messaging paradigms and protocols with a view to establishing how the trend towards convergence in telecommunications and the emergence of new Internet Protocols could support the creation of new modes of operation for sensory networks.

Introduction

Traditional sensor network have applied centralized static models to intercommunication between relatively unintelligent sensor nodes and intelligent management stations. Recent trends are making it increasingly feasible to move away from a centralized model to a more distributed one to a point where a mobile sensor network could conceivably be modeled as ad hoc network of autonomous nodes. The impact of Moore's Law has led to the concentration of greater processing power, memory and storage (and consequently increased levels of intelligence) on small devices. Traditional switched and mobile networks are now converging around the TCP/IP model and Internet protocols are providing a means of rapidly deploying new applications and services across this converged space. IP enables the internetworking of disparate network nodes by providing a standardized addressing scheme and path determination. Higher layer protocols can provide reliability, signaling and quality of service support. One potential outcome of these trends is the repartitioning of capabilities and responsibilities within a sensory network to a more distributed model as intelligence spreads outwards from the centre to the edge of the network. Sensory nodes can now be independent computing platforms capable of peer-to-peer communication and of interacting with interim network nodes in order to provide previously unavailable services. Non-heterogenous nodes could inter-operate across a global inter-network and interact as single nodes or as logical groups. The Java language provides a platform independent application development environment and network operating system for code mobility. It increasingly provides frameworks and APIs for internet protocol implementation and telecommunications and internetworking support. Although limitations still exist in the intelligent services supported across the Internet, new protocols are services are emerging to address these shortcomings. This paper seeks to analyse the potential impact of these developments on sensor network embedded systems architectures, messaging paradigms (we will use the term messaging to include messages for carrying either data or control signals), modes of operation and supporting protocols. We will then briefly apply the analysis to the creation of a simple messaging service which could be used for sensory network communication.

Embedded Systems

An embedded system is a device that contains programmed logic on a chip that is used to control one or more functions of the device. A real-time embedded system provides deterministic performance, often in a mission critical environment. A sensor is often a real-time embedded device which is used to gather data such as the ambient temperature or humidity, in either a static or mobile environment, and report to a central management device. Mobile forms of such devices will increasingly operate in a pervasive computing environment using wireless transport technologies with IP connectivity for communication.

Evolving Architectures

Initially embedded systems were standalone devices with relatively simple interfaces and limited computationally ability. The implementations were essentially proprietary in nature although they did make use of open standards at the Physical Layer (e.g. physical interfaces and signaling) of the OSI model. The advent of wire-based networking made it possible to interconnect these devices in a static deployment, while the need to monitor and control the devices gave rise to centralized management. The proprietary approach persisted. Most sensory systems today are still based on this static centralized proprietary architecture. A server or equivalent central station allows embedded devices to report based on time intervals or triggered events or else it routinely polls the devices for control or monitoring purposes. With the limited processing and memory of the sensor node, data is sent back to the more powerful and intelligent central server for storage and further processing. As edge nodes become more powerful, however, it has become possible to re-partition the distribution of intelligence and processing within the sensory system. If we consider this within a general trend towards IP convergence in the telecommunications and internetworking world, edge devices with partial TCP/IP stack support and modest processing power can provide a more distributed architecture and embedded devices with the full TCP/IP stack and more comprehensive computing power can become independent autonomous nodes in an ad hoc network. This ability to repartition sensory network architecture more flexibly than ever before based on open standards and internet protocols provides a basis for creating more flexible sensor network designs with better functionality, scalability, availability, performance, manageability and security characteristics.

Messaging Paradigms

There are two fundamental paradigms for the messaging between entities in traditional sensor network systems: Push or Pull.

In networked computing, the pull model is based on the generalized request/response paradigm (called data polling or simply polling) where the client sends a request to the server, and then the server answers, either synchronously or asynchronously. This is functionally equivalent to the client "pulling" the data off the server. In traditional sensory networks, it is typically the central or management node that sends a request to the remote sensing node which then replies. Figure 1 illustrates a pull model architecture in which a central node communicates with edge nodes using the Pull paradigm over an event channel.

