Application Communications Requirements for "The Internet of Things"

Bob Dolin
Echelon Corporation

I. Introduction

The purpose of this paper is to document the requirements that the collection of applications sometimes referred to as "the Internet of Things" need based upon the knowledge gleaned from years of experience providing distributed communications and control technology for applications that run on resource constrained devices .

Echelon Corporation has been in the business of providing technology and components to create low cost, peer to peer networks for very resource constrained devices for about 20 years now. Echelon has shipped over 100 million nodes, and other companies who implement ANSI/CEA 709.1 or EN-14908-1 or GB/Z 20177.1, the U.S., European, and Chinese standards of the protocol invented by Echelon, have shipped many additional nodes over Echelon's 100 million node figure.

Of Echelon's shipments about 35 million nodes are Smart Meters and about 65 million nodes are in a variety of building, industrial, home and utility applications. Since this paper is about application requirements, the credit for generating these requirements largely goes to Echelon's customers as they are the creators of most of the applications that use the protocol to help implement their networked applications. Today, none of these nodes are IP based. Back when Echelon began, IPv4 was what was available, and that version of IP is not suited to environments with short application messages (as few as 3 or 4 bytes of application data) and limited network bandwidth. This is due to IPv4's relatively long and not very compressible headers that drive both RAM consumption for buffers and bandwidth consumption in protocol overhead versus the very short amount of application data per packet sent that is found in control applications. Today, with IPv6, it is possible to build resource constrained IP based nodes using an adaptation layer for IPv6 header compression. These nodes can operate on networks with limited bandwidth. It is also possible to build a set of services on top of IPv6/UDP that meet all the communications needs of application developers who currently use the EN-14908-1 protocol. Echelon demonstrated such an implementation at the last IPSO face to face meeting on October 11-12, 2010.

II. Applications

The set of applications envisioned for the Internet of Things is very diverse, as the goal is to provide a single communications technology that can span all device based applications: Smart Meters, Smart Grids, home energy networks, building automation networks, process control networks, factory automation networks, and even some applications that aren't traditionally thought of as networks: street lights, theaters on Broadway in New York, and even the Hotel Bellagio fountain in Las Vegas. These last two applications are mentioned because what the network does is very visible, and is likely to have been seen by many readers.

The Bellagio fountain is a huge pool of water with jets that can shoot water up high and in different directions. Each jet is a node on the network. Synchronized to the jets, is colored theatrical lighting. Each light is a node. The water jets and lights are synchronized to music such that as the music changes, lights are aimed and colored and water jets shoot water at different heights and directions to create a very dramatic show. Rather than read anymore about it, it would be best to go to http://www.5min.com/Video/Learn-About-the-Bellagio-Fountains-278834664 and watch what it does.

The fountain is a distributed control system built with the EN-14908-1 protocol. While the application appears unique, it does many of the things that any industrial or building control system does. Things are turned on and off in real time, motion is timed and controlled in synchronization to other moving parts – all across the network. A great many of the applications envisioned for the

Internet of Things are, in fact, distributed control systems. Sensors sense things and publish their data, controllers or actuators receive sensor data and decide to turn on and at what speed, a controller times events and sends messages to orchestrate actions or modes of the system. Even systems that are referred to as monitoring systems have this requirement for control. In a patient monitoring system, for example, the system detects, for example an irregular heartbeat, and then actuates some sort alarm or page to personnel to come to the patient's aid, and also issue the event to an event monitoring log. Given that there is this commonality across these broad applications, there should be a way to provide a set of common, standard communication services that enable these applications over an IPv6 network.

III. Communications Requirements

All the requirements below do not apply to all applications. However, having a rich set of services in a protocol stack allows that protocol stack to be used across the entire application space, and application developers can know and depend upon the common set of communications services as they implement. The scope of these requirements is for the protocol stack that exists between the MAC/PHY and the application itself. No attempt is made to mandate that some of the requirements be met in the network, transport, or application layers. It is notable that no mention of determinism of the network is made in these requirements. Determinism, while a requirement for some applications, is typically a feature of a MAC/PHY layer and thus, is outside the scope of this document.

The three dominating constraints on these sorts of applications are link bandwidth, application response time and RAM. It follows then that they become the first three requirements:

1. The protocol stack must recover from intermittent packet loss quickly via packet retransmission or report a message failure to the application. Rationale: The sorts of links that these networks run on have very low bandwidth compared to Ethernet, and unlike Ethernet the links are not nearly as reliable. Packets can be lost due to interference and noise as well as due to collisions. When these events happen, and they happen frequently, more bandwidth is consumed to recover from the loss in the form of a packet retransmission. Secondly, because these systems typically have real time constraints, delivering the packet well beyond the application's timing constraints is not important or even desiriable.

2. It must be possible to engineer the communications network such that the real time requirements of the application are met. This involves being able to:

a. Design the network to meet response time criteria by limiting the number of nodes per link, and tune the communications such that the network will not become overloaded.

b. Specify that a given communications transaction will either succeed or fail within a specified time, with the success or failure of that transaction known to the application.

Rationale: In the Bellagio fountain example, a late packet would result in some node not doing its function in synchronization with the show. On a factory floor, a material handler might drop something, in a semiconductor fab line, a wafer handler might fail to place a wafer on a probe station. Late packets are communication failures in most control systems.

3. The protocol stack must implement all services that are needed in all nodes. These services are all the requirements in this document with the exception of number 11. To meet this requirement, RAM consumption for the protocol stack must be limited to provide the application with adequate RAM as well. To put this requirement in perspective, currently available candidate devices the combine both the microcontroller and an IEEE 802.15.4 radio typically have between 8K and 12K RAM to share between the protocol stack and the application. Most

application developers, when they pick such a platform, expect to be able to use much of the RAM for their applications.

