The PEN Low Power Protocol Stack

Gray Girling, Jennifer Li Kam Wa, Paul Osborn
AT&T Laboratories Cambridge
24a Trumpington Street
Cambridge, UK

http://www.uk.research.att.com
{GGirling, JLi-Kam-Wa, POsborn}@uk.research.att.com

Radina Stefanova
Laboratory for Communications Engineering
Department of Engineering
University of Cambridge
Cambridge, UK

http://www-lce.eng.cam.ac.uk
Radina.Stefanova@cl.cam.ac.uk

***Abstract* - Low power consumption is a key design metric for wireless network devices that have limited battery energy. The problem of reducing power consumption needs to be addressed at every level of system design. This paper investigates the issues of designing low power protocols in the context of the PEN system, a mobile ad hoc radio network developed at AT&T Laboratories Cambridge. It describes the ad hoc protocols that have been implemented, outlining both the design of individual protocols and the structure of the overall stack. A summary of the lessons in low power design that were learnt is provided.**

I. INTRODUCTION

In a mobile embedded networking environment, the communicating nodes are small and rely on limited battery energy for their operation. Since energy is a limited resource, and battery technology has been slow to improve, the design of low power architectures and protocols has become a pressing issue. This paper investigates the issues of designing a low power protocol stack for the PEN[1] system, a mobile ad hoc radio network developed at AT&T Laboratories Cambridge [1]. A PEN is a network consisting of a set of autonomous embeddable, and possibly mobile, nodes that communicate intermittently. The maximum achievable bandwidth on such a network is quite low, making it more suited to control rather than data applications. To reduce power consumption, PEN nodes are powered down for much of their duty cycle and they awaken periodically to rendezvous with other nodes. This has interesting implications on the design of medium access control (MAC) and higher level protocols, requiring them to deal with the problem of nodes being powered down and thus unable to respond immediately.

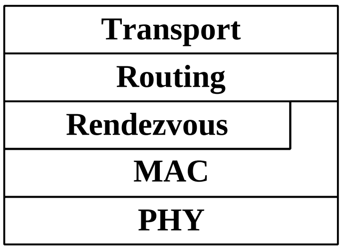

Fig. 1. The PEN Protocol Stack

In order to minimise power consumption and maintain the overall energy efficient properties of the stack, protocols should be as simple as possible. For this reason, existing de-facto protocols are not suited for the PEN environment. IP, for instance, is just too complex, particularly with respect to its routing functions; a typical implementation yields far greater code size and results in system demands that are too excessive for a node. The distributed co-ordination function (DCF) used by IEEE 802.11 [3] provides an extremely complex protocol for controlling a cluster of nodes, particularly when nodes join and leave a cluster. Bluetooth [2] suffers due to the demands of supporting asynchronous and isochronous traffic and so PEN should be able to benefit from the more relaxed requirements of control type applications. Although various energy efficient MAC protocols [6], [8], [9], [12] have been investigated, little has been done by way of energy efficient network protocols above the link layer [5]. The main contribution of this paper is to present an integrated solution with power saving features at various levels of the protocol stack.

Although they are by no means standard, the function of many of the layers in the PEN protocol stack are similar to those in the Open Systems Interconnection (OSI) seven layer protocol stack [4]. In particular the PEN MAC Layer has a similar function to the OSI Link Layer – to share the physical communication medium; the PEN Routing Layer has a similar function to the OSI Network Layer – to route over subnetworks; and, the PEN Transport Layer has a similar function to the OSI Transport Layer – to provide a required quality of service. One major difference, however, is the presence of an additional layer between the MAC and the Routing layers called the Rendezvous Layer which is responsible for scheduling and forecasting times of inactivity and thus has a major role in power saving.

Each protocol has been implemented in independent modules and is accessed through Application Program Interfaces (APIs) which are all supersets of the ones used to access the Transport Layer. This means that layers can be omitted from a system build should the function they provide not be required. For example the Routing Layer can be omitted when the system requires interaction only with local nodes or the Transport Layer can be omitted if large Service Data Units (SDUs) and data integrity are not required. Reduced software size can have a beneficial effect on power consumption if it results in a lower processor overhead for the periods during which a node is active.

