

Low Power Rendezvous in Embedded Wireless Networks

Terry Todd¹, Frazer Bennett² and Alan Jones³

¹Department of Electrical and Computer Engineering
McMaster University
Hamilton, Ontario, Canada, L8S 4K1

²Wireless Technology
PA Consulting Group
Melbourn, Herts, UK

³AT&T Laboratories Cambridge
24a Trumpington Street
Cambridge, UK

Abstract—In the future, wireless networking will be embedded into a wide variety of common, everyday objects [1]. In many embedded networking situations, the communicating nodes will be very small and battery powered. For this reason, it is crucial that power consumption is as low as possible. A technique for reducing power consumption is to place nodes into a *sleep mode* whenever possible, and have them occasionally awoken to interact with other nodes. This type of action is referred to as a node *rendezvous*, and can be used in a variety of different ways.

In this paper we consider power-efficient service rendezvous in embedded wireless networks with external triggering. We first define two basic rendezvous mechanisms, namely, *server beaconing* and *client beaconing*. We show that server beaconing is preferred when the client arrival rate is below a parameter dependent threshold. Above this level, the use of client beaconing results in lower power consumption. We also consider a hybrid technique whereby server nodes independently select the beaconing mode so that total power consumption is reduced over a wide range of system parameter values. The operation of the client nodes is transparent to this selection.

We also introduce the use of *adaptive server beaconing*. In a static server beaconing system, the optimum beaconing rate is an increasing function of the client loading level. It is shown that by adapting the server beacon rate in an intelligent way, total power consumption can be greatly reduced over a large range of traffic loading conditions. A very simple method is introduced for performing this adaptation.

Several other innovations are discussed which can be used to reduce power consumption in embedded networks. We investigate the use of an AC mains-powered *rendezvous server* for power reduction, and we discuss a distributed power reduction technique referred to as *client beacon proxying*. It is shown that by performing rendezvous in an intelligent manner, total power consumption may be greatly reduced in many situations.

I. INTRODUCTION

In the future, embedded wireless networking will be included in many common, everyday objects. It is expected that this added functionality will enable an enormous variety of applications and services which have yet to be achieved or envisioned [1]. Possible examples of such embedded networks include sensor systems which monitor and control the environment, smart toys and games, smart appliances, electronic access control, and a myriad of other in-home, business, and manufacturing applications. At AT&T Laboratories Cambridge for example, the Prototype Embedded Networking Project (PEN) is a testbed for the provisioning of embedded wireless networks for many of these uses [1].

A unique aspect of embedded wireless networking is its

deemphasis on classical performance measures such as MAC channel capacity, and an increasing focus on low cost and reduced node power consumption. It should be noted that even in standard networks such as IEEE 802.11 and HIPERLAN, features have already been included which explicitly sacrifice capacity performance in favor of reduced power consumption [2], [3].

We can view embedded wireless networks to be configured in either *dense mode* or *sparse mode* applications. In sparse mode, the typical number of server nodes per wireless coverage area is very small. There are many applications of this kind, meter reading, temperature and heat sensors, are possible examples. Sparse mode systems consist of islands of service offerings whose performance is clearly not constrained by the available channel capacity.

In dense mode, there is typically a much larger number of embedded nodes per wireless coverage area. An example of this is a bookshelf where each book contains an embedded node which periodically announces the title of the book. In dense mode systems there are often opportunities for collaborative node operation. Even in dense mode systems however, many of the applications will be such that node interactions will be very short-lived and infrequent, so that channel capacity performance is not an important factor.

An important issue in embedded wireless systems is the selection of the objective function used as a basis for power optimization. In this paper we will employ an *average total power* (ATP) criterion, i.e., we will assess various systems on the basis of average total power dissipated. One can argue that this criterion approximates the cost of battery replacement, assuming a constant battery drain when the node is active. However, in many cases other criteria may be just as appropriate.

A technique for reducing power consumption this is to place nodes into a low power sleep (or doze) mode whenever possible, and have them occasionally awoken to communicate with other nodes [2]. This type of action is referred to as a node *rendezvous*, and can occur in a number of different ways. A *service rendezvous* is when nodes first communicate for the purposes of exchanging application information.

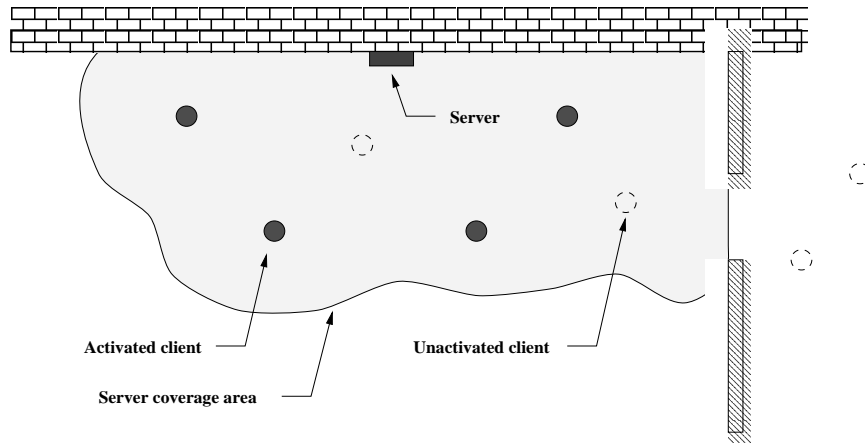


Fig. 1. Embedded Server System Model

Service rendezvous includes the act of determining a point in time where a desired node service is available so that it can be accessed. Once this happens, the nodes involved can engage in a *client/server session*. This involves the exchange of user information in accordance with the application that is being run. In many systems the session may consist of a very short interaction between the nodes involved. An example of this would be where a node queries and records the reading from a power meter node. Conversely, the session may involve a long-term association between the client and server. An example of this would be a wireless door lock/sensor which communicates over a long time period to a control panel in the same room. In this case, the session itself could involve repeated instances of node rendezvous for the purpose of occasional information exchange. This type of rendezvous is referred to as *session rendezvous*. The use of session rendezvous normally implies a relatively long-term association between the nodes in question. The IEEE 802.11 standard includes a session rendezvous mechanism where stations in sleep mode awaken simultaneously and listen to the channel for the duration of an ATIM time window [2].

In embedded systems of this type, it is often important to distinguish between service rendezvous and *service discovery*. A mobile node arriving at a new location may engage in a service discovery process where it learns of the services offered by nodes within its radio coverage area. Once it determines this information, it may decide to use those services that it is interested in. On the other hand, service rendezvous is the act of initiating a client/server interaction once the desired service is known to exist. In many applications, service discovery is inherent in the design of the system and the nodes are pre-programmed to access the service for which they are intended. In this case, service discovery is unnecessary and a client node directly initiates a service rendezvous to access the service. In systems of this kind, *beaconing* is a term used to describe a packet broadcast which is used by an awakening node to advertise or solicit an interaction between two or more nodes.

In systems with long-term client/server associations, the to-

tal energy dissipated due to service discovery and service rendezvous is likely to be very small compared with that dissipated due to the session itself. On the other hand, there are many applications for which the total power dissipated due to service rendezvous can be much higher than that associated with the client/server session. An example of this would be where a hand-held client node queries the contents of a shipping carton. In this case the actual transfer of application data may take just a few milli-seconds, whereas the service rendezvous may expend node energy for several seconds. In such cases there can often be many orders of magnitude difference between the rendezvous and session energy dissipations. In this paper we consider this latter type of system, i.e., where the energy expended due to service rendezvous is a significant fraction of the total. When this is the case, the techniques used for service rendezvous are very important from a power saving viewpoint.

