Chapter 1

Performance of bridging algorithms in IEEE 802.15.3 multi-piconet networks

Jelena Mišić, Muhi Ahmed Ibne Khair, and Vojislav B. Mišić

University of Manitoba, Winnipeg, Manitoba, Canada

In this chapter, we introduce the bridging problem from the viewpoint of the recent IEEE 802.15.3 high data rate WPAN, and present alternative solutions that are possible in 802.15.3 networks. Then, we investigate the performance of a network with two piconets interconnected in a parent-child manner and linked through a bridge device which operates in a master-slave fashion. We have designed an adaptive bandwidth allocation algorithm for bridge down link CTA allocation, and examined the impact of the value of the smoothing constant and threshold hysteresis on the throughput, blocking probability, and average queue size for the downlink queue at the bridge.

1.1. Introduction

The IEEE 802.15.3 standard for high data rate Wireless Personal Area Networks (HR-WPANs) is designed to fulfill the requirements of high data rate suitable for multimedia applications whilst ensuring low end-to-end delay.[1] Devices in 802.15.3 networks are organized in small networks called piconets, each of which is formed, controlled, and maintained by a single dedicated device referred to as the piconet coordinator (PNC). However, there are many applications in which a multi-piconet network must be used: for example, small scale mesh networks[2] could be implemented using 802.15.3 standard. 802.15.3 networks can also be used in multimedia sensor networks, where their high data rate ensures reliable transmission of still images or video feeds from the sensors, e.g., in surveillance or military applications.

The 802.15.3 standard provides the basic piconet interconnection capability in the form of parent-child piconet topology, in which the time on the working channel is shared among the two piconets. However, the standards does not give much guidance as to the actual algorithm for topology construction, not does it provide the algorithms for bandwidth allocation which is needed to achieve the desired performance levels of a multi-piconet network.

In this chapter, we investigate the performance of a two-piconet network in a parent-child topology where the child piconet coordinator acts as the master-slave bridge to allow communication between the two piconets. We describe two adaptive algorithms for bandwidth allocation and examine their performance through extensive simulation. We show that the second algorithm provides much better performance despite its relative computational simplicity.

The chapter is organized as follows. In Section 1.2 we describe the major characteristics of the 802.15.3 standard, and highlight the way in which multi-piconet operation may be achieved in Section 1.3. In Section 1.6, we examine the performance of fixed bandwidth allocation and describe an adaptive algorithm with symmetrical thresholds, which is then modified to include hysteresis in Section 1.7. Finally, Section 1.8 concludes the chapter and provides some directions for future research.

1.2. Operation of 802.15.3 networks

The IEEE 802.15.3 standard describes the requirements for physical (PHY) and medium access control (MAC) layer protocols for a high data rate wireless personal area network. Its high data rate and low latency make it suitable for multimedia applications, but also ensure easy reconfigurability and high resilience to interference, since it operates in the unlicensed Industrial, Scientific, and Medical (ISM) band at 2.4GHz. (Note that this band is shared with a number of other communication technologies such as 802.11b/g WLANs, 802.15.1 Bluetooth piconets, 802.15.4 low data rate WPANs, and others.)

An 802.15.3 network is formed in an ad hoc fashion: upon discovering a free channel, the PNC capable device starts the piconet by simply transmitting period beacon frames; other devices that detect those frames then request admission, or association (as it is referred to in the 802.15.3 standard). The coordinator duties include transmission of periodic beacon frames for synchronization, admission of new devices to the piconet, as well as allocation of dedicated time periods to allow unhindered packet transmission by the requesting device.

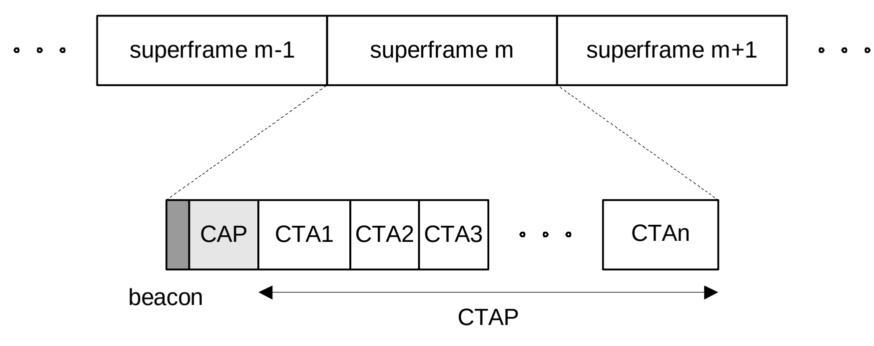


Fig. 1.1. Superframe structure in an 802.15.3 network.

Time in an 802.15.3 piconet is structured in superframes delimited by successive beacon frame transmissions from the piconet coordinator. Each superframe contains three distinct parts: the beacon frame, the contention access period (CAP), and the channel time allocation period (CTAP), as shown in Figure 1.1. During the Contention Access Period, devices compete with each other for access; a form of CSMA-CA algorithm is used. This period is used to send requests for CTAs (defined below) and other administrative information, but it can also be used for transmission of smaller amounts of asynchronous data.

The Channel Time Allocation Period contains a number of individual subperiods (referred to as Channel Time Allocation, or CTA) which are allocated by the piconet coordinator upon explicit requests by the devices that have data to transmit. Requests for CTAs are sent during the Contention Access Period; as such, they are subject to collision with similar requests from other devices. The decision to grant the allocation request or not rests exclusively with the piconet coordinator, which announces its decision in the next beacon frame; CTA allocation may be temporary or may last until explicit deallocation by the piconet coordinator. Once a device is allocated a CTA, other devices may not use it, and contention-free access is guaranteed. Special CTAs known as Management Channel Time Allocation (MCTA) are used for communication and dissemination of administrative information between the piconet coordinator and its devices.

Unlike other WPANs such as Bluetooth and 802.15.4, direct device-to-device communication is possible in an 802.15.3 piconet. In case the communicating devices are not within the transmission range of each other, the piconet coordinator (which, by default, must be able to communicate with both) may be involved as an intermediary, leading in effect to multi-hop intra-piconet communication. It is worth noting that problems of this nature may be alleviated by adjusting the transmission power, but also by making use of the adaptive data rate facility provided by the 802.15.3 standard. Namely, if transmission at the full data rate of 55 Mbps suffers from too may errors because the signal-to-noise-plus-interference ratio (SINR) is too low, different modulation schemes with lower data rate may be used to give additional resilience. This problem and its solutions, however, are beyond the scope of the present chapter.