---

[1] Prototype Embedded Network – this is the working title for the prototype Embedded Network previously known as "Piconet"

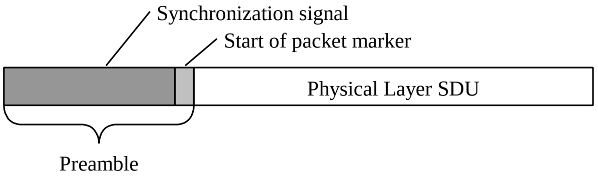

Fig. 2. Physical Layer PDU format

II. PHYSICAL LAYER

The PEN hardware uses an off-the-shelf Radiometrix "BIM-418" module operating at 418MHz to provide the radio interface. This is capable of a raw data-rate of 40kbit/s, half-duplex, but requires a balanced data stream so a 4 bit to 6 bit encoding is used, reducing the usable data-rate to around 25kbit/s. Unfortunately, the BIM provides no useful carrier sense, received signal strength or collision detection mechanism.

III. MAC LAYER

In addition to providing a communications service appropriate for the hardware available, the MAC protocol aims to do so in a manner that minimizes power consumption. Given that data is to be communicated, power can be saved only by mechanisms that attempt to minimise the amount of time that the radio has to be on. One such mechanism reduces the amount of radio traffic simply by choosing a time to transmit that minimises the likelihood of failure and therefore retransmission. Note that design choices in favour of such considered timing are likely to result in less aggressive protocols that would not meet a design choice in favour of high throughput or low latency. In this sense low power protocol operation must be traded against these other qualities of service.

Our MAC protocol uses a novel contention mechanism which bounds the probability of collision given likely communication scenarios and traffic levels. It is part of the philosophy behind PEN that communications should be possible when all nodes in range turn on their transmitters and receivers only occasionally. This effectively removes the possibility of exploiting continuous synchronization signals, such as might be required in a Time Division Multiple Access (TDMA) scheme. The contention mechanism is novel because it makes use of implicit synchronization signals inevitably created by the normal operation of the radio to schedule transmissions in a way that will result in fewer collisions than a pure Data Sense Multiple Access (DSMA) scheme, but more than a TDMA scheme. In a sense, the protocol is an intermediate hybrid between these two extremes.

The hardware prevents transmissions once a valid preamble has been received, but collisions can still occur during a preamble. Acknowledgements provide confirmation of at least one delivery instance, with each unacknowledged retransmission being delayed by successively longer times.

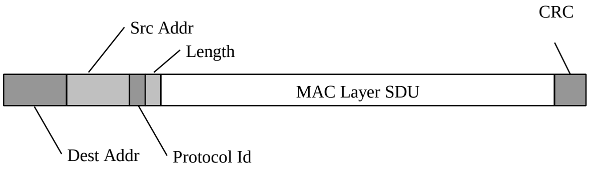

Fig. 3. MAC Layer PDU Format

When there are a number of nodes contending for the radio channel, they will probably all observe the same events. Hence an "end of transmission" event can be used to synchronize the contending nodes that observed it. To minimize collision following such events, the MAC protocol introduces a fixed contention period of c ms. Each node chooses a random time within the contention period to start its preamble of p (1.55) ms and does not collide with another preamble starting in that period with a probability of $(1+p/c)^{-1}$. Thus, when there are n contending nodes, the probability of no collision with the first preamble is $(1+p/c)^{-(n-1)}$, but if there is no collision the transmitting node will have been chosen fairly.

Power management involves the periodic shutdown of the radio transceiver. When power is re-established, it is often in order to perform a radio transmission. Because the radio can detect only data unit preambles, and not the data units themselves no transmission is allowed for the maximum packet transmission time. However, if a preamble is sensed and reported before the expiry of that time, it is possible to contend earlier for the channel along with the other users of the channel.

IV. RENDEZVOUS LAYER

The Rendezvous Layer encapsulates all the power saving decisions and timings independently of the MAC Layer. Since the rate of interaction is expected to be fairly low and a scheduled rendezvous can easily be negotiated when an ongoing association is required, the basic rendezvous scheme is designed for first contact scenarios.