To further reduce power consumption in embedded systems, nodes may use a form of *external triggering*. Consider for example, an application where an operator uses a wireless wand to read the output of a power meter. Rather than having the client node consuming power at all times, the operator may push a button on the wand to activate the node. Upon receiving this external signal, the node would power up, and engage in a service rendezvous with the power meter node. After downloading the pertinent information, the node would then switch its power off until the next activation trigger. Many other types of external triggers may be employed in such systems. Light sensing, magnetic induction, control-channel power sensing, and physical movement are a few examples of other types of external triggers. In this paper we will focus on systems which use external triggering to initiate service rendezvous.

There has been other recent work which considers the general issue of low power MAC protocols for wireless networks. In [4], techniques were considered for error recovery under different channel fading conditions. In [5] and [6], comprehensive studies of the power dissipation performance are given for various wireless ATM MAC protocols. Good discussions are

provided on how to reduce portable station power consumption under basestation control. In [7], a MAC protocol is proposed which uses reservation/scheduling for power reduction in the portable stations. Similar techniques have been proposed in [8], [9].

In this paper we consider power-efficient rendezvous in embedded wireless networks using the ATP criterion discussed above. We first define two basic beaconing mechanisms, namely, *server beaconing* and *client beaconing*. A simple system model is used to characterize the power performance for these classes of embedded networks. We find that when client nodes contend for a single service, server beaconing is the preferred mechanism when the client arrival rate is below a parameter dependent threshold. This result implies that it is important that the network operating characteristics are known sufficiently well in advance, if power efficient operation is desired. A hybrid technique is introduced whereby server nodes can independently select the mode of operation so that total power consumption is reduced over a wide range of system parameter values.

Adaptive Server Beaconing is also introduced in this paper. We show that by adapting the server inter-beacon times in an intelligent fashion, a server beaconing system can greatly reduce total power consumption over a large range of client arrival rates. A simple method is introduced for performing this adaptation.

Several other innovations are discussed which can be used to reduce power consumption in embedded networks. We investigate the use of an AC mains-powered *rendezvous server*. The rendezvous server provides timing information to arriving clients so that the power dissipated in performing the service rendezvous can be reduced. Our results show that this method performs extremely well across a wide range of client loading. When loading is very high however, lower power consumption is attained by client beaconing.

Finally, we discuss a technique referred to as *client beacon proxying*. This is a method where arriving clients co-operate to reduce total power consumption. By performing rendezvous in an intelligent manner, total power consumption may be greatly reduced in many situations.

II. EMBEDDED SYSTEM MODEL

In Figure 1 we show the system model used in this paper. We consider the case where an embedded node provides a service to a possibly large population of client nodes. We will assume a sparse mode situation where the coverage area of the server node does not overlap significantly with that of other server nodes. However, we will also present results which consider the system in the presence of contending node traffic. As discussed in the introduction, we assume that an arriving client is externally activated at some point after entering the wireless coverage area of the server. In the figure we have shown both unactivated and activated client nodes inside the server coverage area. Upon being activated, the client node would proceed

to initiate contact with the server node, i.e., perform a service rendezvous. As mentioned previously, it is assumed that the application is such that service discovery is not needed. Once the client node makes contact with the server, they engage in a client/server interaction session. We assume that following this exchange, the client node powers down and leaves the system.

Normally in such a system, one would specify a response time target in attaining the service rendezvous. In a sensor reading application for example, the design would try to ensure that under light loading conditions, the time to initiate contact with the server is bounded by some factor, denoted by D_{max} . Of course, when the loading is higher than this, users will experience a degradation in this performance, as is commonly the case in packet-switched systems. Also, because of channel errors, this bound is normally a statistical one.

III. CLIENT AND SERVER BEACONING

In this section we introduce two basic beaconing mechanisms, namely, Client Beaconing (CB), and Server Beaconing (SB). To aid in the description, we will focus on the actions of a particular server node and the client nodes which arrive to its radio coverage area. As discussed in Section II, we consider the common case where the visiting clients are enabled through external activation after having arrived at the coverage area. Accordingly, we assume that the system is designed so that service discovery is not required. The objective is to perform service rendezvous so that a service interaction can occur between each client and the server. Client and server beaconing both accomplish this in the following manner.

Server Beaconing (SB): Under the SB mechanism, the server sleeps and awakens periodically, every t_{SB} seconds. Upon awakening, the server sends a server beacon which is transmitted as a broadcast (or multicast) packet. These actions enable service rendezvous to occur and are shown in Figure 2. In the figure, a total of three server beacon transmissions are shown. Following each beacon, the server waits for responses from client nodes which may be waiting for service. An efficient mechanism for signalling this is to include a short response window (RW) at the end of each server beacon. Awaiting clients can transmit a short carrier burst into this RW to signal the presence of one or more awaiting clients. The absence of a response in this RW will cause the server to revert to sleep mode until the next scheduled beacon time. In this case the server only dissipates power for the duration of the beacon and its associated RW. The described interaction is shown in Figure 2.

An arriving client must first be externally activated. An example client arrival and its activation time are shown in the figure. An activated client node turns on its receiver and listens on the communication channel for the next server beacon. When the beacon is received it indicates the presence of the server, and the client proceeds by transmitting its request to the server. We refer to this as the client/server session as shown in the figure. In Figure 2 we have shown a single client and server packet

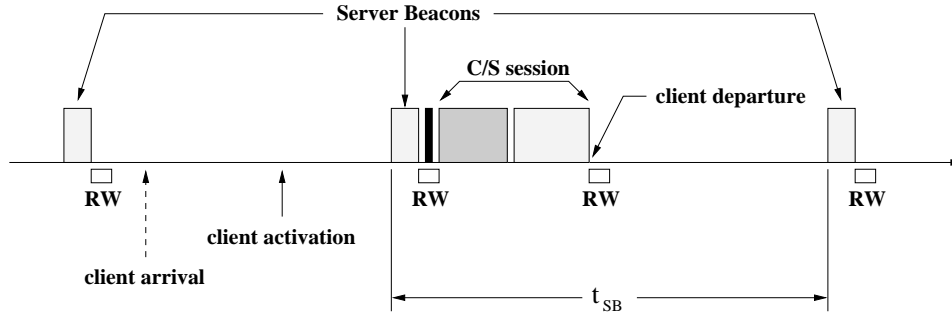


Fig. 2. Server Beacons Example

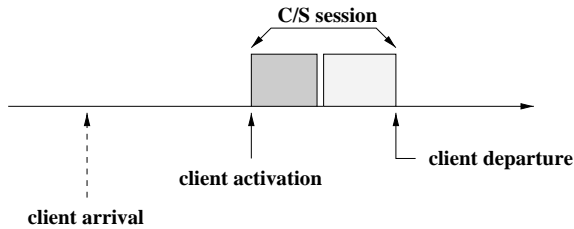


Fig. 3. Client Beacons Example

transmission but in practice the exchange may be more complex. This interaction is obviously application dependent and in some situations may involve a very long term association between the client and server. Conversely, in many systems the interaction may be very short and may conclude once the desired information has been returned from the server. In this case the power dissipated in the client/server interaction may actually be very small compared with that dissipated for service rendezvous. In the case shown, the client dissipates power from the time it is activated until the end of the client/server interaction. In the figure we have shown a RW at the end of the client/server interaction interval that the server uses to test for additional activated clients.

