Proceedings of the IASTED International Symposium
Distributed Sensor Networks (DSN 2008)
November 16 - 18, 2008 Orlando, Florida, USA

TSMP: TIME SYNCHRONIZED MESH PROTOCOL

Kristofer S. J. Pister, Lance Doherty
Dust Networks
30695 Huntwood Ave., Hayward, CA 94544
USA
{kpister,ldoherty}@dustnetworks.com

ABSTRACT
The Time Synchronized Mesh Protocol (TSMP) enables reliable, low power, secure communication in a managed wireless mesh network. TSMP is a medium access and networking protocol designed for the recently ratified Wireless HART standard in industrial automation. TSMP benefits from synchronization of nodes in a multi-hop network to within a few hundred microseconds, allowing scheduling of collision-free pair-wise and broadcast communication to meet the traffic needs of all nodes while cycling through all available channels. Latency and reliability guarantees can be traded off for energy use, though our focus has been on providing high reliability (>99.9%) networks at the lowest power possible. TSMP has been demonstrated in multi-hop networks exceeding 250 nodes per access point, thousands of nodes with multiple access points, radio duty cycles of 0.01%, and with devices at radically different temperatures and traffic levels. With the 802.15.4 physical layer and 10 ms time slots, TSMP can theoretically achieve a secure payload throughput of 76 kbps at a single egress point.

KEY WORDS
Wireless Sensor Networks, Time Synchronization, Mesh

1. INTRODUCTION

Additional to Wireless Sensor Network (WSN) research topics, a growing commercial and industrial application space now exists. These applications require reliable, timely, secure delivery of packets at low power. A survey of the research literature suggests similar objectives [1]. Diligent attention to these requirements led to the development of TSMP, the Time Synchronized Mesh Protocol. The potential of this approach is evident in its recent adoption as the basis for the Wireless HART [2] and ISA100 protocols. Both protocols are intended for industrial automation where applications are extremely performance-sensitive; TSMP's adoption in this regime came only after extensive evaluation in hundreds of real-world deployments over a period of a several years.

The cornerstones of TSMP performance are network-wide time synch, channel hopping, dedicated slotted unicast communication bandwidth, link-layer ACKs, graph-based routing, and multi-layer security on every packet. Much work has been done on time synch in sensor networks [3], and many groups have demonstrated the superior performance that it affords [4, 5, 6]. TSMP, in production since 2004, is the first WSN protocol to use time synch to schedule collision-free channel hopping communication.

Perhaps also novel, and certainly contrary to conventional wisdom in the WSN community, TSMP was developed to be used in a managed network with a centralized controller that coordinates the communication schedule for the network. A managed network has the prime benefit that each channel in the spectrum can be used without concern for collisions or wasteful overhearing of packets. Network management is sometimes dismissed on the grounds of lacking scalability and robustness, and confusion over the viability of network-wide synch. However, implementation and testing of TSMP over several product generations has shown that these criteria can be met (and to date have only been met) in a managed network. In particular, TSMP has been targeted to:

1. Reliability at low-power: >99.9% packet delivery with years of battery life for all nodes
2. Scalability: hundreds of meshed nodes per manager, thousands in the same RF environment
3. Flexibility: support different time-varying traffic patterns from different nodes
4. Security: guarantee confidentiality, integrity, and authenticity of all packets
5. Environment: nodes operate between -40°C and 85°C and in radically changing RF noise levels

TSMP provides a framework for the association of layer 2 resources with layer 3 routes and layer 4 Quality of Service (QoS). Standards specify packet formats and layered transaction models for the pair-wise communication and how multi-hop routes are maintained to a management application. It remains the duty of the network designer to assign data flows based on service requests from nodes and reports on how successful the wireless paths have been.

2. DIVERSITY

TSMP provides reliable data transfer by combining time, frequency, and routing diversity. All nodes in a TSMP network share the same sense of time, accurate to < 1ms. This capability allows time-diversification by scheduling different transmissions to occur during non-overlapping

intervals and thus conserve energy by avoiding collisions. When nodes agree on when communication could occur, both sides of a radio link can reduce their "on time". We will show that the cost of maintaining this synch is small and compares favorably to asynchronous protocols [7,8].

There is no additional preamble used in TSMP transmission beyond the physical-layer (PHY) requirements, and a low idle listen cost is achieved with a short 2ms guard time. Also, it is trivial for sensors attached to time-synched nodes to time-stamp their data.

Packet delivery rates on different channels for the same node pair can vary wildly, as can a single channel over time [9]. By adding channel hopping in addition to Direct Sequence Spread Spectrum (DSSS), TSMP offers frequency-diverse advantages compared to a single-channel solution. First, it increases the network bandwidth linearly in the number of channels. Second, it increases reliability by having each node pair communicate on all potentially good channels decreasing the chance of terminal loss of connectivity. Third, since multi-path fading is frequency-dependent, using more channels increases the effective radio range. TSMP channel hopping is similar to the strategy in [10] to mitigate interferers in 802.11 devices as DSSS alone is insufficient to overcome the deep fades of multipath interference [11]. Lastly, hopping makes TSMP more robust to interference and reduces the impact on other wireless networks.