Reliable data transfer in 802.15.3 networks is achieved by utilizing acknowledgements and retransmission of non-acknowledged packets. The standard defines three acknowledgment modes:

- no acknowledgement (No-ACK) is typically used for delay sensitive but loss tolerant traffic such as multimedia (typically transferred through UDP or some similar protocol);
- immediate acknowledgement (Imm-ACK) means that each packet is immediately acknowledged with a small packet sent back to the sender of the original packet; and
- delayed acknowledgement (Dly-ACK), where an acknowledgment packet is sent after successfully receiving a batch of successive data packets; obviously, this allows for higher throughput due to reduced acknowledgment overhead – but the

application requirements must tolerate the delay incurred in this case, and some means of selective retransmission must be employed to maintain efficiency.

1.3. Interconnecting IEEE 802.15.3 piconets

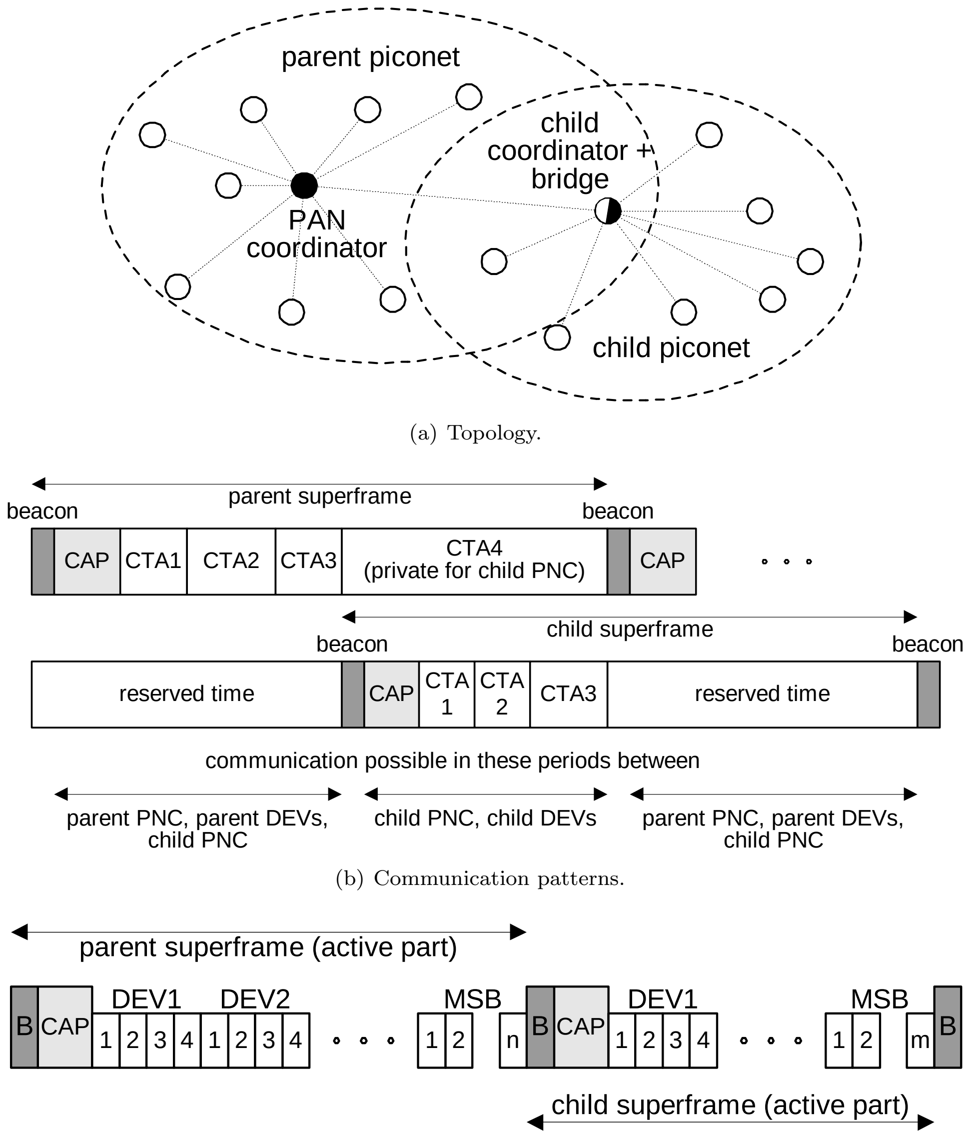
The 802.15.3 standard contains provisions for the coexistence of multiple piconets in the same (or partially overlapping) physical space. Since the data rate is high, up to 55 Mbps, the channel width is large and there are, in fact, only five channels available in the ISM band for use of 802.15.3 networks. If 802.11-compatible WLAN (or, perhaps, several of them) is/are present in the vicinity, the number of available channels is reduced to only three in order to prevent excessive interference between the networks adhering to two standards. As a result, the formation of multiple piconets must utilize time division multiplexing, rather than the frequency division one, as is the case with Bluetooth. Namely, a piconet can allocate a special CTA during which another piconet can operate. There are two types of such piconets: a child piconet and a neighbor piconet.

A child piconet is the one in which the piconet coordinator is a member of the parent piconet, as shown in Figure 1.2(a). It is formed when a PNC-capable device which is a member of the parent piconet sends a request to the parent piconet coordinator, asking for a special CTA known as a private CTA. Regular CTA requests include the device addresses of both the sender and the receiver; a request for a private CTA is distinguished by virtue of containing the same device address as both the sending and the receiving node. When the parent piconet coordinator allocates the required CTA, the child piconet coordinator may begin sending beacon frames of its own within that CTA, and thus may form another piconet which operates on the same channel as the parent piconet, but is independent from it. The private CTA is, effectively, the active portion of the superframe of the child piconet. The child superframe consists, then, of this private CTA which can be used for communication between child piconet coordinator (PNC) and its devices (DEVs); the remainder of the parent superframe is reserved time – it can't be used for communication in the child piconet. Figure 1.2(b) shows the communication patterns in this topology.

The timing relationship of superframes in parent and child piconets is shown in Figure 1.2(b), where the top part corresponds to the parent piconet and the bottom part to the child piconet. Note that the distinction is logical rather than physical, since the piconets share the same RF channel.