Client Beacons (CB): Under the CB mechanism, the server is operated so that its receiver is actively listening at all times (When the server is transmitting, its receiver is turned off, of course). Accordingly, an activated client node can immediately engage in the client/server interaction without waiting. This is shown in Figure 3. Since the server is always available, beaconing is not needed for service rendezvous purposes. Therefore, in systems without service discovery, client nodes can engage the server immediately. It should be noted that in systems with service discovery, the server in a client beaconing system may beacon for service discovery purposes.

It is obvious that in this scheme the server is constantly dissipating power. For this reason, this scheme may be more appropriate when the servers have large battery resources or are connected to AC mains power. However, we will see that under some conditions, the total client node power may be much lower than that of server beaconing due to a reduction in client receive power.

Note that all of the packets shown may be delayed due to channel contention. For simplicity we have not depicted this in the figures. It should also be noted that there are other more sophisticated types of client beaconing. These will be discussed in a future paper.

An important issue in the Server Beacons scheme is that of verifying that a beacon has been successfully transmitted. In normal unicast transmission, this is typically accomplished via an acknowledgement returned from the destination node. If the media access is based on a simple protocol such as CSMA, then a lack of response to the beacon can be interpreted in two ways. Either the beacon was lost due to a collision (or bit error), or, it may have been successful and there are no client nodes listening. In the first case, the beacon would ideally be re-transmitted until it is successful. Of course, when the server coverage area is disjoint from all others, then this is not an issue since the server is in explicit control of the channel and can transmit in a conflict-free fashion. When this is not the case, then a number of techniques can be used to reduce the likelihood of beacon loss. In this paper, the CSMA protocol prioritizes beacon transmissions over that of regular data packets.

As a final point, it should be noted that the “client” and “server” terminology we have used may be confusing in certain situations where nodes possess both types of functionality. More accurate terms might be “requestor” and “responder”. In the rest of the paper we will use the client/server terms.

A. Power Performance

In the embedded system model shown in Figure 1, client nodes arrive randomly to the server radio coverage area. Accordingly, we will assume that activated client arrivals appear at the server according to a Poisson arrival process with a mean of λ_C arrivals per second. As discussed previously, the statistical nature of the client/server interaction is application dependent. In this expository paper, we will assume the case where there is a single client/server interaction session between each client and the server node. Interactions of the type would be typical of many sensor and small database applications. We will assume that the nodes use p-persistent CSMA as the media access control (MAC) protocol [2].

A lower bound on system power performance can be obtained by ignoring the effects of propagation delay and MAC channel contention. We thus start by considering the system when perfect queueing is in effect. We will see that this actually results in a very tight bound on actual performance. This is because when radio coverage areas are in the 10's of meters in range, with a fairly modest channel bit rate, the normalized end-to-end propagation delay parameter, a , is expected to be very low. As a result, overheads associated with larger values of a are not present.

Under these assumptions, the Client Beaconing case can be modeled as a simple M/G/1 queueing system with client arrival rate, λ_C . A client's node power is dissipated from the time the node is activated until it has finished its interaction with the server. This includes the time spent waiting in the "distributed queue" of activated clients. For this reason, the M/G/1 service time of each client consists of the full duration of its interaction with the server. In Figure 3 for example, the service time of the client shown is the duration of the C/S session. The average total client energies dissipated due to transmission and reception over a long interval of length t are given by

$$\begin{aligned} E_{\text{tx}}^c(t) &= \eta_t \sum_{i=1}^{n_c(t)} T_{CS}(i), \\ E_{\text{rx}}^c(t) &= \eta_r \sum_{i=1}^{n_c(t)} (W(i) + T_{SS}(i)). \end{aligned} \quad (1)$$

where η_t and η_r are the energies dissipated per unit time when the node is transmitting and receiving respectively. $T_{CS}(i)$ and $T_{SS}(i)$ are the times to transmit the i^{th} client service and i^{th} server service packets respectively. $W(i)$ is the queue waiting time of client i .

Similarly, over a time period of t seconds, the energies dissipated by the server are given by

$$\begin{aligned} E_{\text{tx}}^s(t) &= \eta_t \sum_{i=1}^{n_c(t)} T_{SS}(i), \\ E_{\text{rx}}^s(t) &= \eta_r (t - \sum_{i=1}^{n_c(t)} T_{SS}(i)). \end{aligned} \quad (2)$$

In the above expressions, $n_c(t)$ is the number of client nodes processed in the time interval considered. By multiplying the above expressions by $n_c(t)/n_c(t)$ and dividing t , then taking the limit as $t \rightarrow \infty$, we easily obtain the following expressions for average power dissipated over all time.

$$\begin{aligned} \overline{P}_{\text{tx}}^c &= \eta_t \lambda_C \overline{T}_{CS}, \\ \overline{P}_{\text{rx}}^c &= \eta_r \lambda_C (\overline{W}_{M/G/1} + \overline{T}_{SS}), \\ \overline{P}_{\text{tx}}^s &= \eta_t \lambda_C \overline{T}_{SS}, \\ \overline{P}_{\text{rx}}^s &= \eta_r (1 - \lambda_C \overline{T}_{SS}). \end{aligned} \quad (3)$$

It will be assumed that the power dissipation of the node is the same when the node is awake and listening/receiving as when

it is awake and transmitting. This is roughly the case for the AT&T Research Cambridge Prototype Embedded Networking Version 3 nodes described in [1], and is typical of other commercial radio products. The results in this paper may be easily extended to accommodate unequal transmit/receive powers. Accordingly, we assume that $\eta_t = \eta_r = \eta$. In this case, the average power dissipated under Client Beaconing is given by

$$\begin{aligned} \widehat{P}_{CB} &= \eta(1 + \lambda_C(\overline{W}_{M/G/1} + \overline{T}_{CS} + \overline{T}_{SS})), \\ &= 1 + \overline{N}_{M/G/1} + \lambda_C(\overline{T}_{CS} + \overline{T}_{SS}). \end{aligned} \quad (4)$$

In the final expression we have normalized the power units so that $\eta = 1$, and defined $\overline{N}_{M/G/1}$ to be the mean number of clients waiting in the M/G/1 queue, i.e., $\overline{N}_{M/G/1} = \overline{W}_{M/G/1} \lambda_C$ from Little's Theorem.