To provide different levels of QoS, TSMP radio resources and packet flows are organized into independent graphs similar to MPLS labels, ATM circuits, or IPv6 flow labels. The flows travel along routing-diverse meshes: packets generally have more than one possible next hop at each step along their multi-hop route which insulates against single-device failure and congestion.

3. PROTOCOL STRUCTURE

TSMP works by providing a bridge between the physical domain of wireless communication and the logical domain of network-wide data flows. Physically, the wireless channel is divided up in time and frequency and each resulting unit of the channel is assigned to satisfy data flow requirements. Logically, data flows from hop to hop along pre-defined routes from source to destination.

Time is divided up into discrete *time slots* (10ms in this paper) long enough for any single transaction. The longest transaction is one where the sender transmits the maximum length packet and the receiver acknowledges it. Every node in the network must agree on the number of time slots since the beginning of the network (the *Absolute Slot Number*, or ASN).

TSMP is designed to operate on multiple channels. Based on shared ASN, each packet is sent on a pseudo-random channel. This paper uses 2.4GHz channels from the IEEE 802.15.4 standard, each with ~2MHz of DSSS bandwidth. TSMP models RF space as a matrix of slot-channel *cells* (Figure 1). Transactions scheduled in different cells are guaranteed to not interfere with each other. The slot-channel matrix extends from ASN=0 to infinity.

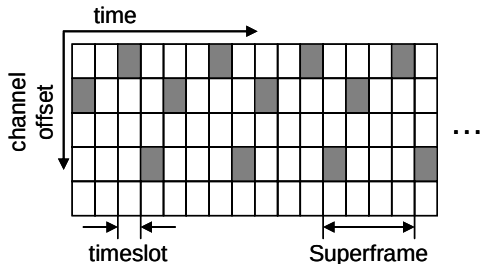

Figure 1. Slot-Channel Matrix for a network with 5 channels. Each cell can contain one transaction.

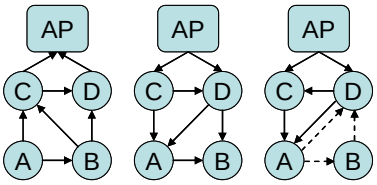

Figure 2. Simple graphs. Left: collection mesh from all nodes to the AP. Middle: broadcast mesh from the AP to all nodes. Right: multicast mesh from the AP to three nodes, and a peer-to-peer mesh from A to D.

A *graph* is a routing structure that forms directed end-to-end connections among devices. For example, a data-collecting graph could join all nodes in the network to the Access Point (AP) where payloads are forwarded to the user. Some simple graphs are shown in Figure 2. Each destination will generally have its own graph, but several sources can share the same graph. In this context, a *destination* could be a single node or a set of nodes. A functional graph is composed of pair-wise *paths* that connect each source to the destination directing the packet at each hop along the way. A typical unidirectional graph terminates only at the destination and does not have loops. A single network generally has multiple graphs, some of which overlap and each individual device can have multiple graphs associated with each of its neighbors. The basic graphs used in TSMP are upstream and downstream meshes: upstream to route packets one hop closer to the manager with each transaction and downstream to route from the manager to all nodes. A downstream graph in this case can implement network broadcast or multicast, where the destination is "all", and allows both source routing and flooding to occur on the same links.

When a packet is originally created at its source, the application layer identifies its flow (final destination and QoS) and supplies it with a fixed Graph ID that will guide it along its multi-hop route to the destination. In a mesh graph, it is possible that multiple next hops are specified at any particular node; redundancy is directly built-in.

A *superframe* is a collection of cells repeating at a constant rate. Events are scheduled to happen in individual cells in a superframe and the network is configured to support these events with periodicity equal to the superframe length. With a 10ms slot, a cell in a 1000-slot superframe repeats every 10s. A cell in a shorter superframe is representative of more bandwidth as it repeats more frequently. In TSMP, the superframe is the translation of the logical data-flows of the graph into the

physical communication schedule for the network. Typically, a network will have several graphs with different superframe lengths. In a network of a single superframe, however, a no-collision guarantee can be trivially made by assigning only one event per cell. For example, the periodically populated shaded cells in Figure 1 show four cycles of a four-slot superframe (16 slots).