A given piconet can have multiple child piconets, and a child piconet may have another child of its own. Obviously, the available channel time is shared between those piconets, at the expense of decreased throughput and increased delay; but the effective transmission range may be increased.

Challenges. As can be seen from the discussion above, the main challenge in forming a multi-piconet network that uses the same RF channel – i.e., a complex



(a) Topology.

(b) Communication patterns.

(c) Superframe structure. MSB denotes the CTA allocated to the MS bridge.

Fig. 1.2. Parent-child interconnection.

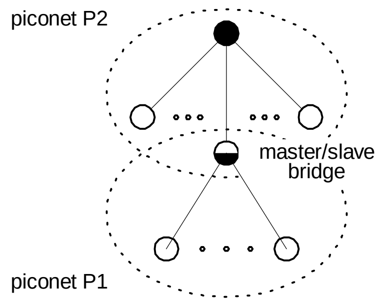
network in which *all* piconets are related through parent-child relationships – is to develop a network-wide distributed scheduling algorithm that will allocate channel time to all devices in an efficient and fair manner. Since time division multiplexing among each parent-child piconet pair is used, we need not worry about collisions between transmissions originating from different piconets in the network which plague

MAC protocols based on CSMA-CA such as 802.15.4:[3] transmissions during allocated CTAs are guaranteed to be conflict-free. However, transmission scheduling and bandwidth allocation pose significant difficulties, in particular when multiple piconets are connected. It is worth noting that the problem of optimal scheduling in a multi-piconet Bluetooth environment (which also uses TDMA) has been shown to be NP-complete.[4]

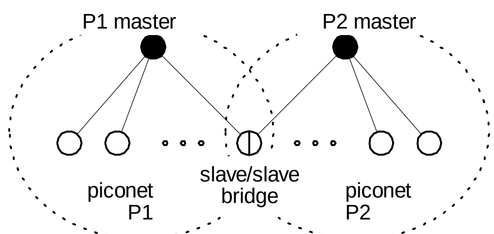
The need to wait until the appropriate active portion of the superframe incurs some additional delays besides the usual transmission delay and access delay in the outbound queue of the source device; furthermore, the bridge device operates its own queues (one for each direction of the traffic) and these can also add delay to the total packet transmission time. Further problems arise from the finite size of various device buffers which packets have to pass on their route from the source node in one piconet, to the destination node in another one. Once these buffers are full, newly arrived packets will be rejected, which leads to packet losses or, if acknowledgments are used, more frequent retransmissions. In the latter case, efficiency is reduced, and the probability of overflowing other buffers earlier in the packet route increases, thus causing rapid performance degradation. If reliable transfer is needed, the possibility of packet blocking necessitates the use of Imm-ACK or Dly-ACK acknowledgment policy. In addition, we must devise an efficient and fair algorithm to partition the available channel time between the piconets, taking into account the traffic intensity both within the piconets and between them.

Using different RF channels. Multi-piconet networks can also be created using a different scenario, in which several multi-piconet networks operate in the same physical space but on different RF channels. While *physical* conflicts between transmissions originating from different multi-piconet networks are still absent by virtue of frequency division multiplexing, *scheduling* conflicts between the piconets will be the main source of complexity, as the device that wants to act as a bridge must alternatively synchronize with piconets that operate according to entirely unrelated schedules. This precludes the use of Master-Slave bridges to interconnect such piconets. Namely, the Master-Slave bridges must not abstain from their duties as the PNCs in their respective piconets for prolonged periods of time. As a result, piconets operating on different RF channels favor interconnection through Slave-Slave bridges, i.e., devices that act as ordinary nodes in each of the piconets they belong to. As such devices have no coordinator duties, their absence from a given piconet will not cause any problems there. In fact, their absence might even go unnoticed if there happens to be no traffic directed to such devices during that time interval.

Neighbor piconets. The 802.15.3 standard also provides the concept of the neighbor piconet, which is intended to enable an 802.15.3 piconet to coexist with another network that may or may not use the 802.15.3 communication protocols; for example, an 802.11 WLAN in which one of the devices is 802.15.3-capable. A PNC-capable device that wants to form a neighbor piconet will first associate with the parent piconet, but not as an ordinary piconet member; the parent piconet coordinator may reject the association request if it does not support neighbor piconets. If the request is granted, the device then requests a private CTA from the coordinator of the parent piconet. once a private CTA is allocated, the neighbor piconet can begin to operate. The neighbor piconet coordinator may exchange commands with the parent piconet coordinator, but no data exchange is allowed. In other words, the neighbor piconet is simply a means to share the channel time between the two networks. Since, unlike the child piconet, data communications between the two piconets are not possible, this mechanism is unsuitable for the creation of multi-piconet networks.



(a) Master-Slave (MS) bridge interconnection.



(b) Slave-Slave (SS) bridge interconnection.

Fig. 1.3. Bridge topologies for multi-piconet networks.

1.4. Implementing Multi-Piconet Networks with 802.15.3

In this Section we will first explain the interconnection (bridging) mechanism, followed by our proposed scheduling algorithm for channel time allocation in a multi-piconet network. The superframe structure of our MAC protocol follows the IEEE 802.15.3 MAC superframe and the channel time allocation is based on TDMA, during the guaranteed access period, and CSMA/CA, during the contention period.

Two common approaches, namely the Master-Slave bridge and the Slave-Slave bridge are used for piconet interconnection in different networks. In the case of a

Master-Slave bridge, Figure 1.3(a), the bridge device is the PNC for Piconet 2 and a normal member of Piconet 1. In the case of a Slave-Slave bridge, Figure 1.3(b), the bridge device is an ordinary member (DEV) in both piconets. We can combine both types of bridges in a multi-piconet environment in order to cover larger areas. The choice of the type of interconnection depends on location of the bridge device within the network. The interconnection will be established through a Master-Slave bridge if a PNC-capable device is located in such a way that it can easily control one piconet and participate in the other one. On the other hand, the Slave-Slave bridge can be used if no suitable PNC-capable device can be found, or if the two piconets operate on different RF channels, possibly because the traffic volume is too high to be serviced with half the available bandwidth.