In a similar way, for the Server Beaconing system we obtain a set of equations, as follows.

$$\begin{aligned} \overline{P}_{\text{tx}}^c &= \eta_t \lambda_C \overline{T}_{CS}, \\ \overline{P}_{\text{rx}}^c &= \eta_r \lambda_C (\overline{W}_G + \overline{T}_{SS}), \\ \overline{P}_{\text{tx}}^s &= \eta_t (\lambda_C \overline{T}_{SS} + T_{SB}/t_{SB}), \\ \overline{P}_{\text{rx}}^s &= \eta_r \lambda_C \overline{T}_{CS}. \end{aligned} \quad (5)$$

The total average power consumed in this case is

$$\begin{aligned} \widehat{P}_{SB} &= \tau_{SB} + \lambda_C(\overline{W}_G + 2(\overline{T}_{CS} + \overline{T}_{SS})), \\ &= \tau_{SB} + \overline{N}_G + 2\lambda_C(\overline{T}_{CS} + \overline{T}_{SS}). \end{aligned} \quad (6)$$

Here we have defined $\tau_{SB} = T_{SB}/t_{SB}$, which can be viewed as a normalized measure of the power dissipated in performing the server beaconing. Since we are considering the ideal queueing case, we have assumed that the response windows shown in Figure 2 are not present. In this case the system is not M/G/1, but is a queueing system with synchronous gating, whose mean waiting time is denoted by \overline{W}_G . In such a system, new arrivals wait until the next gating instant (i.e., the server beacon time). Once this time has passed, then the arrivals are queued for service. Such models are used to describe server systems where (as in our case) the server makes periodic reappearances and remains in the system until customer backlog is eliminated [10]. The queueing behavior of this type of system can be solved for a fixed client/server interaction time by embedding a Markov chain at the server beacon instants. We assume in this case that the client/server session time is fixed in duration, which is typical of many applications. This MC can then be solved by evaluating the set of b boundary conditions for the chain, where b is the number of client/server interactions possible within the chosen inter-beacon time, i.e., $b = (t_{SB} - T_{SB})/T_s$, where T_s is the total client/server interaction time. In this case, we have assumed that there are an integral number of client/server interaction intervals per beacon period, i.e., b is an integer.

A comparison of Equations 4 and 6 show that under conditions where λ_C is small, $\widehat{P}_{SB}/\widehat{P}_{CB} \approx \tau_{SB}$, and thus it is

typically the case that $\hat{P}_{SB} \ll \hat{P}_{CB}$. However, when λ is much larger, $\hat{P}_{SB}/\hat{P}_{CB} \approx \bar{N}_G/\bar{N}_{M/D/1}$. It is well known from queueing theory that the mean delay in an M/D/1 system is a lower bound for all Poisson arrival systems which schedule customers independently of their service requirements [10]. Using Little's Theorem we have that $\bar{N}_{M/D/1} < \bar{N}_G$ and therefore $\hat{P}_{SB} > \hat{P}_{CB}$ under heavy traffic conditions. Thus in the perfectly scheduled case, there is a crossover client arrival point where client beaconing is preferred.

A more detailed comparison of the two basic schemes is shown in Figure 4. In this figure we have plotted the normalized mean power dissipation as a function of total client arrival rate for both client beaconing and server beaconing for different values of t_{SB} . The curves were generated using the above models. In the curves we have normalized the time units to the total client/server session time, T_s . We have assumed that the server beacon is one-half of this value. This represents a fairly short interaction time which would be fairly typical of sensor-type applications. All other parameters are in terms of this normalization. In the figure we have also included simulation points showing the performance of the system when a CSMA MAC protocol is used for channel access. In these results, the 95% confidence intervals are within 3% of the plotted points. In Figure 4 power dissipation is normalized, so that a value of unity corresponds to a single node having its transmitter (or receiver) on continuously.

The first thing to note is that under low client arrival rates the SB schemes all give lower power consumption than does CB. At these values, system power consumption is dominated by that of the server alone. In the CB system the server is never sleeping so that the least power consumption possible is a value of 1, as shown at $\lambda_C = 0$. We also see that as t_{SB} is decreased, the slope of the power curve decreases, making smaller values of t_{SB} better for higher client arrival rates. This trend continues until we reach the point where the channel utilization of the system begins to suffer due to the overhead of server beacon transmission. This becomes significant when the inter-beacon times are within about an order of magnitude of the client beacon transmission time. For this reason, in the example given, there is little to be gained by reducing t_{SB} below the minimum value shown. As a result, under heavy client arrival rates, client beaconing is always preferred over server beaconing.

It can be seen that when queueing becomes a highly dominant power dissipation factor, the bound is not very tight since the effects of CSMA channel contention are not considered. In Figure 4 for the client beaconing case, at a client arrival rate of 0.6, the bound is about 14% lower than the actual power dissipated. The same effect obviously occurs for much larger values of power dissipation in the server beaconing cases. However, it is important to note that in the cases where the bound is not a very tight one, this corresponds to regions of very high power operation. These regions of operation are unlikely to be feasible in many low power embedded network applications. In

Section IV we will use this fact to propose a simple criterion for adapting the inter-beacon time in server beacon systems.

An important point to make in Figure 4 is that the advantage that SB shows under light client loading is a very important one. This is because it is expected that many embedded networks will operate under various client loading regimes. When deployed in a situation where the aggregate client loading is very low, it is highly desirable that the nodes can extend their battery life as much as possible. Operation in this region using an appropriate value for t_{SB} makes this possible.

As a final note, we can conclude from Figure 4 that client beaconing is only preferred in situations where clients are dense and provide relatively high total aggregate loads. In many low-end applications this would not be the case.

The comparison of server and client beaconing is markedly different when the client/server interaction times are comparable to, or larger than the inter-beacon interval. This case corresponds to the situation where the association time between client and server is much larger than the case considered above. In this case, the average power dissipation is dominated more strongly by the service times of the clients instead of by power dissipated in performing the service rendezvous function. Also, the overhead associated with beaconing itself is not a significant power consumption term. Under light loading the server beaconing systems are superior as before. However under heavier load there is less distinction between server and client beaconing since power consumption is dominated by session queueing and service. In this case server beaconing is generally preferred over client beaconing in all regions of client arrival rate.

It is expected that in future embedded networks, there will be many applications for which each of the above situations hold. In many sensor applications for example, it is expected that $t_{SB} \gg T_s$. In many office situations where a mobile node arrives and remains for many hours, then the total client/server interaction time may be very large. In this paper we consider the former case, that is, where the power consumption due to service rendezvous may be large compared to that of client/server interaction.

Note that in the above comparisons we have assumed that there is a single interaction between the client and server. This, of course, is application dependent and there may very well be a long-term association between the client and server where the two nodes awaken and communicate many times.

We have also compared these systems when there is significant background traffic within the coverage area of the server. In the interests of brevity these results are not included here. When there is a significant increase in background traffic, the same general trends exist as in the previous case. Client beaconing is shown to perform significantly better under heavy arrival rates where the power dissipation is high.