A *link* is an event scheduled in one cell. It consists of a superframe ID, source and destination IDs, a slot number referenced to the beginning of the superframe, and a channel offset. The simplest version is a *dedicated* link: one transmitter and one receiver. Here, two nodes communicate periodically once per superframe. If only one transmitter is scheduled, the link is contention-free but Slotted Aloha can be implemented by scheduling multiple transmitters in a single link with a backoff algorithm after packet failures. Either of these two schemes can unicast packets to a single receiver allowing an ACK for timing or reliability purposes. Multiple-destination *broadcast* links are used for quick low-power distribution of packets to several children and work well for flooding nodes along a particular graph. Each time a link arises in a superframe, nodes scheduled to be receivers on this link are required to turn on their radios and await a potential packet from the sender. The senders only initiate transmission if there is a packet waiting for transmission on the graph to which the link belongs.

TSMP links hop pseudo-randomly over a set of channels one packet at a time. Each time a link fires, both sides of the link calculate the radio channel of the communication:

Ch = LUT((ASN+offset)%NumCh)

- Ch is the channel number
- Offset is the link's y-value in the slot-channel matrix
- NumCh is the number of available channels
- LUT is a look-up-table with a static pseudo-random mapping between the inputs from 0 to NumCh-1
- ASN is the absolute slot number

For example, in a monotonic LUT={0, 1, …, 15}, a series of links with consecutive slots and constant offset will cycle through the channels in series, one slot at a time. Blacklists can be used to restrict the set of allowed channels for coexistence purposes. While TSMP changes the 5MHz channel once per slot, Bluetooth changes its 1MHz channel up to 1600 times/s requiring tighter time synch to schedule non-colliding communication.

Based on these structures, the network manager is responsible for assigning links in superframes to ensure on-time delivery of data flows. Today this is done in a central manager. Distributed schemes could allocate blocks of cells for local management.

4. NETWORK PERFORMANCE METRICS

In this section we define a few metrics used to quantify performance. Network *reliability* is the fractional rate of successful delivery of application-layer packets. We measure it in terms of packets received by the manager divided by the number of packets generated by all nodes in the network. Each packet counts the same whether it has a 1B or an 80B payload. In general, higher reliability can be achieved by assigning more links per superframe at the cost of increased power.

Stability is the fractional rate of successful delivery of Data-Link Layer (DLL) packets between any two nodes. For example, if 100 packets were transmitted from A to B and 80 were successfully received by B, the stability of path A-B is 80%. Stability is a function of the wireless channel between two devices, is often seen to change drastically, and in general is not a controlled parameter in TSMP networks. As such, TSMP is designed to enable high reliability networks in the face of low stability paths. Stability is tightly correlated to SNR at the receiver and loosely correlated to distance between devices.

As TSMP has already been implemented on several radio platforms, comparisons of energy consumption are made in this paper according to *radio duty cycle*. This is a measure of the fraction of time that the radio is active (either in receive or transmit mode) at a given node. Since radio operating costs tend to dominate the power budget in WSNs, this provides a useful cross-platform metric and mitigates the need for comparing raw power numbers.

The *latency* of a packet is the difference in time between when it was generated by the application at a node and when it is received at the destination. Latency along a graph can be decreased by adding more links to that graph's superframe, again at the cost of increased power.

In this paper, *throughput* refers to the rate of application-layer payloads at the destination. Packet headers do not count towards this total, nor do packets required for network maintenance, topology building, or time synch.

Our overall strategy in building networks is to allocate enough links to keep network reliability close to 100%. This link number is highly dependent on the stability of paths comprising the network and the network depth. Beyond this allocation, additional links are added if the user is willing to sacrifice battery life for lower latency.

5. PROTOCOL DETAILS

5.1. Physical Layer and Timeslots

TSMP has been implemented on several generations of hardware, comprising 2 different processors and 3 different radios. These include two board-level designs based on the TI MSP430F149 processor. The first used the TI CC1000 at 900MHz with 50 channels of 76.8kbps each and 32 slots per second. The second used the CC2420 at 2.4GHz with 802.15.4 channels and bit rate, and the same slot length. Today, two custom ASICs run TSMP, one with 25 100kbps channels at 900MHz and one 802.15.4 compliant at 2.4GHz. We use the latter in this paper.

The format of a timeslot is shown in Figure 3. Node *A* is transmitting to node *B*, with an optional initial clear channel assessment. Assuming the nodes are already synched (discussed below), *B* knows when to expect the first bit of the preamble. With some model for worst-case

clock skew T_g between *B* and *A*, *B* turns on its receiver T_g seconds before the expected arrival of the first bit from *A*. If a valid preamble and start symbol are detected, *B* listens to the full packet then verifies the 2B CRC and 4B DLL Message Integrity Code (MIC). If both are valid and the packet is unicast, an ACK packet is created with a MIC and CRC. The guard time for the ACK arrival can be small since the nodes are tightly synched by the arrival of the packet. For an 802.15.4 radio the appropriate time slot to accommodate a full packet with current technology is approximately 10ms as shown in Figure 4. If a receiver listens for $2T_g$ and there is no evidence of the preamble, then the receiver turns off. Either no packet was sent or the signal was lost due to inadequate SNR. In typical networks, these "idle listen" events outnumber packets.