Operation of the Master-Slave bridge. The bridge establishes a connection between a parent and a child piconet where the bridge device acts as the PNC of the child piconet. The bridge device maintains two queues to temporarily store, and subsequently deliver, the traffic in both directions. As can be seen from Figure 1.2(b), the superframe duration is the same for both parent and child piconets; in fact, the child superframe is simply a private CTA from the parent superframe. The only setup operation needed in this case is for the child piconet PNC to request a private CTA as explained above. Once such a CTA is allocated by the parent piconet PNC, the child piconet PNC simply begins to send beacons at the beginning of the CTA, which is also the beginning of its own superframe. Devices that need to send data to the other piconet can simply request their own CTAs from their respective PNCs.

Operation of the Slave-Slave bridge. A device that is already associated with a piconet can detect the presence of a new piconet by receiving a beacon sent by its PNC, or a control packet with a piconet identification number (PNCID) that is different from the existing one. Whenever a prospective bridge device detects the presence of two piconets within its transmission range, it initiates the connection establishment algorithm (Algorithm 1.1). First, the device waits for the MCTA period or CAP period to send a request command for bridging. Then it will use the four-way handshake (RTS-CTS-DATA-ACK) to send the request command, piggybacking its current scheduling information to the neighbour PNC. The neighbour PNC adjusts its scheduling information based on the received scheduling information from its neighbour piconet. If the PNC is a Master-Slave bridge in its own right, it will request a private CTA from its parent PNC, trying to accommodate the demands of the bridge device. The bridge requirements are, simply, that the neighbouring child piconets obtain channel time for transmission (i.e., private CTAs) without interfering with each other. A positive response from the parent PNC establishes the connection between the child piconets. After the connection establishment, the bridge device needs to maintain a table that keeps track of the scheduled times of activity in each piconet. The PNCID uniquely identifies each record in the table and helps the bridge device switch in a timely fashion between different piconets.

Algorithm 1.1 The Slave-Slave bridge connection.

---

scan presence of overlapped coverage
if scan == positive **then**
  send join-request to neighbour PNC using four way handshaking protocol
  receive feedback from neighbour PNC
  update scheduling table with PNCID and received scheduling information
end if

Channel Scheduling. The channel time scheduling of the network under consideration will be based on average queue size of the devices. The queue size of an ordinary device (i.e., not a PNC or a bridge device) primarily depends on packet arrival rate of that device. The queue size for a bridge device is based on the incoming packet queue sizes from neighbour piconets and outgoing packet queue sizes to the neighbour piconets. The bridge device will use the average of these two queues size to determine its channel time requirement. The devices send requests for channel time based on the average queue size to their respective PNCs. The PNC uses Algorithm 1.2 based on the request from the bridge in question. In case of a request from a bridge device, the PNC schedules channel time and a private CTA (for the child piconet) in such a way that there will be no overlap of channel time between the two adjacent piconets.

A representative topology that employs both types of bridge interconnection is shown in Figure 1.4. In this network, the parent piconets P1 and P2 are located beyond each other's transmission ranges and, thus, can operate on the same RF channel. However, the presence of two child piconets that can hear each other – they are, in fact, interconnected – presents a challenge for scheduling. In order to resolve this, the two parent piconets P1 and P2 will assign channel time for their children in different time slots, based on the scheduling information they exchanged during connection establishment. Let us consider time slots in the superframe in Figure 1.4. The time slots represented by P1/P2 (or P2/P1) imply that P1 and P2 can communicate at the same time. On the other hand, when a child piconet is operating, no other piconet in its range can talk. In this case we can assume that a single superframe (actually two superframes from two different parent piconets) are divided into four time slots. Within each time slot, the devices will have guaranteed channel time and contention period. There are also MCTAs in each time slot during which a new node can join or a bridge can establish a connection. There is a chance of conflict during the MCTA period as the new devices do not have any knowledge of the current scheduling information resulting in the hidden terminal problem. We will use the four way handshaking protocol to resolve this problem.

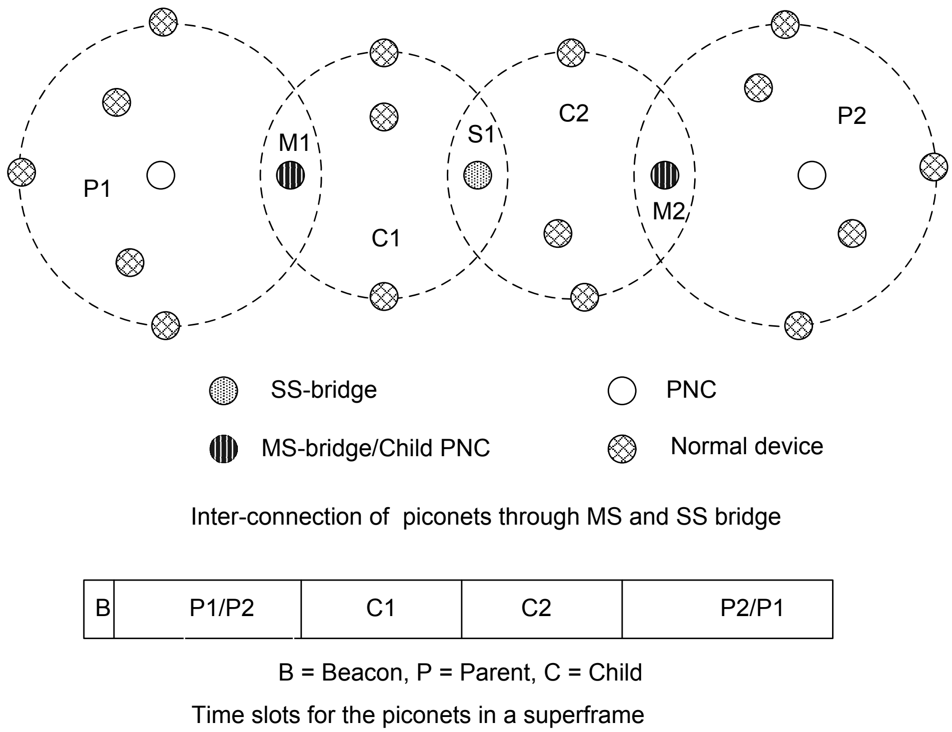


Fig. 1.4. A multi-piconet structure that employs both types of bridge interconnection.