A. Beacons

The basis for this design is introduced in [11] which explores the behaviour of two rendezvous systems: "server beaconing" and "client beaconing" as the rate of interaction increases. "Server beaconing" has the server node sleeping most of the time, but periodically waking up to advertise its presence by broadcasting a packet. Following this broadcast, the node will listen briefly for replies from interested clients. Any client wishing to use a particular server must switch on its receiver and listen for beacons from that server.

"Client beaconing" has the server listening continuously and clients transmitting requests as required. This works well

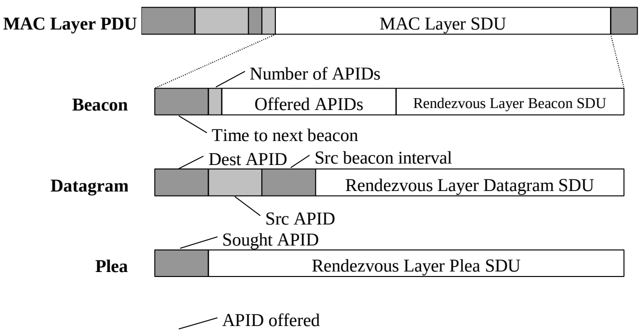

Fig. 4. Rendezvous Layer PDU Formats

TABLE 1
RENDEZVOUS LAYER TRANSMISSION MODES

Mode	Addressed by	Description
TX_SPECIFIC	MAC address and APID	Sends datagram to specified APID on specified node.
TX_ANY	APID	Sends datagram to first node seen offering specified APID.
TX_ALL	APID	Sends datagram to all nearby nodes offering specified APID.

when there are enough requests to justify the energy expenditure of the server, but in the anticipated low utilisation and low connectivity case of PEN it is too inefficient, so "server beaconing" is used.

"Server beaconing" is actually applied to any node that could accept Rendezvous Layer PDUs, whether or not it is a server. Such nodes will transmit periodic beacons using the MAC multicast service on a well-known address. Any node wishing to send Rendezvous Layer SDUs will listen for beacons from such nodes before sending.

Following the transmission of its beacon a node must listen for replies before powering down. This listen window must be long enough to allow all nearby nodes to send their replies, but at the same time be minimised to save power. To resolve these requirements a sliding window system is used. A minimum listen duration T_{min} is specified, along with a window duration T_{window}. The node will remain listening for at least T_{min} and until an interval T_{window} has passed with no further PDUs received.

B. Discovery and Description

Discovering nearby nodes that offer facilities with which one wishes to interact is important and must be power efficient. By having each node transmit a beacon containing a list of the services it offers, the processes of discovery and description are combined. This avoids the need to expend power querying the capabilities of a node after its presence is discovered. To minimise the size of the beacon PDUs, short identifiers called Application Protocol IDentifiers (APIDs) are used to differentiate between services. APIDs are similar to TCP port numbers in that they determine both the application and the protocol being communicated with.

Whilst idle the only protocol activity is waking up periodically to transmit a beacon, so if the interval between beacons is increased, so is battery life. However if the beacon period is too long, nodes wishing to communicate will spend longer waiting for a beacon and will consume more power; this also increases the latency of the communication. As a result there is a trade off between battery life and communication latency.

C. Application specified QOS

Different applications will have differing requirements for latency and battery life, so the selection of beacon interval is best left to the application. This also ensures that application developers are aware of the power decisions that they are making, leading to power efficient applications [7]. However, differing requirements must be reconciled when a node contains many applications. This is straightforward as the overhead in advertising multiple services in a single beacon is minimal, so only a single beacon is required[2]. The interval for this beacon should be the minimum of those requested by the applications. Since at least one application has requested this interval, there is a justification for expending the power needed to satisfy it, and all other applications will benefit from the reduced latency it provides.

D. Transmission Modes

The above beacon system offers a number of useful transmission modes when sending a datagram SDU. With the TX_SPECIFIC and TX_ANY modes, a node listens until a suitable recipient is found, often for significantly less than the APID's beacon period. The last mode (TX_ALL), however, has a serious power overhead because the sending node must listen for at least a full beacon period in order to hear all nearby nodes.

