

what conference?

A Lightweight Protocol for Wireless Sensor Networks

Prashant Agrawal[†] Tan Sun Teck Ananda A.L.
Centre for Internet Research, School of Computing,
National University of Singapore, Singapore
Email : prashant@lit.a-star.edu.sg, {tanst, ananda}@comp.nus.edu.sg



Abstract- Wireless sensor networks are widely used in data acquisition frameworks. Unlike conventional networked devices, wireless sensors run on low power energy sources, and have limited memory buffers, minimal computational complexity, and low processing speeds. As such, the complexities involved with conventional network protocols cannot be realized in wireless sensors. A data dissemination protocol for wireless sensor network should be well aware of sensor limitations. In addition, it should also take into consideration, the unique aspects of applications running over wireless sensor networks. In this paper, we propose a new network protocol, Simple Wireless Sensor Protocol (SWSP) for wireless sensors which provides a reliable data communication between wireless sensors with minimum resource overhead

Keywords- wireless sensor networks; data dissemination; lightweight protocol

I. INTRODUCTION

Wireless sensor networks are a judicious choice for data acquisition frameworks as wireless sensors can be conveniently placed in non-accessible places, support mobility and dynamic topology and can scale better than their wired counterparts. As such, they are widely used in sensing applications like temperature, pressure, acoustics, motion detection as well as for data acquisition in monitoring, manufacturing, biodevices etc. A wireless sensor network is a collection of wireless sensors covering a small area, typically less than 10m. Every sensor performs independent sensing, processing and transmission of data over wireless link to a data collection node. The data collection node aggregates data from multiple sensors and processes it.

In general, wireless sensor networks are deployed for data acquisition purposes. They have a fixed pattern of data flow. Sensors transmit small amount of data at periodic intervals of time. The data may be as small as few bytes. Usually the data contains the value of one or more sensed attributes. The time interval between successive data transmission by a sensor depends on the requirement of the application. For example, for a sensor for surveillance purposes, the data transmission can be quite frequent while for environment monitoring it could be significantly infrequent.

Several low power wireless sensor designs have been proposed in literature [1][2][3][5]. A wireless sensor can have size of the order of millimetres. In this, resides a transducer unit that converts the sensed analog signals to digital, a control unit that receives digital input from the sensing unit and perform protocol operations, a radio unit which transmits the packet over wireless link and a battery which is the power source of the entire device. The control unit can function at micro-watts level while the radio transmissions typically require energy in milliwatts. The focus of our attention is the control unit which hosts network protocol. Although the control unit consumes energy of the orders of magnitude less than radio unit, the complexities involved with the conventional network protocol stack like TCP/IP largely determine the behaviour of radio transmissions and also increase the cost of a sensor node.

Wireless sensor networks are expected to support dynamic join and leave of the sensors, service discovery and device auto-configuration. At present, these features form a part of higher layer services which function over TCP/IP. This makes it difficult to eliminate TCP/IP from sensor protocol stack. Also, there are a number of legacy implementation of such services but none of them is adopted as a standard.

The characteristics of wireless sensor networks are highly related to their topological configuration. A wireless sensor network can be an ad-hoc network or a fixed network. In an ad-hoc network, sensors can organize themselves in clusters of subnetworks and perform higher level functions of routing or mobility management between themselves. In the fixed network, a higher peer usually called an access point, manages communication between a group of sensors. Sensors of different groups communicate via their access points. Thus, for fixed wireless sensor networks, much of the functionality of IP layer may be redundant.

In the above discussion, we see that TCP/IP protocol stack does not fit well into the requirements and limitations imposed by wireless sensor design, wireless sensor network and wireless sensor data transmission.

[†] Presently working at the Laboratories for Information Technology, 21 Heng Mui Keng Terrace, Singapore 119613

why TCP?

Really?