Finally, it is important to note that in certain situations, the preferred beaconing mechanism is obvious. For example, when the server node is connected directly to the AC

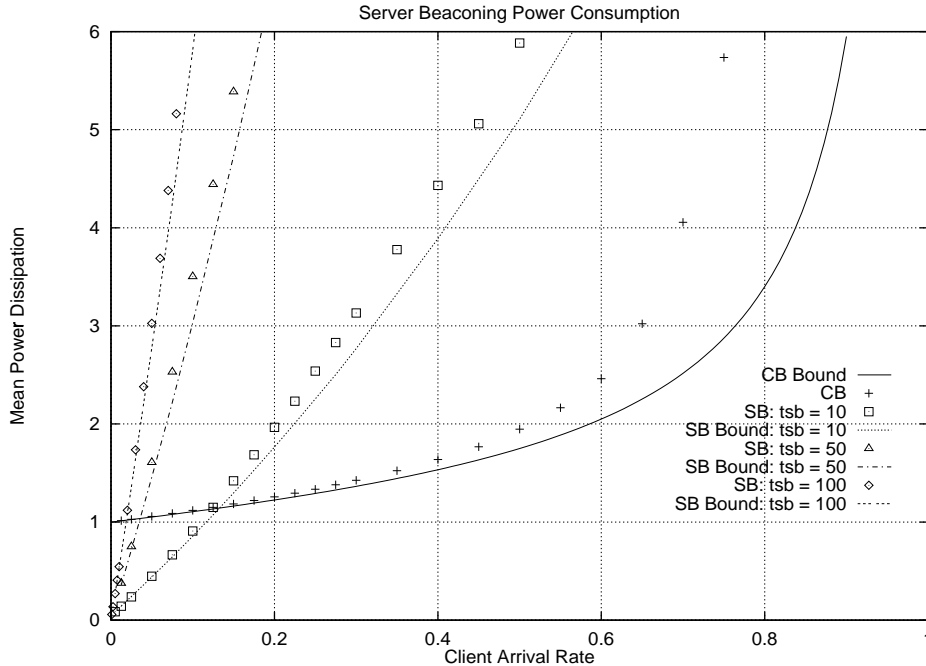


Fig. 4. Server and Client Beaconsing, $T_s = 1$

power mains, then the ideal scheduling model gives $\hat{P}_{CB} = \lambda_C(\overline{T}_{CS} + \overline{W}_{M/D/1})$ and $\hat{P}_{SB} = \lambda_C(\overline{T}_{CS} + \overline{W}_G)$. In this case $\hat{P}_{CB} < \hat{P}_{SB}$ and thus client beaconsing is always superior to server beaconsing since the server is being powered “for free”. Conversely, when the clients are mains-powered, we have that $\hat{P}_{CB} = 1$ and $\hat{P}_{SB} = \lambda_C\overline{T}_{CS} + \tau_{SB}$. In this case $\hat{P}_{CB} > \hat{P}_{SB}$ and server beaconsing is always preferred. An example of the first case is in the Olivetti Active Badge System [11]. In this system, battery-powered badges awaken and transmit a 12 ms IR beacon every 10 seconds. The badges communicate with AC mains-powered nodes whose known physical location is used to track the location of the mobile nodes. In this system we can view the fixed nodes as providing a location registration service to the mobile client nodes.

B. Hybrid Server and Client Beaconsing

In Section II we found that when client/server interaction times are small, SB is heavily preferred under light to moderate client loading, and that CB may be preferred otherwise. This raises the question of how a system should operate whose loading characteristics are either unknown in advance, or varies widely over time. In this section we consider a very simple scheme whereby a server node independently selects the mode of operation to use, in a way that is transparent to the client nodes.

Hybrid SB/CB: The intent of this scheme is to operate as a client beaconsing system under heavy client loading, and as a server beaconsing system under light loading. The scheme is such that the server can set the mode of operation dynamically and transparently with respect to the arriving clients.

Ordinarily, an activated client would have to know the mode of operation so that it knows whether to wait for a server beacon or not. In the hybrid scheme, all activated clients transmit their beacon immediately. If the server is currently in CB mode, then the server will respond and the client/server interaction proceeds. If the server is currently in SB mode, then the client will not receive an immediate response to its beacon. In this case the client then remains with its receiver on, listening for a server beacon. When the server beacon occurs, the system proceeds as in a conventional server beacon system. Deadlocks may occur if, as an example, a client misses the final server beacon just before the server switches from SB to CB mode. To prevent this, a client will timeout and re-beacon if it does not hear the server beacon within a period of time equal to t_{SB} . This behavior is also needed for recovery in the event of a server response which is lost due to channel contention or bit error.

Using Equations 4 and 6 from Section II, it can easily be shown that under light loading conditions, the ideal queueing CB and SB curves intersect when

$$\lambda_C \cdot t_{SB} \approx \frac{t_{SB} - T_{SB}}{t_{SS} + t_{CS} + t_{SB}/2 + T_{SB}}. \quad (7)$$

Using this approximation, a server node can select the mode of operation by making a local measure of the current client loading. A simple heuristic is to apply SB when the measured loading is less than the crossover predicted by Equation 7, and to use CB otherwise.

In Figure 5 we show an example of the mean power consumption of this hybrid system where the server uses the num-

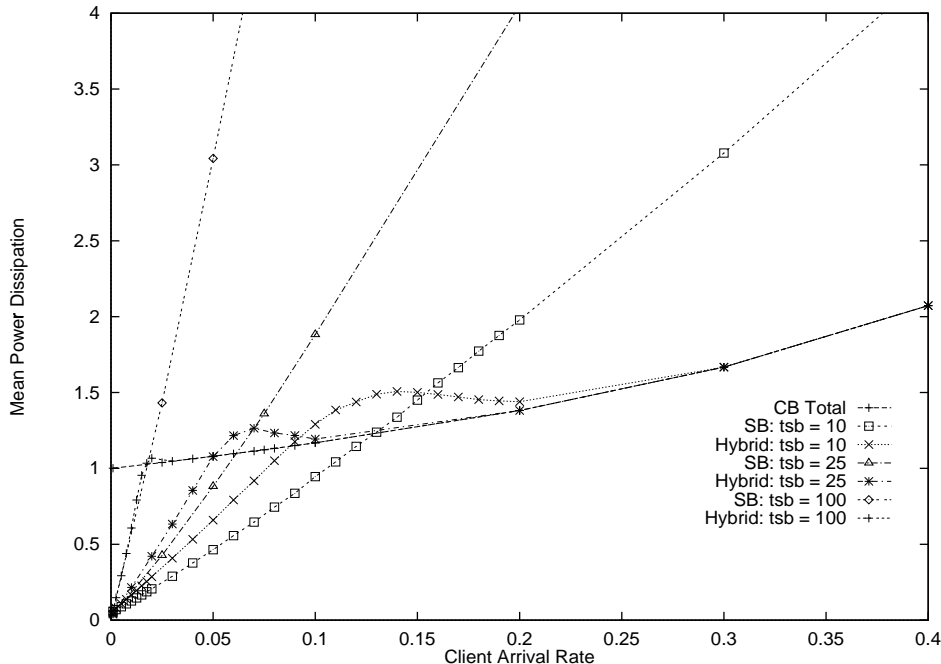


Fig. 5. Hybrid SB/CB Scheme

ber of clients served over a window of 10 beacon periods to estimate the client load. The mode is then switched, if indicated. In the graphs we show the CB and SB systems for 3 different values of t_{SB} . We have also plotted the power consumption of the hybrid scheme. We have made these comparisons over a wide range of parameter values, and this graph is representative of the results we have found.

In general we find that the system approximates the best of the SB and CB curves. This approximation is better for larger values of t_{SB} where the loading crossover point is at lower values of client arrival rate. Note that in the hybrid scheme client request packets are always transmitted twice when the server is in SB mode. For this reason, when the crossover point occurs at higher arrival rates, these additional transmissions contribute more significantly to the total power consumed. This accounts for the significantly poorer power consumption approximation in the $t_{SB} = 10$ case. However, the scheme is generally useful as an adaptive technique which achieves better power consumption over a wider range of parameter values than either SB or CB alone. However, it should be cautioned that the loading crossover points may be at relatively large values of client arrival rate, which may limit the usefulness of this scheme in many applications.