Algorithm 1.2 Scheduling of channel time.

```
if request command from Master-Slave bridge then
    assign private CTA anywhere in the superframe
end if
if request command from Slave-Slave bridge then
    check piggyback data for neighbour scheduling information
    if no scheduling information then
        request for scheduling information
    end if
end if
scheduling information received
calculate required channel time based on average queue size
determine private CTA position
assign private CTA and channel time
```

1.5. Related Work

The MAC protocols for wireless multi-piconet networks are different from the traditional wireless MACs in terms of self organization, distributed nature, multi-hop, and mobility. Multi-piconet networks can be designed with a single RF channel or multiple RF channels. For simplicity, we will focus on a single channel network with a parent-child interconnection.

Multi-piconet networks have often been developed using the IEEE 802.11 DCF MAC protocol, and most of the research work in this area was based on using the features of 802.11 MAC protocol, possibly slightly modified to improve network performance. Bicket *et al.*[5] have evaluated the performance of 802.11b networks; their experiments have shown that an ad hoc network implemented using 802.11b technology can achieve sustained throughput of around 630 Kbps, significantly below the supported data rate of 11 Mbps. Similarly, Yamada *et al.*[6] have identified two problems of 802.11b based networks: limited throughput and degradation of fairness. To solve these problems they have introduced two new control packets, namely Invite-to-Send (ITS) and Copied CTS (CCTS). The use of ITS and CCTS packets has brought some improvements in throughput, but at the cost of increased control overhead and delay. Also, the overhead due to ITS and CCTS packets and end-to-end packet delay will increase with the network load. However, in an 802.15.3 network, data communications are accomplished using dedicated time periods, hence there is no need to introduce additional control packets such as ITS and CCTS.

MACA was developed to solve the hidden and exposed terminal problems of traditional CSMA[7] protocols. In MACA, the sender and receiver exchange RTS and CTS control packets before sending a data packet to avoid collisions. Fullmer and Garcia-Luna-Aceves[8] describe the scenario where MACA fails to avoid collisions due to hidden terminals. MACA may also make a device wait for a long period to access the medium because its use of the BEB * algorithm.[7]

To overcome the problems of MACA, a new solution was proposed by Bharghavan et al. called Media Access Protocol for Wireless LANs (MACAW).[9] Basically MACAW is a modification of the BEB algorithm in MACA. It introduces acknowledgement and data-sending (DS) control packets producing the RTS-CTS-DS-DATA-ACK sequence for data transfer. The IEEE 802.11 standard[10] has been developed by adopting the CSMA and MACAW with further modifications to support wireless LANs.

However, neither IEEE 802.11 MAC nor MACAW provide support for real time data transfers because of the absence of guaranteed time slots. Therefore, Lin and Gerla[11] proposed an enhanced version of MACA called MACA with Piggybacked Reservation (MACA/PR) to support real-time traffic. The MACA/PR protocol is a contention based protocol with a reservation mechanism. It has been designed to support multimedia data in multihop mobile wireless network providing guaranteed bandwidth through reservation. Every node keeps the reservation information of sending and receiving windows of all the neighbour nodes in a table, which is

*In Binary Exponential Backoff (BEB), a device doubles the size of its backoff window if a collision is detected.

refreshed after every successful RTS-CTS-DATA-ACK (known as four way handshaking protocol) cycle. The RTS and CTS packets are exchanged for the first packet in the transfer of a series of real-time data packets. The reservation information for the next data packet is piggybacked with the prior data packet and the receiver confirms this reservation in the acknowledgement control packet.

The limitation of MACA/PR is that it requires help from the network layer routing protocol. However, MACA/PR has better performance in terms of latency, packet loss, mobility, and bandwidth share than both asynchronous packet radio network (PRNET †) and synchronous TDMA based MACs. The use of fixed reserved time slots in MACA/PR can result in wastage of bandwidth. Manoj and Ram Murthy[12] have proposed a modification to the reservation mechanism of MACA/PR to prevent bandwidth wastage. In the modified scheme, the reserved slots can be placed at any position in the superframe and unused resources (channel time) are released after a successful transmission.

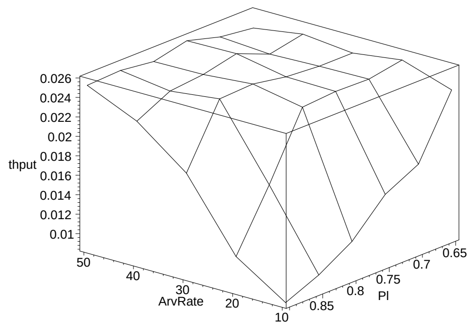
We note that the 802.15.3 MAC uses TDMA based channel allocation to provide guaranteed time slots for data transfer. However, the piggybacked reservation information of MACA/PR can be employed together with the TDMA based MAC to support real-time data transfer along with best-effort traffic in 802.15.3 based multi-piconet networks.

Xiao[13] has performed a detailed performance evaluation of the IEEE 802.15.3[1] and IEEE 802.15.3a[14] standards through simulation and mathematical analysis. He has also done a throughput analysis of the 802.11[10] protocol, which uses backoff with counter freezing during inactive portions of the superframe. The freezing and backoff techniques are essentially the same in the 802.11 and 802.15.3 MACs, except that different ways of calculating the backoff time are utilized. The backoff and freezing have an impact on the performance of the network; especially the backoff has a direct impact on the delay parameter. Large backoff windows can result in longer delays. On the other hand, small backoff windows may increase the probability of collisions. Xiao used the backoff procedures defined in the 802.11 and 802.15.3 MAC specifications; this work gives us performance of the protocol in terms of throughput over various payload sizes, but the performance of reliable transmission in error-prone wireless network during contention period needs more study.

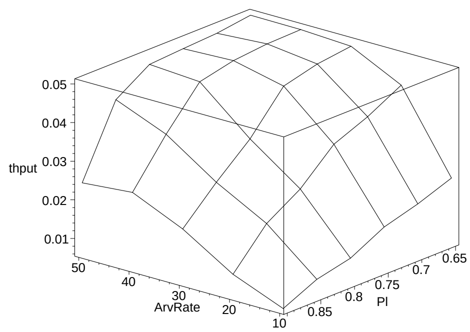
1.6. Fixed vs. adaptive CTA allocation

To investigate the performance of an 802.15.3 network formed by two piconets interconnected with a MS bridge, we have built a simulator using the object oriented Petri Net engine Artifex by RSoftDesign, Inc.[15] In the topology considered, the parent piconet consisted of nine devices, while the child piconet consisted of five devices (not counting either coordinator). The child piconet coordinator is also the

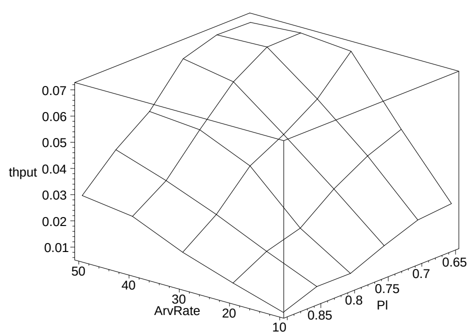
† In a PRNET, the devices use the same channel and share it dynamically.