True

How does this fit into your proposal

MIT
University
but OK

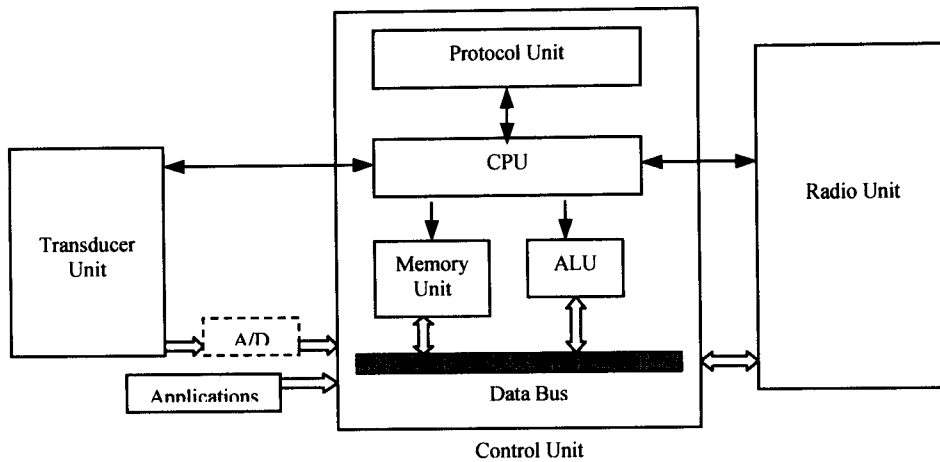


Figure 1. Functional Components of a typical wireless sensor

In this paper, we discuss the design considerations of an alternative network protocol for wireless sensors which we name as Simple Wireless Sensor Protocol (SWSP). We then present the syntax and semantics for the same. The rest of the paper is organized as follows. In Section II, we discuss the wireless sensor network design for SWSP implementation. In Section III, we present a formal introduction of SWSP and describe the semantics of the protocol. In Section IV, we give the performance analysis of SWSP.

II. WIRELESS SENSOR NETWORK SYSTEM DESIGN

As discussed before, the major challenges in the design of a wireless sensor network are the power, storage, processing and computing limitations of the sensor. In this section, we present the wireless sensor and wireless sensor network architectures for our system as they bear a direct relationship with the network protocol design. Our system design gathers results from prior research works [4][5][6] done in this area. Whereas [4],[5] gives a hardware perspective, [6] gives a topological view of system design.

Figure 1 gives a schematic diagram of the functional components of a typical wireless sensor. We envision a wireless sensor node with on-chip configuration information about the applications. The network protocol is realized in the control unit which is a microcontroller. The network protocol runs through the information and configures the sensor dynamically. The data flows unidirectionally from transducer unit to the control unit. The control unit contains memory units to save data, an Arithmetic and Logic Unit (ALU) to do calculations, comparisons and other data manipulations, a timer unit to schedule data transmissions, a protocol unit to implement the network protocol functions, and finally a CPU to control all the above units. For low power consumption, the CPU should function at low speeds. The radio unit hosts the MAC protocol functionality. The MAC protocol handles data transmission over wireless link. Again, low power demands low duty cycle transmissions.

One of the important characteristics of sensor networks is that the data transmission does not require higher power radio signals as the sensors are located close to each other. This implies that the power required for transmission and reception in sensor networks is nearly the same. Thus, even the ~~receptions are fairly expensive~~. If the sensors are to implement routing, they are required to process every packet they receive even when it is not the destination. If no routing is present, they may shut down their radios when they do not expect any packets destined to them. Since, in a data acquisition framework, sensors are data relay units and do not directly communicate with each other, they can be organized in a fixed network topology. In a fixed network topology, routing decisions can be relegated to the access point, thus giving major power benefits. Hence we advocate fixed wireless network topology for our system.

Again, for wireless sensor networks for data acquisition purposes, a hierarchy of devices is recommended. As the sensors are short of computational capabilities and storage, the data collection points cannot be the sensors themselves. In our system, the Access Points for wireless communication are also the data collection nodes. However, there always exists a trade-off between the limits to processing at the sensor and the amount of data transmission.

Since we contend that TCP/IP is not a necessity for sensor communication and the data relayed by wireless sensors is relevant only to the local network, IP addressing is not required for the sensors. Instead, we propose a dynamic addressing scheme wherein every sensor is assigned a unique identifier when it joins the network. The identifier is taken back when the sensor leaves the network, and is reassigned to new sensors. The benefits of IP addressing are not apparent in sensor networks as we do not take routing into consideration. On the other hand, using identifiers facilitates better management of the system.