5.2. Link-Layer ACKs

In latency-constrained applications, end-to-end ACKs often impose too much overhead. As such, many latency-sensitive internet applications avoid them. At the same time, if data reliability is critical, then packets must not be deleted until successful receipt has been ACKed. This is essential for WSN traffic where data can travel several hops over paths with Packet Error Rates (PER) in the tens of percent: end-to-end ACKs are not time or energy efficient. Link-layer ACKs provide a compromise.

If the received packet has a valid CRC and DLL MIC, an ACK is transmitted while no ACK is sent if the CRC or MIC tests fail. The receiver sends either a positive or negative ACK after a valid packet. The transmitter will only delete the transmitted packet from its queue on reception of a positive ACK which indicates the receiver has accepted the packet. Negative ACKs are generated if the receiver's queue is full for the particular packet type and are useful for distinguishing network congestion from PHY errors. Both ACK types may contain piggybacked time synch information to decrease the marginal cost of maintaining a single network time-base.

ACKs must be cryptographically secure to prevent misinterpreted ACKs from other receivers or networks, undetected ACK corruption, and malicious attack. A bogus or corrupt ACK interpreted as valid by the sender could result in packet loss or incorrect synch information. The combination of MIC and CRC may be overkill for both the packet and the ACK, but the MIC is required for cryptographic integrity, and the CRC is required for 802.15.4 compliance. The standard also specifies a 4B minimum MIC size. As future standards push to shorter timeslots and latency the size of the combined error checking at the MAC layer will be reduced.

ACKs would be expected to have lower PER than longer packets for a constant Bit Error Rate (BER). Additionally, the properties of the channel are relatively stationary over ms timeframes [9]. As such, the BER of a link-layer ACK is lower than the BER of a packet sent at a random time as it necessarily follows a successful transmission. As a lost ACK results in both sides of the link having a copy, the lowered effective BER for ACKs helps naturally reduce duplicate packets in the network.

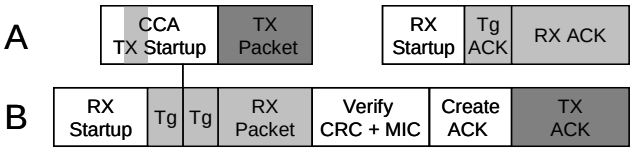

Figure 3. Node A is a worst-case late transmitter to B (lengths not to scale). Shading is dark for TX, light for RX. Node B expects the preamble midway through the guard time if the nodes are exactly synched.

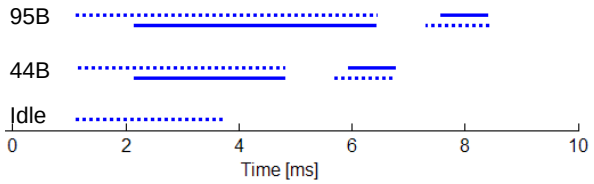

Figure 4. Measured radio on-time for various packet sizes. Lines are dotted for receive and solid for transmit. In the 95B and 44B cases, the upper series is measured at the receiver and the bottom at the sender; the packet transaction is followed by the ACK.

5.3. Time Propagation

TSMP uses regular updates to maintain a shared sense of time in multi-hop WSNs. Pair-wise agreement on active paths must be accurate to within the receive guard time, T_g, or nodes risk losing connection with the network.

Nodes transmit the first bit of their packet as close to the ideal start time as possible in their time frame. The receiver measures the time of arrival of this first bit in its own time frame. The accuracy of this sending and receiving timing is generally better than a few tens of microseconds on most hardware, and can be made substantially better [3]. At this point the receiver knows the difference between the time frames and sends this information back to the transmitter in the ACK, either explicitly in the message, or implicitly with transmission time. Timing is shared on every ACKed packet.

Our implementation propagates time from a single time-master, the AP, to all other devices. Devices with direct AP links use each transaction to synch their clocks to the AP. This results in a clock change of at most T_g seconds. Similarly, when these "first hop" devices talk to their children, the child adopts the parent's clock. This establishes the concept of a "time parent", which may be different than a routing parent. This approach is only optimal in the sense that it is simple, and it is proven to work in a loop-free time parent graph.