C. Transmit-Only Servers

In this section we briefly introduce a variation in the above system which may be used in certain very low power, low cost situations. In some applications the amount of information provided by an embedded server node may be very small. In addition, this information may be provided free to any interested

node, so that identification and authentication of clients is not needed. An example of this might be a node with an attached temperature sensor. In this case, there may be no need for any interaction on a client by client basis and the current temperature value could be simply included in the payload of the server beacon packet. In this case there may be no need for a receive function at the server, from an application standpoint. In this case it is possible for the node to be deployed with a radio transmitter only. We refer to such a node as a *transmit-only server* (TOS). Obviously one must be careful in this type of situation since normally a node receiver is required in order to properly engage in channel access at the MAC level. Clearly one would never deploy such nodes where traffic conditions could develop to the point where channel contention adversely affects the operation of the system. This may not be an issue in many cases because of the small radio coverage areas and the low transmit duty cycle of such nodes. In the example given, with a 10 second inter-beacon time and a 10 ms beacon packet, the duty cycle is only 0.1%.

In a system with a single server of this type, a successfully transmitted beacon can service all awaiting clients. In this case, using the above power model we have that, $\overline{P}_{tx}^c = 0$, $\overline{P}_{rx}^c = \eta_r \lambda_C (t_{SB}/2 + T_{SB})$, $\overline{P}_{tx}^s = \eta_t \tau_{SB}$, and $\overline{P}_{rx}^s = 0$.

In Figure 6 we have plotted the power performance of this system compared with client beaconing. The client beaconing case is the same as before where clients must individually solicit responses from the server. We have assumed that the server beacon in the TOS system has a transmission time equal to the session time of the server beacon system, which is being very generous.

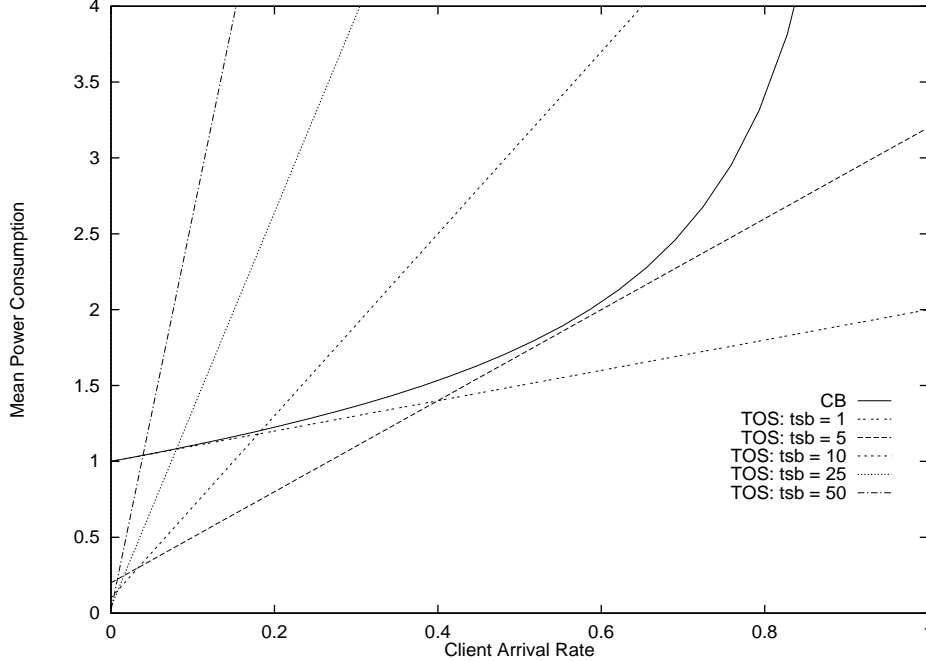


Fig. 6. Transmit-Only Server (TOS)

It can be seen that in this case there is no power advantage in client beaconing regardless of loading. Even for very high values one can always choose an inter-beacon time so that power dissipation is lower for the TOS. This is reasonable since the TOS system has eliminated the individual client/server interaction sessions. For this reason, the dominant term is that due to client receive power. However, the choice of inter-beacon time is a difficult one for this type of system. In the next section we will discuss how inter-beacon times in server beaconing systems can be adapted for power reduction. Unfortunately, this is difficult for a transmit-only server since there is no direct way for the server to determine the conditions that it is operating under.

IV. ADAPTIVE SERVER BEACONING

In Figure 4 we saw that as t_{SB} decreases, the slope drops for the SB power consumption curves. A cursory look at this figure might suggest that there is no advantage to using larger values of server inter-beacon time. This of course is not a correct conclusion because of the behavior of the curves at very low values of client arrival rate. Although it cannot be seen in Figure 4, the curves all cross at some point, the ones for higher values of t_{SB} having lower power-axis intercepts. For this reason, using a very large value for t_{SB} may be highly desirable to ensure a very long battery lifetime for a server node which happens to be deployed where the client arrival rate is small. Another example would be where at night, the client arrival rate drops to zero for many hours. This situation also suggests the use of a large value of t_{SB} over this period. Conversely, Figure 4 shows that as the arrival rate increases, very

large values of t_{SB} lead to excessively large values of power consumption. Clearly, a rational strategy would be to provide for adaptation of the server inter-beacon time so that this period decreases in situations where the client loading is higher. In this section we consider the simple use of this technique and show that it can greatly reduce power consumption in many situations.

It can easily be shown that for a server beaconing system, the average power dissipated is approximately given by

$$\hat{P}_{SB} \approx \frac{\overline{N}_{SB}t_{SB}/2 + (\overline{N}_{SB}^2 + \overline{N}_{SB})T_s/2 + \overline{N}_{SB}T_s}{t_{SB}} + \tau_{SB}, \quad (8)$$

when $\overline{N}_{SB}T_s \ll t_{SB}$, where \overline{N}_{SB} and \overline{N}_{SB}^2 are the first two moments of the number of clients found waiting at the start of the inter-beacon interval. Using the Poisson arrival assumption and the fact that the inter-beacon interval is fixed in length, this expression can be re-written as follows,

$$\hat{P}_{SB} \approx \frac{\lambda_C t_{SB}}{2} + \lambda_C T_s \left(2 + \frac{\lambda_C t_{SB}}{2}\right) + \tau_{SB}. \quad (9)$$

In Equation 9 the contribution of each term is clear. The first is for clients awaiting the server beacon. The second term is due to the client/server session, and the last term is the power dissipated due to server beaconing alone. It can be easily shown that this function is convex and has a global minimum at $t_{SB} = t_{SB}^*$, where

$$t_{SB}^* = \sqrt{\frac{2T_{SB}}{\lambda_C(1 + \lambda_C T_s)}}, \quad (10)$$

$$\approx \sqrt{\frac{2T_{SB}}{\lambda_C}}.$$

This result suggests a very simple way of performing the beacon rate adaptation. The server continuously monitors the value of λ_C , then simply uses Equation 10 to compute the proper inter-beacon time to use. When applying this scheme, the server would set the inter-beacon time to $t_{SB} = \max\{t_{SB}^*, \mathbf{D}_{max}/2\}$, so that the adaptation always tries to meet the \mathbf{D}_{max} delay constraint.