In all three modes, datagram SDUs are sent using a MAC unicast transmission, which provides automatic retries and an acknowledgement. When a transmission is pending, a node will actively listen until a suitable destination node is heard. This power consumption accounts for much of the energy consumed in Rendezvous communications. In certain situations "low priority" primitives that are cheaper in terms of their power usage may be desirable. For that reason a "best effort" flag can be set in SDUs sent to the Rendezvous Layer for transmission, indicating that the node should not expend power listening on behalf of this SDU. Frequently there will be other pending SDUs, which cause the node to listen, so "best effort" SDUs may still be successful. Continuously powered nodes incur no penalty for listening, so can treat "best effort" SDUs like any other.

E. Pleas and Offers

Nodes with large and cheap power reserves gain little by powering down. However, with the above algorithm a node wishing to send to a continuously powered node would still wait for it to broadcast a beacon.

By speculatively transmitting small "plea" PDUs when a datagram SDU is queued, any suitable recipient that is awake can respond with an "offer" PDU, allowing the datagram to be forwarded immediately. If no offer is received, the node falls back on waiting for a beacon; it can often power down sooner with this optimisation.

F. Results

A standard PEN node runs from a 9V lithium battery, chosen so as to maximise energy density. Whilst asleep the node consumes only 16µA, but this can rise as high as 90 mA when active. The addition of a power efficient idle task enables a node with an active transceiver to achieve an average consumption of 55 mA.

A node that does not use the Rendezvous Layer must keep its transceiver permanently active to receive packets.

---

[2] Unless the number of advertised APIDs is too great to fit within a single beacon, in which case a tiered beacon structure could be used, advertising high priority APIDs in one frequently broadcast beacon, and others in a less frequently broadcast beacon.

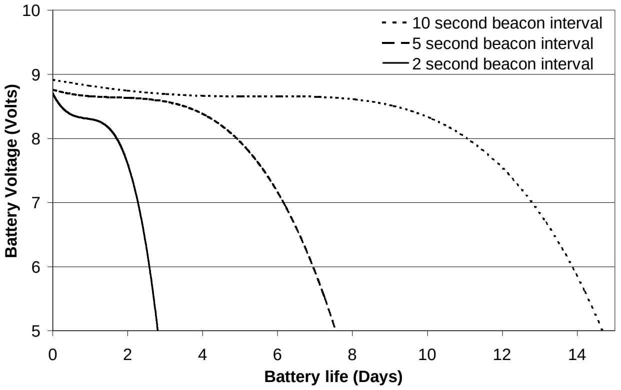

Fig. 5. Battery Life for various Beacon Intervals

Due to the design of the hardware this prevents the CPU from powering down, so such a node will drain its battery in approximately 20 hours. With the Rendezvous Layer it can power down, resulting in savings proportional to the time during which it is asleep. To demonstrate this, an experiment was set up where the variation of the battery voltage of a single node was measured for three different beacon intervals of two, five and ten seconds respectively. As Fig. 5 shows the battery life of a node is considerably extended when the Rendezvous Layer is in operation, and the longer the beacon interval, the longer the battery life is.

In the next experiment a single cluster of ten nodes was set up where all nodes are within range of each other. The nodes beacon at the same rate, and the mean Rendezvous SDU arrival rates are also equal. At each node a Rendezvous datagram with a payload of 200 bytes is generated at a chosen inter-arrival rate. All nodes support the same APID and the SDUs are sent using the TX_ANY primitive. One node in the cluster was picked at random and Fig. 6(a) shows its measured average current consumption over the course of the experiment. As expected the higher the data rate, the greater the average power consumption is. Fig. 6(b) shows that the data rate does not have much effect on the delivery success rate. Over anticipated datagram arrival rates about 90-95% of all the Rendezvous datagrams are delivered successfully.