5.4. Clock Skew and Drift

Nodes keep track of time by counting the oscillations of a quartz crystal nominally at 32,768Hz, but manufacturing differences lead to ~10ppm deviations at constant temperature, and material properties cause variation >100ppm over the industrial range of -40°C to 85°C. Actively compensated crystal oscillators can be used to keep deviations over temperature and aging under 10ppm. Whatever the hardware, any two nodes will have a bound

on the difference ε in the rate at which their clocks tick. For nodes synched at time t_0, with a synch error δ_{sync}, the worst-case error in their shared sense of time is:

$$\Delta t_{max} = \varepsilon (t-t_0) + \delta_{sync}$$

To maintain connectivity, a node must keep $\Delta t_{max} < T_g$ placing an upper bound on the time between synch events:

$$T_{sync} = (t-t_0)_{max} < (T_g - \delta_{sync})/\varepsilon$$

For example, with a ±1ms guard time, 50μs of synch error, and a ±10ppm clock accuracy:

$$T_{sync} < (0.95\text{ms}) / (20\text{ppm}) = 48\text{s}$$

To stay synched, this node must update every 48s. In a network with 10ms slots, this is a radio duty cycle <0.01%. As time synch improves, these guard times will shrink to hundreds of microseconds and increase their relative benefit over protocols with long listening phases.

TSMP allows two options for synch updates. The first is a child-initiated unicast request for a time update through an ACK called a *keepalive*. The second is a parent-initiated broadcast update commonly known as a *beacon*.

In a network with regular data-reporting to a time-master AP, nodes close to the AP can see traffic much more often than every T_{sync}. These nodes track the AP's clock with no additional traffic when time synch is piggybacked on top of data ACKs. It is the responsibility of the manager to schedule sufficient links to meet the needs of time synch. In this regime, beaconing is wasteful. Conversely, nodes in an alarm-only network, or near the edge of a data-reporting network may not naturally initiate enough traffic to stay synched. Here beaconing can be more energy efficient as several children of a node can be synched with one message. The disadvantage of beaconing stems from the inability to ACK a broadcast packet so the parent cannot know who was synched. Mixing beacons and keepalives may be the most efficient.

5.5. Advertising and Joining

Devices join the network by responding to periodic advertisements from nodes already in the network. When initialized, the AP begins advertising using all allowed user-specified channels. When nodes are asynchronously powered-up, they begin duty-cycled listening across all channels. A node hearing an advertisement synchs to the network, listens for other advertisements, and then sends a join request indicating which neighbors it has heard. If the join request passes security checks, the manager provisions some graphs with links to the best reported neighbors and the new node begins advertising.

The speed of joining depends both on the advertising rate and the duty cycle of the listening nodes. The mean time for a joining node to synch with the network is:

$$T_{sync} = CA/(NPD)$$

- C is the number of channels used
- A is the interval between advertisement transmission at a node
- N is the number of neighbors
- P is the packet deliver rate (1-PER)
- D is the duty cycle of the joining node

Decreasing A or increasing D requires more power, but the ultimate energy used in joining may actually decrease for faster joining periods. TSMP allows the user to turn off advertising entirely after all nodes have joined.

5.6. Service Requests

Whatever networking protocol is used, there is a trade-off between having readily available bandwidth and the cost to maintain it. In an LPL network [*e.g.* 8], shorter polling intervals are required to decrease the latency of allocating new communication between nodes which result in higher average power. In a TSMP network with time-varying traffic demands, our implementation requires nodes to request bandwidth from the manager. Based on the nature of the request, the manager will add in links as part of a redundant mesh, single chain, or anything in-between. Activating a service can occur in seconds; the network adapts energy use quickly to suit changing demands.

5.7. Health Reports

Independent of the data collection services operating in the network, each node periodically generates and sends a *health report* to the manager. This report summarizes MAC and NET statistics of used paths from this node which allow the manager to make beneficial changes in graphs and prepare for path failures by learning of new neighbors. Health reports include the number of MAC packets transmitted and failed, application-layer packets dropped, and battery life information. Health reports aid in traffic flow analysis and network diagnostics.

5.8. Neighbor Discovery

TSMP uses a periodic neighbor discovery process to learn about inter-node connectivity after a network has formed. During a prescribed slot, all nodes randomly choose to transmit a discovery packet or to listen. The transmission happens with low probability so often there are zero or one nodes transmitting. All nodes hearing a non-colliding transmission record and report the sender's ID and RSSI in the next health report. Maintaining this list of connections allows for network optimization and repair.

5.9. Security

TSMP has two security layers managed by a centralized application. The transport layer encrypts the application payload and authenticates the payload and network and transport headers. The DLL authenticates the entire packet or ACK. Keys are 128 bits, and use the AES-128 block cipher in CCM* mode. Prior to deployment, all nodes are assigned the Network ID and the cryptographic Join Key for the network they will join. The Join Key can be shared by all nodes or unique to each with the manager having an access control list of node IDs and keys.