Since the applications hosted by a sensor are limited and fairly descriptive in nature, every sensor can be considered a unique service point. Thus, sensors are suitable for future communication models, where devices are recognized by the services they offer. In a managed network, sensors act as service providers and not service consumers. They need to broadcast their services to a central server which, in our model, happen to be the access point of wireless network. In our system, we combine this sensor functionality with the network protocol.

III. SWSP : DESIGN AND SEMANTICS

In this section, we discuss the basics of SWSP design and compare them with TCP. We then describe the protocol semantics.

TCP is a widely used transport layer protocol. Its strengths are efficient congestion control, flow control and reliable communication. However, it also has some drawbacks when implemented in wireless networks. Significant research work has been done in congestion avoidance mechanisms of TCP and their impact on wireless lossy links. If TCP is to be implemented in wireless sensor networks, the congestion avoidance algorithms can be disabled as they are low bandwidth networks. However, as discussed in [7][8], there are some additional issues in implementing TCP in a chip or in a microcontroller.

One of the issues is the availability of buffers. A node may have several simultaneously active TCP connections. If every connection has a large number of packets to send or receive, the buffer space required may exceed available sensor resources. In TCP, reducing the window size will ensure less buffer space requirement. We follow the same model with a very small window size.

As opposed to multiple connections in TCP, our system relies on a single connection between the sensor and data collection node. The connection is established when a sensor registers itself (explained later) with the data collection node and is released after the device is powered off. In order to maintain the liveliness of connection, keep-alive (KA) messages are exchanged between the sensor and data collection node. Failure to receive keep-alive message invokes failure recovery mechanisms in data collection node. In a similar manner, every data collection node broadcasts a 'Ready to Receive' (RR) message to the network. A sensor node which does not receive RR message cannot communicate with the data collection node and is unregistered by itself. In such a case, if a mobile sensor receives RR message from another data collection node, it registers itself again to that node.

For reliability, every data packet sent requires an acknowledgement. In a standard TCP implementation, group acknowledgement is sent for a sequence of segments. Since in our system, a single data collection node sends acknowledgement to multiple sensors, we can have a single acknowledgement packet for data from all sensors. This will

ensure proper bandwidth utilization as well as low aggregate processing cost for the acknowledgements.

TCP requires retransmission timers for every data packet sent. If no acknowledgement is received before the timer times out, the packet is retransmitted. A sensor is incapable of maintaining timer for every data segments of every connection. In [8], retransmission is replaced by negative acknowledgements from the requester. However, sensors may also send unsolicited data to the data collection point. In our design, a retransmission counter is maintained for every data packet sent. The counter is incremented each time a group acknowledgement is received from the data collection point. The data is retransmitted if the counter exceeds retransmission counts.

Having discussed the foundations of the protocol, we now describe SWSP in detail. SWSP is an event driven protocol. Figure 2 gives the state transitions for the sensor. There are six states to which the sensor can transition: connecting, connected, requested, ack_wait, disconnecting, disconnected. The first state and the last two of the states relate to the entry and exit of the sensor in the network. A sensor spends most of its time in the second, third and fourth state. The events governing the state transitions are simple and do not require timers or extensive input/output communications. We now describe each of the state and their related protocol aspects.

A non-powered sensor is always in a 'disconnected' state. When a sensor powers on, it enters in the 'connecting' state. It reads the on-chip application and configuration information. The protocol unit prepares a 'register' message and broadcasts it to the network. The broadcast address of the radio interface hardware is used for this purpose. The 'register' message contains services (applications) it supports and the hardware address of its MAC interface. The former is a unique identifier of the service as suggested by SWSP. The latter has to be obtained from the radio unit. Such information is sufficient enough to identify the device and its services.

Since many data collection nodes may receive the 'register' message, they need to be synchronized so that only one acknowledges the message. The 'register' message is retransmitted if not acknowledged after receipt of a RR message. On receipt of acknowledgement, the sensor goes in 'connected' state. The acknowledgement assigns an identifier to the sensor which is maintained throughout its connection with the data collection node. A sensor in 'connected' state may self transmit the data or require a request from the data collection node. In the latter case, the sensor enters into 'requested' state, wherein, it gathers the data and sends it to the data collection node.