(a) Throughput at $b = 3$.



(b) Throughput at $b = 5$.



(c) Throughput at $b = 7$.

Fig. 1.5. Bridge throughput under fixed bandwidth allocation.

bridge toward the parent piconet. Each of the ordinary nodes in either piconet has Poisson traffic with a specified packet arrival rate, with packets of 1200 bytes each (chosen to correspond to an average IP packet size). Packet destinations are randomly selected within the same piconet with the total probability of P_l, and in the other piconet with the probability of $1 - P_l$. For convenience, we refer to the traffic from the parent to child piconet as downlink traffic, and the traffic in the opposite direction as uplink traffic. The structure of the superframe is shown in Figure 1.2(c). Each of the ordinary devices is allocated a CTA sufficient for four packets, while the bridge device is allocated an additional CTA with b packet slots for inter-piconet traffic; this holds for both parent and child piconet superframes. In this manner, the total superframe duration is close to the maximum value of $65.535ms$ prescribed by the standard.[1]

The total run length of our simulation is 100 seconds, after the initial warm up period of 6 seconds. We have varied the packet arrival rate from 10 to 50 packets/s, while the locality probability was varied from 0.65 to 0.9. The size of the downlink queue at the bridge was fixed at 100 packets. Measured bridge throughput is shown in Figure 1.5 as a function of packet arrival rate (in packets per second per node) and locality probability P_l. The number of packet slots in the CTA allocated to bridge downlink traffic (which, in Figure 1.2(c), corresponds to the MSB period in child piconet superframe) has been varied from $b = 3$ to 7. As can be seen, higher bandwidth allocation results in higher throughput and reduced blocking probability at the bridge; but at the same time, it takes some time away either at the expense of CTAs allocated to other devices. Higher bandwidth may also be wasted if there is no traffic.

A much better solution would be to allocate the bandwidth to the bridge in an adaptive manner. This, however, is complicated by the essentially random character of inter-piconet traffic due to Poisson packet arrivals. To smooth those fluctuations, we apply a simple transformation known as exponentially weighted moving average, or EWMA.[16] In this approach, the bandwidth allocated by the bridge is determined on the basis of exponentially smoothed value of bridge downlink queue size, which is recalculated at each new packet arrival as

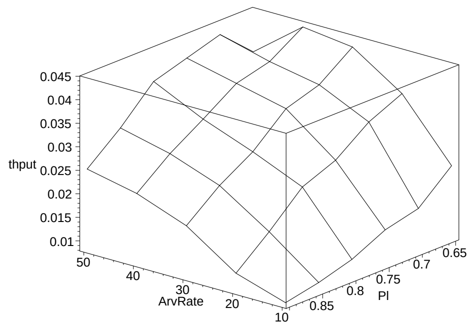
$$\overline{Q}_{i+1} = \alpha Q_{i+1} + (1-\alpha)\overline{Q}_i \tag{1.1}$$

where Q and $\overline{Q}$ denote the instantaneous and smoothed bridge downlink queue size, while the index refers to the time instant of packet arrival. The level of smoothing depends primarily on the smoothing constant α: the higher its value, the more weight is assigned to most recent reading. The actual bandwidth allocation follows Algorithm 1.3. Changes may be explicitly requested by the bridge and granted by the coordinator of the child piconet; alternatively, the bridge can just report the instantaneous queue size in each superframe, and the coordinator can use the algorithm to calculate the required bandwidth allocation itself.

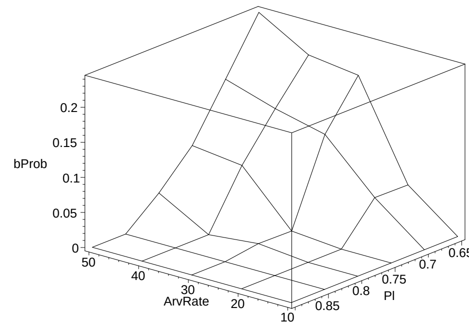
Algorithm 1.3 Adaptive CTA allocation for down link

1: Q_{i+1} measured and/or reported by the bridge
2: $\overline{Q}_{i+1} = \alpha Q_{i+1} + (1-\alpha)\overline{Q}_i$
3: CTAforDLqueue $= (\overline{Q}_{i+1} - q_0) \bmod \Delta q + b_0$

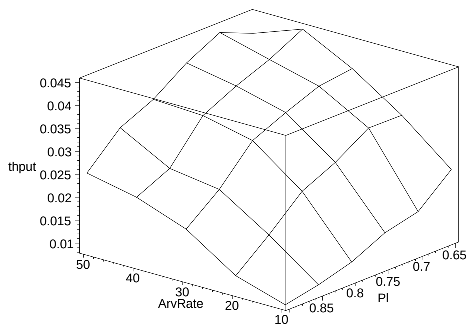
Measured results for bridge throughput are shown in the diagrams in the left



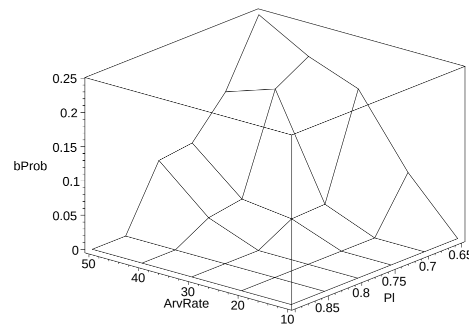
(a) Throughput, $\alpha = 0.3$.



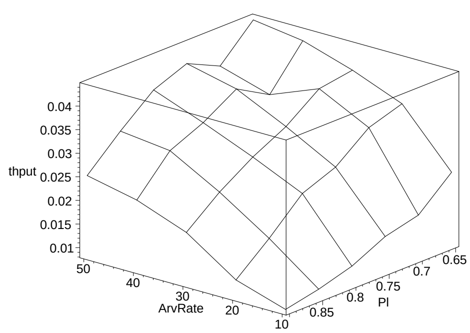
(b) Blocking probability, $\alpha = 0.3$.