In Figure 7 we have plotted a set of normalized curves which compare the total average power dissipated for various values of server inter-beacon time compared with that of the adaptive scheme using the above approximation. In the curves, both the client arrival rate and t_{SB} are normalized to the client/server interaction time. As before we have assumed that the server beacon transmission time is 50% of this value.

Consider the following example which illustrates the power saving which is possible using server inter-beacon adaptation. Consider a an embedded node which is acting as a server for a temperature sensor. Let us assume that we would like to satisfy an application response time of $\mathbf{D}_{max} = 10$ seconds. This means that when client loading is light, then from the time that a node is activated, we would like to have the initial rendezvous occur within 10 seconds. In addition, we would also like to minimize the power dissipated when the node is not experiencing client arrivals. Accordingly, we set $t_{SB} = 10$ seconds so that an activated client must wait at most this long to hear the server beacon. For this example we will choose a client/server interaction time of 10 ms which might be typical for this type of application. To use Figure 7 we first determine the normalized value for t_{SB} , which is 10 seconds divided by 10 ms, which gives us 1000. Looking at the extreme left of the figure using the $t_{SB} = 1000$ curve we see that the normalized average power dissipation is about 0.0005, which is very low. However, now consider what happens when a node using this inter-beacon period is used in a system where the client arrival rate is slightly higher than this, at say, a value of one client arrival every 100 seconds. This is clearly still a fairly low client arrival rate. The normalized value of this client arrival rate is $(100/10\text{ms})^{-1}$ or 10^{-4} . From Figure 7 we see that at this arrival rate and server inter-beacon time, we are dissipating an average power of 0.05. However at this value of client arrival rate it can be seen that if we were to use the adaptive scheme and decrease our inter-beacon time, we would actually improve our time responsiveness, and our power dissipation would be only 0.01, i.e., failing to adapt the server inter-beacon time has resulted in a power dissipation which is 500% higher than necessary. It can be seen from the curves that if the arrival rate increases by another order of magnitude, then the fixed inter-beacon system will be dissipating over 1400% more power than the adaptive scheme. It is clear that in this type of system there are substantial gains to be made by adapting the inter-beacon times so that power dissipation is reduced for the operating environment considered. This example also

illustrates the negative consequences of a greedy power allocation mechanism. The static setting used initially is the one that minimizes power in the server, yet results in a much higher client power dissipation than is necessary.

In Figure 7 we have also shown the curve resulting from the use of the adaptive scheme suggested above. It can be seen that total power consumption using this technique tracks the lower limit of the power curves shown for the non-adaptive server inter-beacon values. The power savings possible using this method can be readily seen from the graph.

Finally, it should be noted that a form of adaptive beaconing is used in the Olivetti Active Badge System [11] referred to in Section III. In this system the badges normally beacon every 10 seconds. However, the front of the badge is fitted with a light-dependent resistor which reduces this rate to effectively zero when in darkness. This reduces the power consumption of the badge in many situations when it would not normally be used, such as during the night, or when enclosed in a briefcase. It is expected that this type of on/off adaptation may be very useful in many embedded radio applications.

V. RENDEZVOUS SERVER

In some embedded networking situations it is possible that there may be one or more AC mains-powered nodes within the coverage area of a battery-powered server node. When this is the case, the mains-powered node can be used to reduce the total power consumption in the system. We refer to this node as a *rendezvous server* (RS). It is possible to do this in a way that is transparent to the server node.

The rendezvous server listens to the radio channel continually and tracks the inter-beacon times of the server or servers in question. To enable this, the server beacon packet would contain the scheduled time of the next server beacon. The rendezvous server caches this information along with the identity and other appropriate details of the server node that it is functioning on behalf of.

An example of rendezvous server operation is described as follows. When a client node arrives, it initially functions as in the hybrid server/client beacon system described in Section III-B. It starts by transmitting its request to the server node. This request is intercepted by the rendezvous server which replies back to the client node. This reply includes a field which contains the time until the next scheduled server beacon instant. Upon receiving this information, the client node can resume sleeping until the next server beacon time. The client awakens just before the scheduled beacon time for the server.

An inefficiency in this scheme is that the rendezvous server may intercept the client packet even though the server node is still listening. A more efficient scheme would be to prioritize the server response over that of the rendezvous server so that the latter only responds if the server node is currently sleeping.

In Figure 8 we show an example of the power savings possible using the rendezvous server. In this figure we have taken Figure 7 and added the rendezvous server curve. In a design of