After the data is sent, the sensor is in 'ack_wait' state wherein it waits for the acknowledgement. In order to reduce the number of message exchanges, we piggyback the data acknowledgements on RR message.

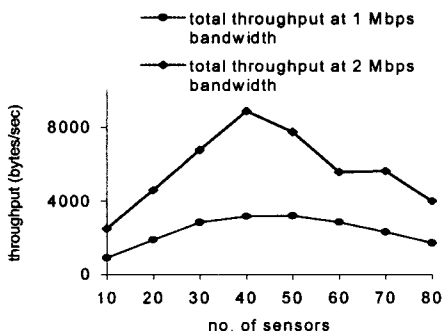
The throughput of useful data transmission is expected to be lower than the TCP throughput for the same. This is because of the small window size and group acknowledgement mechanism. Due to low-bandwidth requirements of sensor networks, the low throughput is acceptable. In the later part of this section, we examine the relation of throughput with the number of sensors in the network. This is relevant as we wish to study the effect of increase in the number of sensors on the throughput and delay characteristics of the network. We apprehend some degradation in the system performance as sensors are increased. We will then analyze if SWSP has any role to play in it.

To begin with the SWSP performance evaluation, we implemented the protocol on a PC running Linux operating system. We studied some of the existing light-weight implementations of TCP, Tiny-TCP and LwIP[9]. These implementations gave us useful directions to prepare the source code for SWSP. Unfortunately, we do not have the resources to port the protocol implementation on hardware. Since we presume the present day PC processors to be quite fast in comparison with the sensor processing, we did not make efforts to optimize the code for processing gains. Also, since we wished to simulate a number of sensor devices, we run them as independent processes on a single PC. We believe that because of the fast network cards and

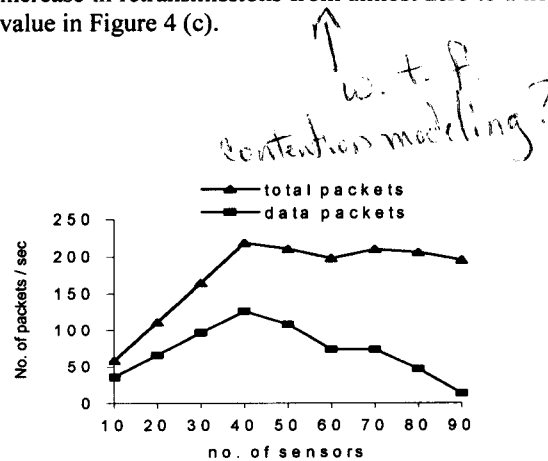
high processing speeds, there will not be any performance bottleneck by doing so. The data collection node or the access point of wireless network resides in another PC. The two PCs communicate through a wireless channel. The network interfaces are 802.11b compatible wavelan cards.

We performed a series of performance tests. In each test, we are interested in the throughput and round trip delays of the network. We defined throughput as the total number of bytes received by the access point per unit time. The round trip delay was calculated as the time elapsed at the sender between a packet sent and its acknowledgement received. In each run, the length of data was fixed and the number of sensors was varied from 20 to 80 with an increment of 10 each time. Initially, the wavelan card was set at 1 Mbps bandwidth. Later we repeated the whole set of experiments with 2 Mbps bandwidth.

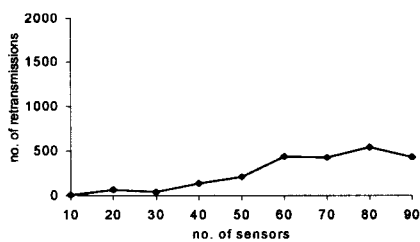
Figure 4 (a) shows the plot of the throughput of the network vs number of sensors at data length of 60 bytes. The throughput is observed to increase to a maximum value and decrease thereafter. The increase in throughput is obvious as the number of sensors is increased. When the number of sensors are increased beyond a threshold, the access point is unable to receive the deluge of packets. Thus many data packets need to be retransmitted. This is reflected in the increase in retransmissions from almost zero to a non-trivial value in Figure 4 (c).