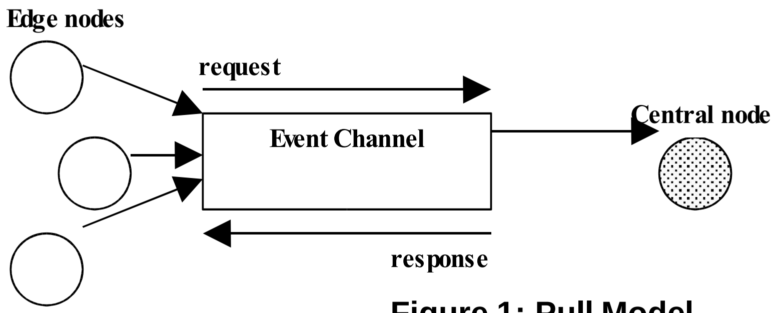

Figure 1: Pull Model

The alternative Push paradigm is based on the publish/distribute model in which the remote nodes push data into the central node at pre-determined intervals or on triggering events. In this model, agents first advertise what data they support, and what notifications they can generate; the administrator then subscribes the manager to the data of interest and specifies how often the manager should receive this data. It is then the responsibility of the agent to “push” data to the manager, either on a regular basis via a scheduler or asynchronously through message notifications. Figure 2 illustrates a push model architecture in which edge nodes communicate with a central node over an event channel.

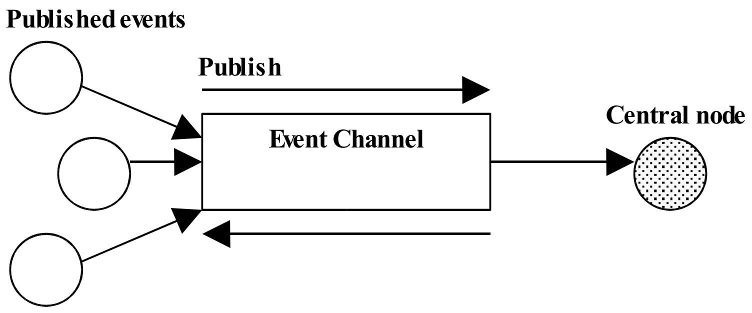

Figure 2: Push model of event transfer

The Pull paradigm has the advantage of simpler, centralized monitoring and management. It requires minimal intelligence in the edge node and is therefore conducive to a centralized architecture. It works well for regular monitoring or troubleshooting but is not appropriate for dynamic response functionality. The Push paradigm requires at least some independent decision-making at the network edge and is therefore consistent with a more distributed architecture. It is useful for generating reports triggered by events but this does not preclude it from being used for regular monitoring. It can improve bandwidth utilization by connecting only when there is something worth reporting. Where nodes become heterogeneous both paradigms require an initial registration where the node is established as a valid device and where capabilities are established.

Hybrid Push/Pull Models

It is possible to combine both modes of operation in a hybrid which seeks to combine the benefits of each paradigm. As the edge nodes and the central node can be completely decoupled from the event channel, push and pull models can be mixed in a single system. For example, edge nodes can connect to an event channel using the push model, while central or management nodes connect using the pull model. It is also possible to use different paradigms on a messaging function basis. System monitoring functions could be executed using pull messages while data retrieval functions (parameters readings, alarms, etc) could use push messages.

Same Paradigms – New Modes of Operation

Although these fundamental paradigms remain in use, the repartitioning of intelligence within the network and the use of IP make it possible to consider modes of operation which use these paradigms in other ways than those strictly based solely on a direct event channel between an edge node and a central node. One possibility is the use of an interim repository node to store edge node messages which in turn is either polled or pushes the message to the central node. This mode of operation seems attractive for mobile sensor networks in which the edge nodes may not always be able to link directly to a central node. Another mode of operation could be direct peer-to-peer messaging between edge nodes using either the push or pull paradigm. This mode of operation between autonomous or semi-autonomous nodes now moves beyond decision-making at the edge node to collective decision-making by a logical group of nodes. This type of decision making could support not just core networking functions such as availability and path determination, but also application layer decision-making e.g. a logical group of mobile sensors could transfer their collective linkage to a new central or interim node based on the group's changing geographic centre of gravity and replicate their data accordingly. With such expanded modes of operation, it is feasible to design systems not only where central nodes invoke objects on edge nodes (and vice versa) bit where any type of nodes can invoke objects on any other type of node. Other modes of operation are possible and will be increasingly relevant with mobile devices that are capable of 'multi-band' transmission, are deployed in an increasingly pervasive computing environment and interface to different inertial reference frameworks. An example of such a deployment is a sensor platform that combines its core parametric sensing with GPS positional sensing, Wi-Fi networking to nearby nodes and GSM cellular sensing and communications capabilities for more remote nodes.