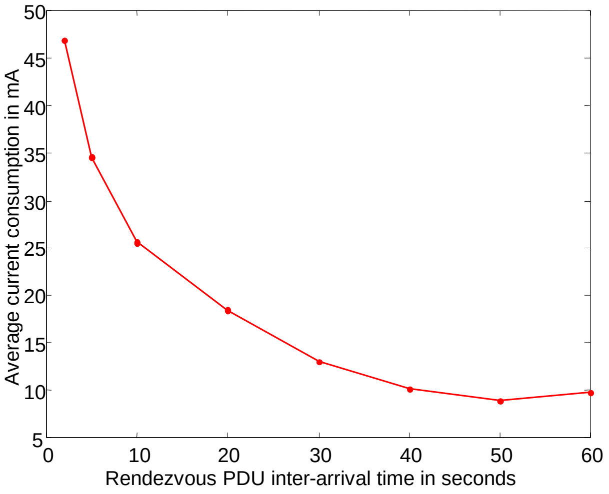

Fig. 6(a) Average current consumption per node

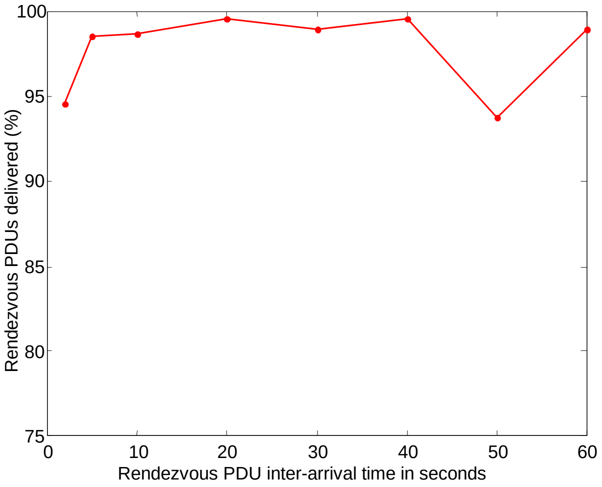

Fig. 6(b) Percentage of datagrams delivered

V. ROUTING LAYER

The Routing Layer provides the same layer service as the Rendezvous Layer, but can relay data over a number of intermediate nodes. Most importantly, the routing algorithm allows nodes to continue operating in a low power mode.

In addition to data delivery, the Routing Layer consists of two functions: route discovery and route maintenance. Routes are discovered on demand so that nodes do not have to expend energy discovering and maintaining routes they may never use. The penalty is a greater delivery delay when a node does not have a route to a destination as it must first perform the route discovery. Route maintenance also takes an on demand approach. A route is detected as broken because a node cannot use it, and only then are other nodes notified or the route fixed.

A. *Route Discovery*

When any transmission using the Rendezvous Layer is made, a far larger amount of energy is expended in synchronizing with the receiving node than in transmitting the routing PDU. This implies that the PDUs used in synchronizing nodes should be used to exchange as much routing information as possible. Beacon PDUs are used to carry two types of routing information: routes that the node needs and routes that it has available. On receiving a beacon PDU containing route requests, a node sends a gratuitous offer containing any of those routes in its routing table. This extra offer is the only protocol exchange that is not due to the normal operation of the Rendezvous Layer. On receiving an offer or a beacon PDU, a node updates its routing table with any new or better entries. Competing routes to the same node are compared by hop count.

Although discovered on demand, routes are cached for future use; this may cause a node to use a stale route, but considering the time and energy expense of discovering a new route it is better to rely on route maintenance to detect and fix broken routes.

B. *Route Maintenance*

The route maintenance procedure should detect a broken route and either attempt to route around the breakage or inform other nodes using it that the route is no longer valid.

Detecting a broken link is difficult because nodes are often powered down and the failure of a node to respond could mean that it has powered down, moved, or that the channel is busy. A node must detect reliably that a neighbour has moved out of range with as few false positives as possible because each time a broken link is detected route maintenance is invoked at one or more nodes. This is a waste of energy if the route is not broken and may result in a worse route being used (assuming the network has previously converged on optimal routes). While a node has its receiver on, it will receive a number of third-party MAC PDUs that would normally be discarded. However they contain the important information that their transmitter is within range. Using this information each node's Routing Layer maintains a list of all its neighbours and the time it last received a transmission from them, and this is used in determining whether a given neighbour has moved out of range.

The route maintenance procedure must be invoked when a neighbour has moved. If a node assumes that its neighbour has not moved, then the transmission must have failed because the two nodes failed to synchronize, probably due to radio interference. The routing PDU is then given another transmission opportunity but this time using the "best effort" facility. This allows a routing PDU to be retried without exerting any energy overhead. In fact the only extra energy that the node consumes is in re-transmitting the PDU.