During joining, the node and the manager authenticate each other through their shared knowledge of the Join Key. A node sends a join request message encrypted with its Join Key. If the Join Key and identity of the device are valid, the manager admits this node to the network, sends it a random node-specific Session Key and the shared

Table 1. Measured Atomic Radio On-Times for TSMP using an 802.15.4 PHY with 10ms timeslots

	Application Layer Payload		
Action	**0B**	**80B**	**95B**
TX	2.40 ms	4.96 ms	5.44 ms
RX	3.14 ms	5.70 ms	6.18 ms
Idle Listen	2.62 ms		

Network Key, both encrypted with the node's original Join Key. Thereafter, all traffic between node and manager is encrypted and authenticated at the transport layer with the unique end-to-end session key, and every hop is authenticated with the 4B MIC and Network Key.
Transport and application nonces contain both the source address and an increasing counter while the DLL nonce is formed by concatenating the ASN and the source address. Network-wide time synch allows the nonce counter to be omitted from the MAC layer header as only packets sent with the current ASN are accepted. Only the least significant byte of the transport nonce counter is sent. If this counter is outside a sliding window, the receiver discards the packet and notifies the next higher layer.
The MAC MIC protects the network from external attack, both malicious and random. A large fraction of packet errors are due to loss of blocks of many symbols. For these types of errors, CRC16 has a 2^{-16} probability of incorrectly identifying the garbled packet as valid. For 10% PHY PER, this bounds reliability in a multi-hop network to ~99.999% if the application can catch these errors. If not, they can be disastrous.

6. MEASURED PERFORMANCE

6.1. Atomic Radio Duty Cycles

The *atomic costs* of communication, in terms of radio on-time, are consistent across devices and same-length packets (Table 1). These values are used to compute the duty cycle of various functions. For example, one idle listen per 4000 slots increases the duty cycle by 2.62ms/4000s = 6.55×10^{-5}. The busiest possible node receives a 95B packet per slot for a 61.8% duty cycle.

6.2. Case Studies

In this section, TSMP's flexibility is shown with examples of real deployments using the same strategies. First, a base bandwidth is established for time synch and control packets, then services requested by nodes determine the final number of links between each node pair. Links can be quickly added, deleted, and (de)activated allowing the network to respond to changing demands. These particular case studies were chosen to show different extremes of radio range, data rates, reliability, and low power than can be achieved with TSMP. Unless otherwise specified, multi-hop mesh graphs to and from the AP are constructed to enable a base level of network maintenance. All packets are formatted with MAC and NET layer security and 10ms timeslots are used.

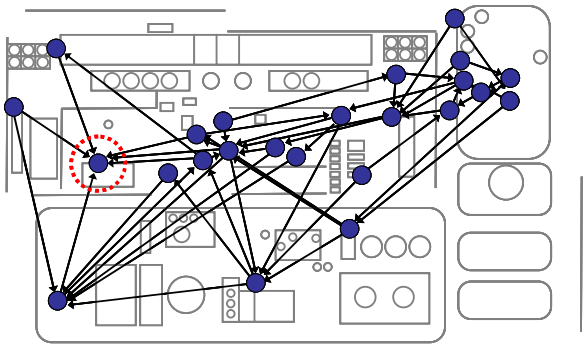

Figure 5. A snapshot of the multi-hop network topology from the refinery laid on top of a building plan. Arrows represent the network graph used to route data to the Access Point (circled). The network spans over 400m.

6.2.1. Coker Deployment - Range

The 24-node network shown in Figure 5 was installed in a double coker unit at a US refinery. The installation was performed per normal practices by a contractor with no wireless experience. As installed, with no configuration, the network achieved greater than 99.97% reliability over three months of testing. Many of the hops in this network were more than 100m, despite the challenging RF environment and a software-imposed lower bound on link margin of 85dB.

6.2.2. Print Shop Deployment - Reliability

A 44-node network was deployed at a printing facility. The facility was a 3-floor, 15,000 m^2 concrete and steel structure with substantial heavy machinery. During a 26-day network evaluation period, 3.6 million application-layer packets were generated and only 17 were lost. All this was achieved in the presence of path-channel stabilities that occasionally dropped to 0% for periods of one day. This means that all DLL packets between two nodes on a particular channel would fail, but the application-level packets were rarely lost since nodes maintained connectivity by automatically cycling through channels that had lower PER during these times.

6.2.3. "Dozer" Network – Low Power

Burri *et. al.* deployed a 40-node indoor network using their time-synched Dozer protocol [6]. In their analysis, the authors provide time-series data for the duty cycle of a node with 5 children and 13 descendants (node 114 in Fig. 9 of [6]). This node had an average radio duty cycle of 0.32%. We deployed a 14-node network and enforced the same topology beneath the node (shaded in Figure 6).

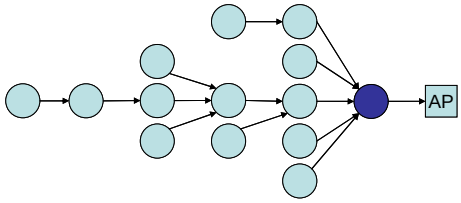

Figure 6. Sub-tree of the busiest node from [6].