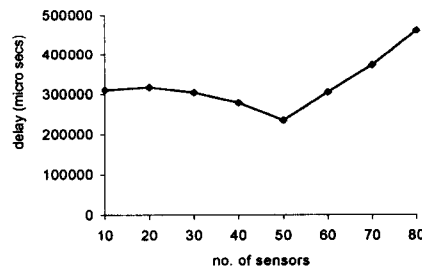
a) Throughput Vs No. of sensors



b) No. of Packets Vs No. of sensors



c) No. of retransmissions Vs No. of sensors



d) Round trip delay Vs No. of sensors

Figure 4. Performance of SWSP

Interestingly, the value of the number of sensors where the throughput reaches a maximum is almost the same i.e. 50 for the two bandwidths. Also, for our series of experiments for different data lengths, it remains the same. However, the peak throughput value is different for each case. We observed frequent re-setting of the wireless interface card at the receiver due to buffer overflow problems. Thus we conclude that the throughput behaviour as observed in the above test cases is a result of wireless interface characteristics.

Figure 4(b) shows a plot for the number of total as well as data packets received per second at the access point. The plot shows that the number of data packets received drop significantly as the number of sensors are increased. However, the total number of packets received did not decrease in that proportion. This implies that the number of control packets increased sharply. This is expected since the sensors are kept waiting for a period of 3 RR before they can retransmit the packet.

Finally, we also plot the round trip delay for the network in Figure 4 (d) at data length of 60 bytes. The round trip delay is minimum at 50 sensors when the throughput is maximum. It then increases gradually with the increase in the number of sensors. The maximum RTT delay observed was 800 ms (not shown in figure). For sensors with very high data transmission rates, this may not be a suitable value. We regard this as a limitation of our protocol. Given the constraints, we are unable to sacrifice other benefits of the protocol for improvement in latency beyond this value.

V SUMMARY

TCP is unsuitable for wireless sensor networks which are constrained by their size, memory, power and speed. We proposed an alternative network protocol which inherits the basic characteristics of TCP but is significantly less complex and less resource demanding than TCP. We justified various aspects of the design of this protocol. Our performance analysis show that the protocol does not significantly influence the throughput behaviour. The average round trip delay time for a packet can be very high. However, given the resource limitations, we cannot reduce it appreciably.

ACKNOWLEDGMENT

The first author would like to thank Mohan Kankanhalli for his support and the Centre for Internet Research, NUS for providing the facilities.

REFERENCES

- [1] G. Asada, et al., "Wireless Integrated Network Sensors: Low Power Systems on a chip," *Procs. of the European Solid State Circuits Conf.*, 1998
- [2] Kristopher S.J. Pister, Joseph M. Kahn, Bernhard E. Boser, "Smart Dust: Wireless Networks of Millimeter-Scale Sensor Nodes," *Research Summary*, Electronics Research Laboratory, Deptt. Of Elect. and Comp. Sc., Univ. of California, Berkely.

- [3] Asad A. Abidi, Gregory J. Pottie, William J. Kaiser, "Power-Conscious Design of Wireless Circuits and Systems," *Proc. of IEEE*, Vol 88, no. 10, pp 1528-45, October 2000
- [4] Anantha Chandrakasan, et al., "Design Considerations for Distributed Microsensor Systems," *Proc. IEEE 1999 Custom Integrated Circuits Conf. (CICC '99)*, May 1999, pp. 279-286
- [5] Vijay Raghunathan, et al., "Energy-Aware Wireless Microsensor Networks," *IEEE Signal Processing*, March 2002
- [6] Zygmunt J. Haas, "A Communication Infrastructure for Smart Environments: A Position Article," *IEEE Personal Comm.*, October 2000, pp.54-58
- [7] David D. Clark, Van Jacobson, John Romkey, Howard Salwen, "An Analysis of TCP Processing Overhead," *IEEE Comm. Magazine*, Vol. 27, No. 6, June 1989, pp. 23-29
- [8] Janne Riihijarvi, "Providing Network Connectivity for Small Appliances: A Functionally Minimized Embedded Web Server," *IEEE Comm. Mag.*, October 2001, pp. 74-79
- [9] Adam Dunkels, "Design and Implementation of LWIP TCP/IP Stack," *Technical Report*, Swedish Institute of Computer Science, Feb, 2001