If a particular route is found to be broken, the source of the failed packet must be notified. The routing PDU is marked as failed and returned along a route to its source if such a route is available. If at some point a failed PDU cannot be transmitted, it is discarded. This is a rather drastic approach as the source of a route may not be notified for some time that the route has broken, but the route maintenance overhead is minimised. Furthermore, as the PDU has just used the route, up to the broken link, that portion of the route is probably still intact, so this failure scenario is rare.

C. *Summary*

The Routing Layer service achieves low power operation by accessing details of lower layers that might otherwise have been unavailable. The first is the use of Rendezvous layer primitives (Beacon, Plea, Offer) to transfer routing information. The second involves the use of information gathered at the MAC Layer to make the routing protocol more robust.

D. *Results*

The following results have been obtained using a simulation of a PEN network that has been calibrated using empirical measurements. Fig. 7 shows the average energy consumption, i.e. the average amount of time spent powered on, per node in a network of ten nodes. The network is assumed to have a static topology and therefore the probability of a link is used to define the binomial probability of any two nodes in the network being able to communicate with each other directly. A probability of one corresponds to a fully connected network with all nodes within one radio hop and as the probability is decreased the network becomes increasingly more partitioned.

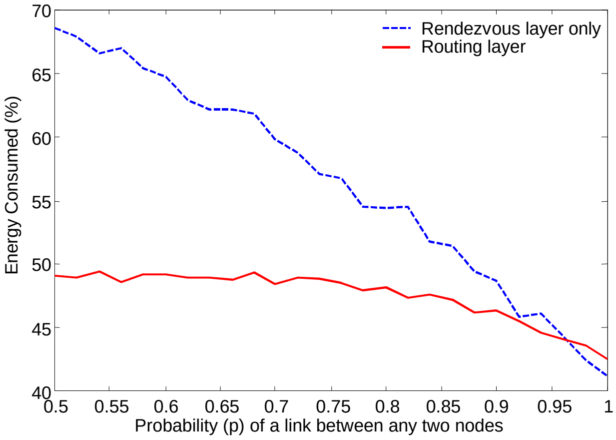

Fig. 7. Energy Impact of the Routing Layer

The graph shows that not only does the routing protocol not present a considerable power consumption overhead, but also that using routing actually results in lower power consumption in the majority of cases. This is most pronounced for the more sparsely connected network topologies.

These results should be considered in the context of the Rendezvous Layer: when a Rendezvous transmission is made, the node is kept powered on until it either synchronizes with the intended destination or the datagram times out. For a sparsely connected network the intended destination will not be within one radio hop of the source node and so it will remain powered on for the full timeout period before deciding that this PDU could not be transmitted. In the presence of the Routing Layer however, the node will attempt to discover a route to the destination and any discovered route is stored and used when needed. On average, the synchronization with a neighbour takes less than a beacon interval and so the node can transmit its datagram and power down. As the network becomes more reliably connected, the power saved by using the Routing Layer is reduced and for a fully connected network the Rendezvous Layer consumes less power than the Routing Layer. This is because in this case all nodes can communicate directly with any node in the network and a routing function is unnecessary. However, the overhead due to routing is relatively small precisely because the Routing Layer uses Rendezvous Layer SDUs to transport its routing information.

VI. TRANSPORT LAYER

The Transport Layer provides a data integrity service enabling reliable end-to-end transmission of application-specific Transport SDUs over the unreliable radio channel.

The protocol operates on a homogeneous network where the endpoints are PEN or PEN-like devices, and are addressed using a combination of MAC address and Transport Port number. To reduce energy consumption the protocol adopts a two, rather than three, phase connection approach. A two-way handshake is used to set up a temporary connection: the transmitter sends a connection request to the recipient specifying the desired Transport Port, and the recipient sends an acknowledgement confirming that