Rationale: In the world of low cost systems on a chip (SOCs), RAM is the most precious resource. In communications applications RAM is needed for buffers, to maintain state to know when to retry a packet, to detect a duplicate packet, to put packets in correct order for delivery to the application, etc. These memory requirements are all in direct competition with what the application needs for its memory – the more used by the communications, the less that can be used by the application. Given that these nodes are in a cost sensitive environment, the use of SOCs is a way to meet the cost constraints of the application.

4. The protocol stack must be independent of the underlying MAC/PHY. Rationale: There is no single link that meets all communication needs for the Internet of Things. Today multiple RF links, multiple power line links and a variety of wired solutions are needed to implement the various applications. Furthermore, transceiver development is an area of active research and investment, so the protocol stack must be able to take advantage of new technologies as they become available.

5. The protocol stack must scale to thousands of nodes and multiple links of different speeds in a single logical network. Rationale: Many building and factory systems today are composed of well over 1000 nodes spread out among multiple links with a high speed backbone.

6. Network-wide (spanning all links within the system) multicast must be supported. The notion of multicast group membership must be supported so that all applications do not see all multicasts and consume the node resources required to do more than discard the packet at a low layer in the stack. Rationale: multicast conserves bandwidth and improves response time over multiple, serial unicast messages. When closing a control loop over a network it is sometimes critical that all nodes that subscribe to a sensor value get that value at very nearly the same time. Applications cannot be constrained to have all the nodes in their multicast group on their link as some such messages, such as emergency messages must go to most or even all of the nodes on the network. Applications do not have the memory to process all multicasts just to discover that they do not apply to their node.

7. Confirmed, network-wide (spanning all links within the system) multicast must be supported. Rationale: in applications where the message must get through or a major equipment shutdown is required, for example for safety reasons, the sending node must be able to have confirmation that its message was received by all the members of the multicast group.

8. The protocol stack must support duplicate packet detection and resend the previously generated response without reprocessing or regenerating it. Rationale: There are some sorts of transactions that are inherently not idempotent. For example, an electricity customer is on a pre-pay contract with the utility. The customer adds money to his account and the additional credit is transferred to the customer’s meter, but the meter acknowledgement is lost. So, the utility re-sends the add credit message. Correct behavior would be that the meter only adds the credit one time.

9. The protocol stack must support a mechanism that allows emergency messages to be routed in an expedited manner to overcome queuing delays within the nodes as well as queuing delays in routers between links. Rationale: in control systems sometimes all the nodes are synchronized to an external event that causes a flood of messages. Not all those messages are important in dealing with the external event (like the oil refinery is about to catch fire), but some

messages, that could avoid the impending problem must be propagated quickly across the network.

10. The protocol stack must ensure that packets are received in the order they are sent between any source and destination. Rationale: There are many control operations that depend upon a sequence to prevent damage to equipment or simply to just work correctly.

11. The protocol stack must support a sender node being able to poll multiple nodes in sequence without waiting for the response from one node to arrive before going on to the next node. Rationale: Most control systems have supervisory nodes that ping the status of all the nodes in the network and drive an operator display of the system health. In this operation, if a node is down, the update of the entire display will halt until communication with the down node finally times out after some number of retries unless the protocol supports having multiple responses outstanding and a means to correlate those responses to original requests.

12. The protocol stack must support peer-to-peer communications. Rationale: For response time reasons, nodes cannot wait to be granted access to the network by a server. For finely distributed systems, the nodes must interact as groups performing a function across the network, e.g. light dimmers and lights.

13. It must be possible to provision timers in the protocol stack to indicate when re-send a packet that has not been confirmed. These timers should be individually provisioned according to the destination address in the packet. Rationale: This can improve response time and limit bandwidth consumption. How quickly a node should retry a message is a function of the round trip delay to/from the destination address. Retrying too early wastes bandwidth and may cause network congestion and packet collisions. Retrying too late makes response time suffer when a packet is lost. In a network composed of multiple links of differing speeds, a resource constrained node cannot be expected to know the round trip delays to all the subscribers of its data. Therefore, for large networks, a node with topology knowledge needs to provision these parameters.

14. It must be possible to discover the application level information that a node can publish over the network. Rationale: Some networks are ad-hoc in their formation and allow nodes to come and go without a management station being available or required.

15. The protocol must have a means to transfer a sequence of packets as a logical unit, for example a firmware upgrade, a data log, provisioning information. Rationale: Data logging is a common control network function, being able to upgrade firmware is often required, and provisioning such as linearization tables for sensors, calibration data, etc. is often needed by sensor nodes.

16. Future versions of the protocol must work with prior versions and provide all the same capabilities as prior versions. Rationale: Control networks are long-lived, as long as 20+ years in many cases, yet they are networks, so additional nodes and additional applications are added to them over time. It is unreasonable, and often prohibitively costly to upgrade all the existing nodes in a network to the new version – they may not have enough memory, or there may be some other constraint.

17. There must be some lightweight application interoperability model to facilitate the exchange of data between publishers and subscribers. Rationale:

Resource constrained nodes cannot support parsers of data streams, for example, today’s xml parsers are code space and RAM intensive.

18. Today, the vast majority of control networks are not secured. This will change, so the protocol must be able to support strong encryption, mutual authentication, and protection against record/playback attacks. To sell networks to the U.S. government, the requirement is that the algorithms used be NSA suite B approved. Rationale: as the world moves from hardwired control systems and closed, unconnected networks to networks that can be connected to the internet, the attacks perfected on the internet can be repurposed to attack the Internet of Things.