With the parameters from Table 2 of [6] and a TSMP deployment, the busiest node has a mean radio duty cycle of 0.27% in steady-state with advertising deactivated. The leaf nodes, set to not route, have radio duty cycles averaging 0.02% (compared to 0.07% in the Dozer) from join time as they never advertise. Mean latency for the network is 56.4s with measured mean path stability of 93%. The qualitative difference between TSMP and Dozer is that empirically TSMP lost 0.1% of the packets generated, whereas in this example the Dozer mean loss was 1.2% and node 112 was excluded from the analysis because it had 30% loss. Additionally, TSMP showed a single node rejoin during the building phase compared to tens of connection attempts per node per day in Dozer. By deleting downstream listens at the leaf nodes, we have measured radio duty cycles of 0.01%.

6.2.4. Single-Node File Upload – Versatility

In this test deployment, 13 nodes are forced into a 13-hop deep topology. In addition to the regular superframe used for network functions, a 64-slot superframe with 4 links per hop is added for high-speed data transfer. At any given time, any single node *M* initiates a block transfer of 100 95B packets at a rate of 5 per second. The buffer between the 160ms inter-link interval and the 200ms packet generation interval is sufficient to not lose packets in the network which averages above 90% path stability. At this transfer rate, we measure 475B/s arriving at the AP from any single node in the chain without changing the link assignments.

The end node requires four upstream transmit slots and each intermediate node requires four upstream receives and transmits per superframe. When the end node is selected for the block transfer, it requires at most an additional 3.4% of radio on-time and the intermediate nodes an additional 7.3%. When unused, the source node does not require any additional radio on-time while the intermediate nodes increase 1.6% idling with this active superframe. This is compared to the near-100% radio on-time required for the network in [12] running at 441 B/s.

6.2.5. One-hop Network – High Throughput

TSMP also allows several nodes to have high data rates. Modifying a network for higher data rates throughout can be as easy as changing the number of links assigned to each node in the data collection superframe. For this experiment, we deployed 16 nodes within communication distance of the AP and formed a mesh network. The nodes were given an average of 40 links in a 10s superframe and allowed to generate data as quickly as they could successfully transmit it out. At the AP, 95B payloads were received at a rate of 47.3 packet/s for a total application layer throughput of 35.9 kbps. Reliability over the course of the deployment was >99.99%. In a network with this level of activity, synch info is always piggybacked on top of packet ACKs and no explicit keepalive or beaconing traffic is required.

6.3. Current TSMP Theoretical Limits

This section discusses the limits of TSMP as it pertains to current commercially available hardware.

6.3.1. Maximum Throughput – Single AP

TSMP could be used with a two-slot superframe chain to collect data rapidly from any single node in the network. Use the odd slot for the odd hops along the route and the even slots for the even hops. Different frequency offsets would be required to ensure that the simultaneous messages did not collide. For throughput it matters less how many hops are taken by the packet, but more the lowest path stability along the route. Since the manager has estimates of all path stability values, it can make the best choice for throughput. Conversely, latency is directly dependent on the number of hops in the chain. It is the mean path stability along the chain, S_{mean}, that factors into the mean packet latency = 20ms*hops/ S_{mean}. Again, the manager can choose the route that optimizes latency, or it could choose one that minimizes latency given a certain throughput requirement. Using two separate chains, the source node can send a 95B packet per slot. Of course, the AP requires some small fraction of the slots for network maintenance, but this scheme can deliver close to 76 kbps from a single node to the AP. A similar limit can be approached for services from different nodes.

6.3.2. Maximum Throughput – 16 APs

For a 16-channel spectrum, a manager could globally synch 16 different APs, each with its own cloud of nodes. Each cloud would then cycle through each channel evenly without colliding with traffic from others. This amounts to 1216 kbps of potential egress bandwidth to the manager from a single radio space. Alternatively, an AP built with 16 receive circuits capable of simultaneously receiving 16 packets could have the same performance.

6.3.3. Minimum Radio Duty Cycle

TSMP permits operation similar to that in a star-topology random access network. Consider a network of 500 nodes all within communication distance of the AP. If all nodes are given the AP as their only parent, power requirements can be kept very low while maintaining a low packet latency if data generation is sparse and uncorrelated. In a network of this type, beaconing for synch is more efficient. One unacknowledged beacon per 30 seconds requires a radio duty cycle of $<10^{-4}$ for the nodes to receive. On the AP side, all but the beaconing slots can be set to be Slotted Aloha shared by all 500 nodes. The nodes use energy in these potential transmit slots only when actually transmitting. Any node should be able to transmit a generated packet within one slot of its generation time, keeping latency very low. Additionally, a node missing a beacon can rapidly send an upstream packet to synch with an ACK.

6.4. Ultimate Limits of 802.15.4 Radios

One benefit of scheduled communication is that both sides of the transaction share substantial state information.