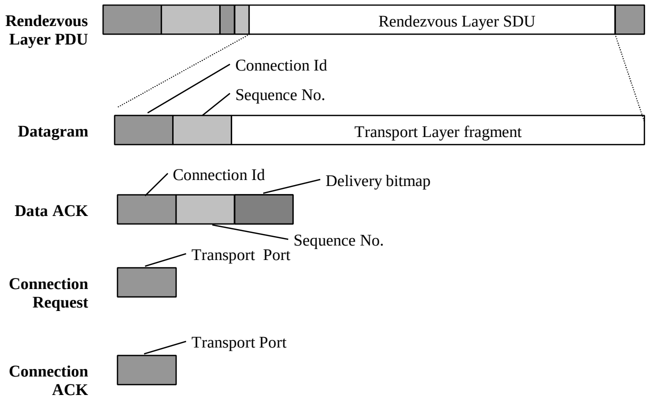

Fig. 8. Transport Layer PDU Format

it can receive on that port. The data transfer phase then proceeds, with the protocol performing the required segmentation and reassembly functions. In particular, the transport SDU is segmented into a number of transport datagram PDUs, which are then transmitted block by block.

Since PDUs can be lost or duplicated, the Transport protocol needs to ensure that the SDU is received intact at the recipient end. Each transport PDU is therefore assigned a unique sequence number. At the destination, the sequence numbers are used to identify and suppress duplicates. The recipient also maintains a bitmap of the sequence numbers of PDUs it has received so far; it periodically sends the transmitter an acknowledgement containing the sequence number of the first PDU being acknowledged and a bitmap indicating the receipt status for the next N PDUs (N being the bitmap size).

The acknowledgement is relatively infrequent (helping to reduce power use), but a transmitter cannot have more than a certain number (window size) of outstanding PDUs at a time. The window size is decremented each time a PDU is sent and reset when an acknowledgement is received. Such a mechanism provides a basic form of flow control and liveness control indication on a per connection basis.

As soon as a recipient receives the last PDU, it sends back an ACK to the transmitter and closes its end of the connection. On receiving that last ACK, the transmitter terminates its side of the connection. There is no explicit “closing” of a transport connection, thus avoiding the use of extra protocol primitives.

The transport service is advertised by a particular APID and an associated beacon interval. Typically the chosen transport beacon interval is large so that transport beacons are not sent too frequently. When a transmitter initiates a connection on a given Transport Port, it sends a connection request on the transport APID. A recipient that is listening on that Transport Port accepts the incoming connection request and sends a reply on the transport APID. For the data transfer phase, the transport protocol opens a transient APID with a smaller beacon interval. Using that APID transport PDUs are then transmitted as quickly as possible. Once all the PDUs have been sent, the transient APID is closed, and the Transport protocol reverts back to its normal beaconing mode.

VII. CONCLUSIONS

In this paper a MAC protocol was proposed that saves power by minimising the likelihood of collision and reducing the number of retransmissions. Above that, a Rendezvous protocol enables nodes to remain asleep for much of their duty cycle and switch on their transceiver only when it is required. An optional Routing protocol uses routing information opportunistically gleaned during lower layer protocol exchanges and provides a power saving even when used in a relatively well-connected network. A Transport protocol, built on top of these protocols, integrates with the Rendezvous Layer to provide energy efficient transfer of arbitrarily sized SDUs on an end-to-end basis.

Each protocol has some mechanism to reduce power consumption, and there is a strong requirement for applications to be power-aware. This underlines one of our main conclusions – that low power must be considered throughout an embedded system’s design. From the experience gained in the implementation of the PEN protocol stack, we have been able to identify the following power saving measures at each level of the stack:

Physical

- The majority of power saving is derived from the fact that the hardware can selectively power down different system elements.
- Protocols can sometimes require additional hardware features such as timer precision, more complex modulation techniques, frequency hopping etc, but omitting such features permits a lower power design to be used.
- Transmission is sometimes assumed to be more expensive in terms of power than reception. In the area of low power and low range radio, reception is likely to be at least as expensive as transmission. Identifying reception as an expensive activity has a significant impact on the protocol design.

MAC

- Because of the “hidden terminal” problem [10], the MAC must be designed without an absolute requirement for synchronization via radio signals. However, the probability that most of its users do observe the same events can be used to provide hints of channel availability.
- Delaying transmissions during period of high channel contention can save power by avoiding the need for retransmissions.

Rendezvous

- Protocols that establish periods of availability enable periods of radio dumbness and radio silence to be implemented relatively low down in a protocol stack. In lightly loaded nodes such periods can allow power to be removed in the rest of the system.