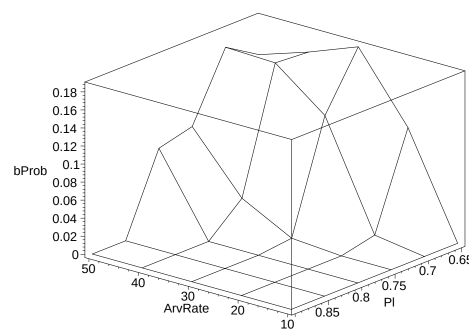
(c) Throughput, $\alpha = 0.5$.



(d) Blocking probability, $\alpha = 0.5$.



(e) Throughput, $\alpha = 0.7$.



(f) Blocking probability, $\alpha = 0.7$.

Fig. 1.6. Bridge throughput and blocking probability under adaptive bandwidth allocation.

hand column of Figure 1.6, for different values of the smoothing constant α; we have set $b_0 = 3$, $q_0 = \Delta q = 20$. As can be seen, the adaptive algorithm manages to maintain the throughput at values close to the one obtained under fixed bandwidth allocation with $b = 5$. On the basis of these measurements, it seems that the value $\alpha = 0.5$ would be the best choice for the smoothing constant, although the differences are rather small.

Unfortunately, exponential smoothing does not solve all problems, as can be seen from the diagrams of blocking probability, shown in the right hand column of Figure 1.6, which demonstrate that this parameter is not significantly affected by the choice of the smoothing constant α. Namely, as soon as a packet leaves the downlink queue, the allocation will be adjusted downward, and the queue will take longer to empty. As a result, the amount of bandwidth allocated to the bridge is still too sensitive to the average downlink queue size, and benefits of adaptivity are not fully realized.

1.7. Adaptive CTA with threshold hysteresis

In order to correct this, we have introduced a small amount of hysteresis into the adaptive bandwidth allocation algorithm. Namely, the thresholds for allocation adjustment will differ, depending on the direction of the adjustment, and upward adjustments will have higher thresholds than their downward counterparts. In this manner, when a sufficient number of packets arrive to the downlink queue, the bandwidth allocation will remain high until the bulk of the increase is processed, i.e., delivered to their respective destinations. The modified algorithm is shown as Algorithm 1.4 below.

Algorithm 1.4 Adaptive CTA allocation with hysteresis threshold for downlink.

Q_{i+1} measured and/or reported by the bridge
$\overline{Q}_{i+1} = \alpha Q_{i+1} + (1-\alpha)\overline{Q}_i$
if $\overline{Q}_{i+1} > \overline{Q}_i$ **then**
  CTAforDLqueue $= (\overline{Q}_{i+1} - q_0) \bmod \Delta q + b_0$
else
  CTAforDLqueue $= (\overline{Q}_{i+1} - (q_0 - \chi)) \bmod \Delta q + b_0$
end if

The dependency between the bandwidth allocation and average queue size in the adaptive algorithm with hysteresis can be graphically depicted as in Figure 1.7.

The measured results, including throughput, blocking probability at the bridge, and average downlink queue size, are shown in Figure 1.8. For illustration, we have shown these results for three values of the smoothing constant α.

Finally, Figure 1.9 shows the corresponding throughput and blocking probability. As can be seen, throughput shows a marked improvement in the case $\alpha = 0.5$, being close to the value achieved under fixed bandwidth allocation with $b = 7$, Figure 1.5(c). Furthermore, blocking probability is much lower than for other values of α, and it remains lower in a wider range of values of packet arrival rate and locality probability. Similar observation can be made for the average downlink queue size at the bridge, which in the most part of the observed range stays well below the maximum size of 100 packets. This confirms the validity of the adaptive bandwidth

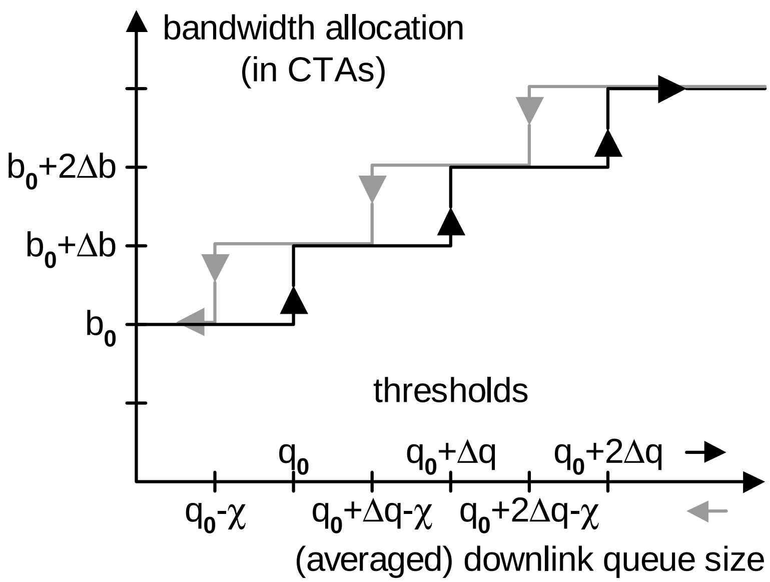


Fig. 1.7. Adaptive CTA allocation with hysteresis.

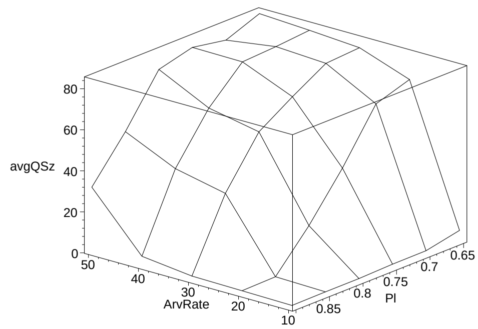
allocation approach.

1.8. Conclusion

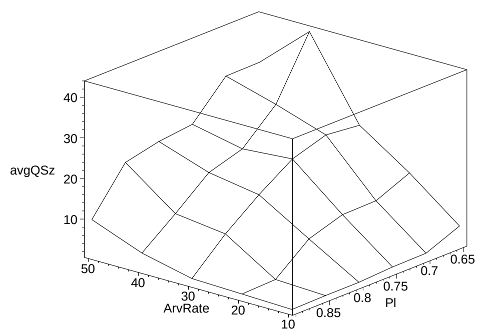
In this chapter, we have considered the performance of a two-piconet network built upon the recent IEEE 802.15.3 high data rate WPAN standard. The parent and child piconets are interconnected through a Master-Slave bridge which also acts as the coordinator of the child piconet. We have shown that an adaptive algorithm for allocating downlink bandwidth to the bridge, utilizing threshold hysteresis, can easily outperform any fixed bandwidth allocation algorithm. Further work will focus on extensions to control the bandwidth allocation for uplink traffic, as well as that of ordinary nodes in the network.