Internet Protocols & Embedded devices

While it may be possible to design these new architectures and modes of operation, how feasible would such designs be to implement in the NGN? If we assume that IP will provide best effort delivery in the NGN, we can focus our examination of implementation feasibility on higher layer (of the TCP/IP model) issues and largely ignore the lower data link and physical layer issues. This does not preclude the importance of key sensor network design issues such as interfacing and bandwidths but assumes that the convergence trend and constant improvements in bandwidth availability will overcome any such obstacles to implementation.

Recent innovation in embedded device technology has resulted in the placement of a thinned-out and even a full TCP/IP stack on embedded devices. This means that networked embedded devices are now in a position to leverage the power of the internet using well established protocols such as HTTP and FTP and developing protocols such as SIP (Session Initiation Protocol) for signaling.

Message Delivery

In the Internet protocol suite, IP is used as the network addressing protocol and the routed information by which paths can be determined between nodes. IP provides a best-effort connectionless delivery service. Reliability is left to the Transport layer protocol. IP delivers packets based on unicast (one-to-one), multi-cast (one-to-many) and broadcast (one-to-all) addressing.

Internet Application Protocols

The main existing Internet Application Protocols and their underlying messaging paradigms are overviewed in Table 1 below. As can be seen from the table, most of these applications are based on using either the Push or Pull paradigm. Messaging applications based on these protocols can operate across a network of TCP/IP enabled embedded devices.

Application Protocols	**Function**	**Mode of Operation**	**Architecture**	**Delivery Mechanism**	**Reliability**
HTTP	Data transfer	Pull	Client-Server	Unicast	Yes
FTP	Data transfer		Client-Server	Unicast	Yes
Upload		Push		Unicast	Yes
Download		Pull		Unicast	Yes
SMTP	Data transfer	Push	Client-Server	Unicast	No
IMAP/PO P	Data Transfer	Pull	Client Server	Unicast	No
SNMP	Management		Client-Server	Unicast	Yes
SIP (and SDP)	Signaling	Push	Client-Server Peer-to-peer	Unicast Sig/Media Multicast Media Support	Yes

Table 1: Internet Application Protocols

Two preliminary observations can be made at this point:

a) There is a lack of application protocols which can leverage underlying IP delivery services to provide the more advanced modes of operation
b) Within the existing set of protocols, the SIP/SDP combination appears to offer the most flexibility for supporting advanced modes of operation for messaging.

Rationale for use of SIP

SIP is an application layer signaling protocol, designed to establish calls over which multimedia packets can be separately transferred in real-time. It is not designed to be used to transfer data/media but normally relies on RTP/RTCP to do this. This separation of signaling from the data exchange is an important characteristic that makes it possible to use different paradigms and modes of operation for signaling, control messaging and data transfer as appropriate. In typical SIP operation, SIP signaling is used to establish a call, the session characteristics are negotiated between the end devices using SDP and the media is transferred using RTP/RTCP. Further investigation of SIP reveals other advantages for building sensory network messaging:

Scalability: SIP is a very scalable protocol. It works from end-to-end across the LAN and the WAN. It does not rely solely on multicast/broadcast technologies to reach a destination endpoint. It has a strong concept of routing which enables a packet to traverse from source to destination using intermediate existing routes; hopping from one node to another till it reaches its final destination. Further SIP can operate on both UDP and TCP which allows SIP based servers to scale well.

Flexibility/Extensibility: In SIP it is very simple to add extensions to support new features. The protocol is defined in a way that any provider can define extensions easily to the existing grammar set to add features which may not exist in the core SIP specification.

Registration/Location: In SIP it is not necessary that a calling device needs to know exactly where to locate the called device. A participating device registers its current location with a central server node.