Routing

- Although this may break the ideals of “information hiding”, power can often be saved by promoting information to higher protocol layers that might otherwise be discarded. For example the Routing protocol exchanges route requests and routing information by using the Rendezvous Layer protocol primitives.

- Features such as low priority packets can avoid unnecessary power expenditure.
- In a radio environment with no global synchronization nodes expend a significant amount of energy in synchronizing with communication partners. Relaying through ubiquitous neighbours decreases the amount of time, and thus energy, necessary before communication can occur.

Transport

- An explicit three-phase connection approach is avoided. Additional protocol elements supporting more phases, particularly those only involved in maintaining the shared knowledge that a connection is in place, represent additional power use: not only in their transmission and reception; but also because their regular occurrence may prevent a node from powering down.
- A windowing protocol involving "lazy" acknowledgements saves power.

As a general conclusion, the modular implementation of the PEN stack has a role to play in reducing power consumption. This is especially relevant when the system becomes large and dictates the use of additional power-consuming hardware resources (e.g. external memory). Software extent and complexity may also require a greater level of activity in the hardware base (for example, coming out of low-power modes more frequently, or accessing relatively "power-expensive" memory). It is important, therefore, that the software incorporated into an embedded application can be tailored to match the application's requirements closely. This can most conveniently be achieved by constructing the system from largely independent software modules that can be assembled as required.

As mentioned before, the PEN nodes have been constructed from off-the-shelf components that have their limitations. Despite these limitations, our empirical measurements show that the above techniques have enabled us to extend the battery life of PEN nodes significantly. Further work is required to evaluate the effect that purpose built, more integrated, hardware would have on power consumption, both through a higher quality radio transceiver and through more efficient components. For example, higher bandwidth or faster processors – although they might require additional power – might enable a node's workload to be dispatched more quickly and thus enable it to power down sooner. Also the current Rendezvous Layer does not take advantage of the predictable nature of other nodes' availability when scheduling transmissions – modified protocol implementations that do are currently under investigation.

REFERENCES

[1] Bennett F, Clarke D, Evans J B, Hopper A, Jones A, Leask D, "Piconet – embedded mobile networking", IEEE Personal Communications, Vol 4 No 5, Oct 1997

[2] Bluetooth Specification Volume 1, Bluetooth Consortium, July 1999

[3] Institute of Electrical and Electronics Engineers Std. 802 and International Organization for Standardization, Std. ISO/IEC 8802 "Information Technology – Telecommunications and Information Exchange – Local and Metropolitan Area Networks – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications.

[4] International Organization for Standardization, ISO 7498-1, "Information processing systems – Open Systems Interconnection – Basic Reference Model", Oct 1984

[5] Lettieri P, Schurgers C, Srivastava M, "Adaptive link layer strategies for energy efficient wireless networking", Wireless Networks, Volume 5, pp 339-355, 1999

[6] Mangione-Smith W, Chang P S, "A low power medium access control protocol for multimedia systems", Proceedings of 3rd International Workshop on Mobile Multimedia Communications, Sep 1986

[7] Schlatter Ellis C, "The case for higher-level power management", Hot Topics in Operating Systems VII, 1998, Institute of Electrical and Electronics Engineers.

[8] Sivalingam K M, Srivastava M B, Agrawal P, "Low power link and access protocols for wireless multimedia Networks", Proceedings of IEEE Vehicular Technology Conference, May 1997

[9] Sohrabi K, Gao J, Ailawadhi V, Pottie G, "A self organizing wireless sensor network", Proceedings of 37th Annual Allerton Conference on Communication, Control, and Computing, September 1999

[10] Tobagi F A, Kleinrock L, "Packet switching in radio channels: Part II – the hidden terminal problem in carrier sense multiple-access modes and busy-tone solution", IEEE Transactions on Communications, Vol 23, No 12, pp 1417-1433, 1975

[11] Todd T D, Bennett F, Jones A, "Low power rendezvous in embedded wireless networks", Proceedings of 1st Workshop on Mobile Ad Hoc Networking and Computing, MobiHOC, August 11, 2000

[12] Zorzi M, Rao R R, "Energy management in wireless communications", Proceedings of 6th WINLAB workshop, New Brunswick, NJ, March 1997