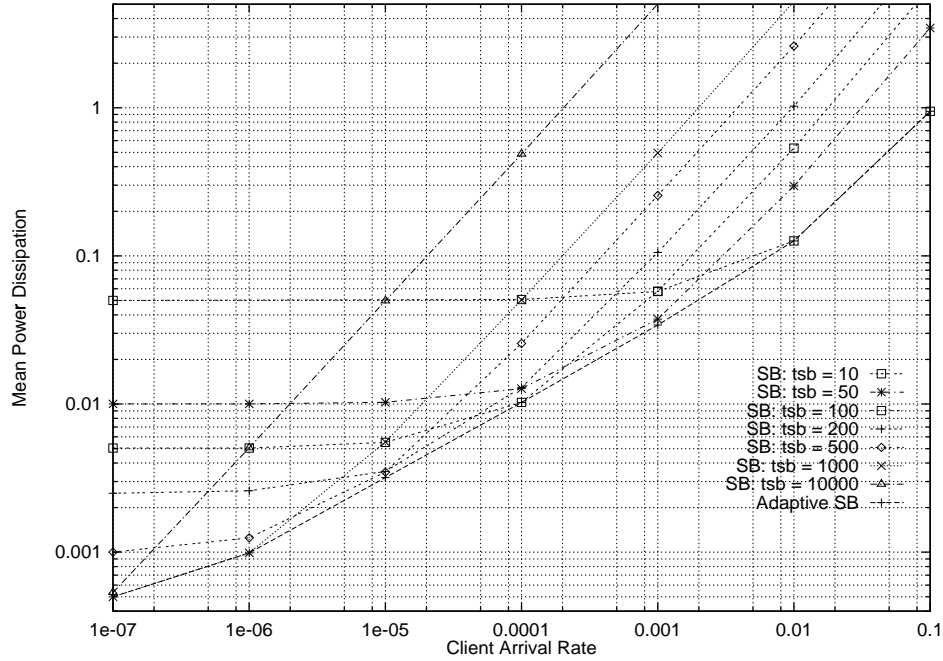


Fig. 7. Adaptive Server Beacons

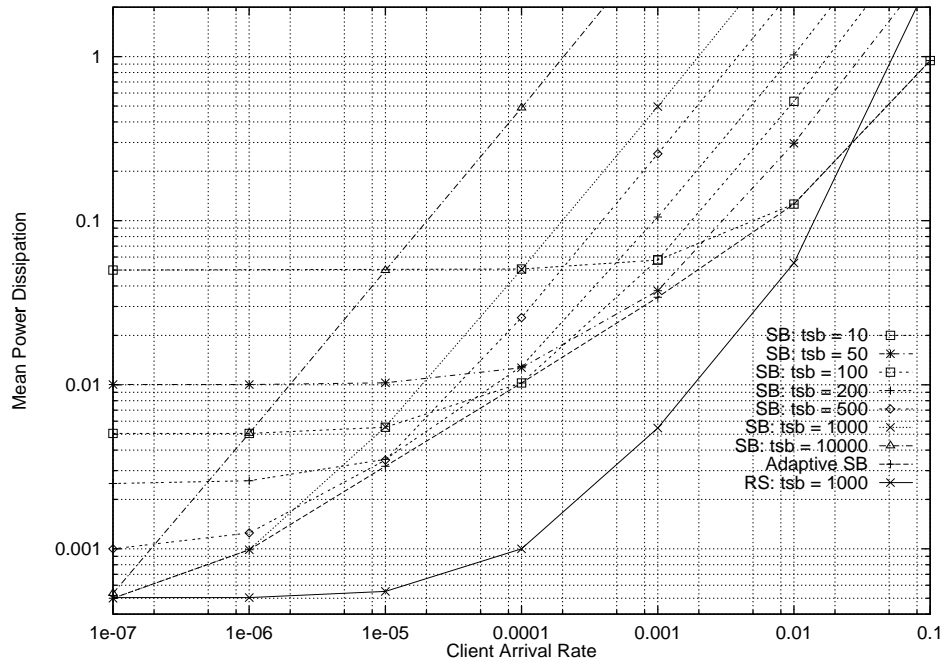


Fig. 8. Rendezvous Server System

this type one would set the server inter-beacon to the maximum possible, subject to the desired access time constraint. In the example curve shown we have used the value of $t_{SB} = 1000$ which we quoted in the example given near the end of Section IV. It can be seen from Figure 8 that very large power savings over a wide range of client loadings can be obtained using this technique. At the right of the graph, the power sav-

ings are eventually lost as indicated in Figure 8. It is clear however, that if the use of a rendezvous server is an option, it should be considered. An important design issue in this case is to ensure that the nodes have the needed time-base accuracy to perform the rendezvous functions described.

Note that the good power performance of RS does not continue for all values of client loading. As seen in Figure 8,

eventually at very high arrival rates the power consumption increases and this system is outperformed by basic client beaconing. The reason for this effect is that in the basic scheme outlined above, each client request is possibly transmitted twice, i.e., once to the RS and once to the server. As a result, when these requests are a significant fraction of the client/server session time, this overhead limits the capacity of the system.

VI. CLIENT BEACON PROXYING

In this section we briefly discuss a simple technique for power conservation referred to as *client beacon proxying*. This is a collaborative method where arriving client nodes exploit each others presence to reduce power consumption. There are many options. We will give an example of one possibility.

Arriving nodes operate as in normal hybrid client/server mode. The server beacon rate is set to twice the requirement dictated by desired response time considerations. As in hybrid client/server beaconing, when client nodes are activated they beacon immediately. If there is no response to the beacon, the client proceeds as in hybrid client/server mode and remains with its receiver on. It becomes the "proxying node". While waiting in this mode, if it sees another client beacon, it will respond and the two nodes arrange to rendezvous, t_{SB} seconds after the initial arrival of the proxy node. This permits the second node to immediately enter sleep mode, while the first remains active, waiting for the server beacon. Other arriving nodes work in the same way. Note that the rendezvous time is the same for all nodes served by the proxying node. Once the server beacon occurs, the active client synchronizes to its beaconing. This information is related to the other client nodes at the time of rendezvous.

Various options and enhancements are possible. An unfortunate aspect of this scheme is that higher reliability dictates that the first node be the one which remains active throughout the waiting period. This is to help ensure that all sleeping nodes were within reception range of the proxy node at the time of activation.

VII. CONCLUSIONS

In this paper we have considered power-efficient rendezvous for embedded wireless networks. We first defined the server and client beaconing rendezvous mechanisms. Under a total power dissipation criterion, client beaconing is preferred when the system loading exceeds a parameter dependent threshold. When loading values are below this level, server beaconing results in lower power consumption. We investigated the use of a hybrid technique whereby server nodes can independently select the beaconing mode of operation. We find that this method works quite well especially when the inter-beacon times are large. However, it should be noted that the crossover points tend to occur at relatively large values of client loading when the inter-beacon times are small. For this reason, the use of this technique is probably restricted to high power dissipation regimes.

In this paper we also introduced the use of adaptive server beaconing. In a static server beaconing system, the optimum inter-beacon rate is an increasing function of the client loading level. For this reason power savings can be made by properly adapting the inter-beacon rate as loading conditions change. Very large savings in power consumption are possible using this technique. This is a very useful mechanism since it is expected that embedded nodes will be deployed in situations where there may be many orders of magnitude difference in the operating conditions encountered. Also, loading conditions may vary greatly over time. A very simple method was introduced for performing this beacon rate adaptation based on having the server node track the current client loading level.

Several other innovations were also discussed which can be used to reduce power consumption in embedded networks. We investigated the use of an AC mains-powered *rendezvous server* for power reduction. It was shown that a rendezvous server can dramatically decrease the average power dissipated under low power client loading situations. Finally we discussed a distributed power reduction technique referred to as *client beacon proxying*. It is shown that by performing rendezvous in an intelligent manner, total power consumption may be greatly reduced in many situations.

VIII. ACKNOWLEDGEMENTS

The authors would like to thank Jennifer Li Kam Wa and Paul Osborn for discussions and input to this paper.

REFERENCES

- [1] F. Bennett, D. Clarke, J. B. Evans, A. Hopper, A. Jones, and D. Leask. "Piconet: Embedded Mobile Networking". *IEEE Personal Communications*, 4(5):8–15, October 1997.
- [2] IEEE Standards Department. "*IEEE Draft Standard - Wireless LAN*". IEEE Press, 1996.
- [3] C. Rohl, H. Woesner, and A. Wolisz. "A Short Look on power Saving Mechanisms in the Wireless LAN Standard Draft IEEE 802.11". In *6th WINLAB Workshop on Third Generation Wireless Systems*, March 1997.
- [4] A. Chockalingam and M. Zorzi. "Energy Consumption Performance of a Class of Access Protocols for Mobile Data Networks". In *VTC'98, Ottawa, Canada*, May 18-21 1998.
- [5] K. M. Sivalingam, M. B. Srivastava, and P. Agrawal. "Low Power Link and Access protocols for Wireless Multimedia Networks". In *VTC'97 Phoenix, AZ*, May 4-7 1997.
- [6] J. Chen, K. M. Sivalingam, M. B. Srivastava, P. Agrawal, and S. Kishore. "A Comparison of MAC Protocols for Wireless Local Networks Based on Battery Power Consumption". Submitted for publication.
- [7] K. S. Natarajan. "A Hybrid Medium Access Control Protocol for Wireless LANs". In *Proc. IEEE Intl. Conference on Selected Topics in Wireless Communications*, 1992.
- [8] T. Imilienski, S. Vishwanathan, and B. R. Badrinath. "Energy Efficient Indexing on Air". In *Proc. SIGMOD*, May 1994.
- [9] G. Sfikas, C. Apostolas, and R. Tafazolli. "Power Saving Mode of Operation in the WATM MAC Protocol". In *International Conference on ATM, ICATM'98, Colmar, France*, June 22-24 1998.
- [10] J. F. Hayes. *Modeling and Analysis of Computer Communications Networks*. Plenum Press, 1984.
- [11] A. Harter and A. Hopper. "A Distributed Location System for the Active Office". *IEEE Network*, 8(1), January 1994.