Simplicity: An important attribute for protocols to be used in the NGN is that they be simple enough to support the rapid deployment of new services. SIP enables such service creation and deployment.

Security: SIP provides both authentication and encryption to provide end-end security.

Event Notification: SIP has been extended to introduce SUBSCRIBE and NOTIFY messages which enable elements to "subscribe" to certain events and can be notified when they occur.

Unicast/Multicast Support: when used in conjunction with the Session Description Protocol (SDP), a separate protocol designed to negotiate session parameters, SIP can establish calls which will can use unicast or multicast delivery for content transfer.

Example: A SIP-based Instant Messaging Service

An interesting possibility is to consider extending the capability of SIP to include at least the transfer of short messages in real-time, which is a typical requirement of sensor nodes. Such an approach would not necessarily require SIP to establish a call but would require SIP's ability to register and relove the names of participating nodes. Such an implementation would be useful in a mobile "bursty" environment or an environment in which bandwidth utilization is at a premium.

Instant messaging is defined as the exchange of content between a set of participants in real time. We will consider short simple textual messages only. Although forms of Instant Messaging have been in existence for quite some time, most implementations are proprietary and there is no Internet Application protocol specifically designed to support this function. Messaging between the nodes of a real-time mobile sensor network could be considered as an Instant Messaging application. Such an application could be implemented by using SIP but without requiring the establishment of a call. There is currently a proposal to extend the SIP specification by adding a new MESSAGE method. This method supports both the addressing and the transfer of any MIME type content between nodes but does not require prior call establishment. A MESSAGE request may traverse a set of SIP proxies using a variety of transport mechanisms (UDP, TCP) before reaching its destination. The destination for each hop is located using the address resolution rules detailed in the SIP specifications. During traversal, each proxy may rewrite the request address based on available routing information. This method leverages Routing like functionality (the pre-pending of proxy information in this case) to provide a reply path. Provisional and final responses to the request will be returned to the sender as with any other SIP request.

An example message flow is shown in Figure 3. The message flow shows an initial IM sent from User 1 to User 2, both users in the same domain, "domain", through a single proxy.

```
     |  F1 MESSAGE          |                          |
     |--------------------> |  F2 MESSAGE              |
     |                      | ----------------------->|
     |                      |                          |
     |                      |  F3 200 OK               |
     |                      | <-----------------------|
     |  F4 200 OK           |                          |
     |<-------------------- |                          |
     |                      |                          |
     |                      |                          |
     |                      |                          |
 User 1                   Proxy                    User 2
```

Figure 3: Example Message Flow

Message F1	Message F4

`MESSAGE    im:user2@domain.com`
`SIP/2.0`
`Via:               SIP/2.0/UDP`
`user1pc.domain.com`
`From: im:user1@domain.com`
`To: im:user2@domain.com`
`Call-ID: asd88asd77a@1.2.3.4`
`CSeq: 1 MESSAGE`
`Content-Type: text/plain`
`Content-Length: 18`
 	
`Conor, hello world`
	`SIP/2.0 200 OK`
	`Via: SIP/2.0/UDP user1pc.domain.com`
	`From: im:user1@domain.com`
	`To:im:user2@domain.com;tag=ab8asdasd9`
	`Call-ID: asd88asd77a@1.2.3.4`
	`CSeq: 1 MESSAGE`
	`Content-Length: 0`
	
	Note that most of the header fields are simply reflected in the response. The proxy receives the response, strips off the top Via, and forwards to the address in the next Via, user1pc.domain.com and the result message is F4

Conclusions

Modern embedded systems can support either partial or the entire TCP/IP stack and will be inter-networked over the NGN using Internet protocols. This implies that more distributed architectures will possible and in the area of mobile sensor networks, it will be possible to use a model based on ad hoc networks of autonomous nodes. Although the fundamental push and pull messaging paradigms will still provide the basic linkage, improved modes of operation such as interim and peer-to-peer messaging will be needed to support mobile sensors in increasingly pervasive computing environments. Traditional internet protocols will not support these improved modes of operation but emerging protocols such as SIP and SDP will. The design and implementation of an Instant Messaging application for embedded devices may be a convenient vehicle to research new protocols and the extension of the existing SIP protocol.