Like GSM or ATM cells, transmission of virtually all normal header information can be avoided for certain packet flows. For unicast links on a data flow, no part of the MAC or NET headers is actually necessary, and the packet format can be as shown in Figure 7. Both sides of each hop share the following state information: source and destination addresses, network ID, security information, and sequence number. All of this information can be included in a virtual header for MIC calculation: deleted from the packet by the transmitting node, and filled in at the receiving node. The virtual header ensures that communication is still secure and insensitive to random errors or protocol errors.

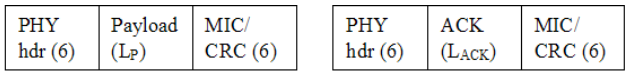

Figure 7. Minimal slot: 24B header and security total.

For tightly time-synched nodes, the minimum time slot for an acknowledged 802.15.4 packet could be as short as 24B of PHY header and DLL MIC/CRC plus the length of the payload and the ACK. With a 1B ACK, and removal of the now superfluous CRC, this allows 8B of payload in a 1ms time slot. For full-length packets, a 121B acknowledged, secure, application payload can be delivered in every 5ms time slot, or just over 190kbps. This represents a payload to radio bit rate ratio of 77%.

7. CONCLUSION

Deployments of networks using TSMP have shown that time synch enables better performance in low-power networks than has been achieved in asynchronous networking. At the MAC layer, by dedicating a small amount of radio time to explicitly synching the network, substantial amortized radio time is saved on each packet. At the network layer, synch allows channel hopping, which in 802.15.4 networks at least, provides links with higher stability thus requiring less network overhead to maintain connectivity.

Graph-based routing provides a mechanism for binding together routing and provisioning. By providing dedicated bandwidth for regular network operations, most commercial requirements are met. Bundling services into graphs that can be turned on and off according to need reduces network overhead. For those applications with unpredictable traffic patterns, shared slots can be used, with two advantages over non-synched approaches: long preambles are not required, and Slotted Aloha can handle twice the traffic of Aloha. TSMP operates at lower power in this mode than existing preamble sampling protocols, synched or not.

Centrally managed networks have some advantages. They can calculate optimal routes and guarantee collision-free traffic. TSMP provides mechanisms for synching nodes, assigning bandwidth, and providing network health information. None of these mechanisms is intrinsically biased to a centralized manager – simulations confirm the viability of distributed TSMP networks. However, the idea that distributed networks are required to hit the scale of deployments that are of commercial interest has been disproven. So far, centrally managed networks, and only centrally managed networks, have been shown to be capable of providing the highly reliable, low-power wireless connectivity necessary for mesh networking hundreds of nodes in industrial environments.

REFERENCES

[1] S.D. Glaser, R.A. Shoureshi, D. Pescovitz, Frontiers in Sensors and Sensing Systems, *Smart Structures and Systems*, *1*(1), Jan 2005, 103-120.

[2] http://www.hartcomm2.org

[3] S. Ganeriwal, R. Kumar, M.B. Srivastava, Time-Sync Protocol for Sensor Networks, *Proc. SenSys 03*, LA, CA, 2003.

[4] S. C. Ergen, P. Varaiya, PEDAMACS: Power Efficient and Delay Aware Medium Access Protocol for Sensor Networks, *IEEE Trans. on Mob. Comp.*, *5*(7), July 2006, 920-930.

[5] W. Ye, F. Silva, J. Heidemann, Ultra-Low Duty Cycle MAC with Scheduled Channel Polling, *Proc. SenSys 06*, Boulder, CO, 2006.

[6] N. Burri, P. von Rickenback, R. Wattenhofer, Dozer: Ultra-Low Power Data Gathering in Sensor Networks, *Proc. IPSN 07*, Cambridge, MA, 2007.

[7] J. Polastre, J. Hill, D. Culler, Versatile Low Power Media Access for Wireless Sensor Networks, *Proc. SenSys 04*, Baltimore, MD, Nov. 2004.

[8] M. Buettner, G. V. Yee, E. Anderson, R. Han, X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks, *Proc. SenSys 06*, Boulder, CO, 2006.

[9] L. Doherty, W. Lindsay, J. Simon, Channel-Specific Wireless Sensor Network Path Data, *Proc. ICCCN 07*, Honolulu, HI, 2007.

[10] R. Gummadi, D. Wetherall, B. Greenstein, S. Seshan, Understanding and Mitigating the Impact of RF Interference on 802.11 Networks, *Proc. SIGCOMM 2007*, Kyoto, Japan, 2007.

[11] J. Werb, M. Newman, V. Berry, S. Lamb, D. Sexton, M. Lapinski, Improved Quality of Service in IEEE 802.15.4 Networks, *Intl. Workshop on Wireless and Industrial Automation*, San Francisco, CA, 2005.

[12] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, M. Turon, Health Monitoring of Civil Infrastructures Using Wireless Sensor Networks, *Proc. IPSN 07*, Cambridge, MA, 2007.