References

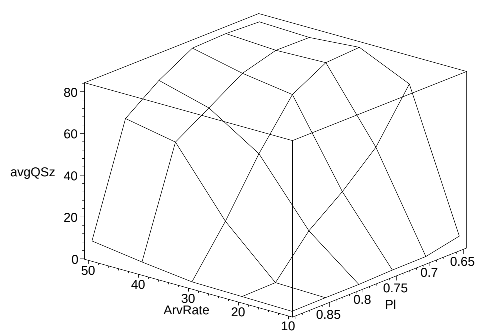
1. Standard for part 15.3: Wireless medium access control (MAC) and physical layer (PHY) specifications for high rate wireless personal area networks (WPAN). IEEE standard 802.15.3, IEEE, New York, NY, (2003).
2. I. F. Akyildiz, X. Wang, and W. Wang, Wireless mesh networks: a survey, *Computer Networks*. **47**, 445–487 (March, 2005).
3. J. Mišić and V. B. Mišić, *Wireless Personal Area Networks: Performance, Interconnections and Security with IEEE 802.15.4*. (John Wiley and Sons, New York, NY, Jan. 2008).



(a) Average queue size at $\alpha = 0.3$.



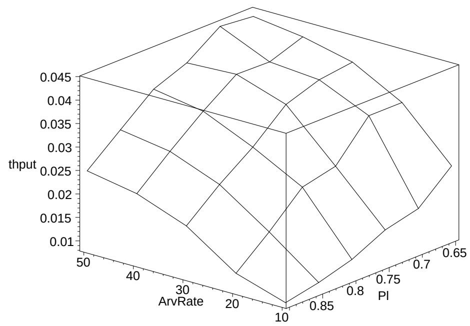
(b) Average queue size at $\alpha = 0.5$.



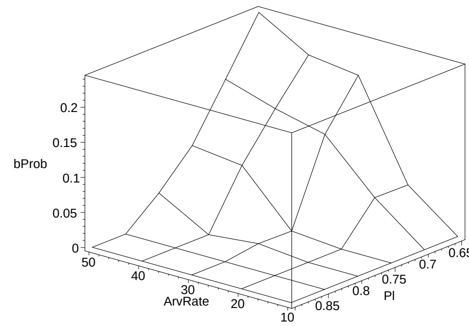
(c) Average queue size at $\alpha = 0.7$.

Fig. 1.8. Average queue size under adaptive algorithm with hysteresis.

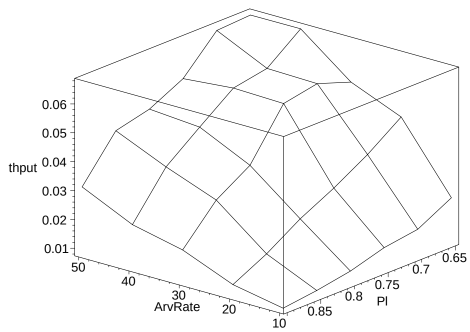
4. N. Johansson, U. Körner, and L. Tassiulas. A distributed scheduling algorithm for a Bluetooth scatternet. In *Proceedings of the International Teletraffic Congress – ITC-17*, pp. 61–72, Salvador de Bahia, Brazil (Sept., 2001).
5. J. Bicket, D. Aguayo, S. Biswas, and R. Morris. Architecture and evaluation of an unplanned 802.11b mesh network. In *11th ACM International Conference on Mobile Computing and Networking*, vol. 1, pp. 31–42, Cologne, Germany (September, 2005).
6. A. Yamada, A. Fujiwara, and Y. Matsumoto. Enhancement of mesh network oriented



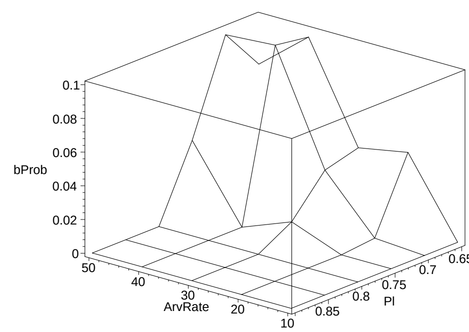
(a) Throughput at $\alpha = 0.7$.



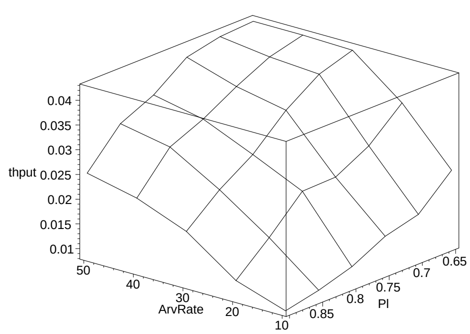
(b) Blocking probability at $\alpha = 0.7$.



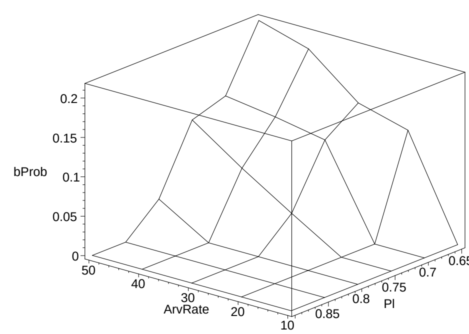
(c) Throughput at $\alpha = 0.5$.



(d) Blocking probability at $\alpha = 0.5$.



(e) Throughput at $\alpha = 0.3$.



(f) Blocking probability at $\alpha = 0.3$.

Fig. 1.9. Performance of adaptive algorithm with hysteresis.

IEEE 802.11 MAC protocol. In *10th Asia-Pacific Conference on Communications and 5th International Symposium on Multi-Dimensional Mobile Communications*, vol. 1, pp. 142–146, Kanagawa, Japan (September, 2004).

7. C. Ram Murthy and B. Manoj, *Ad Hoc Wireless Networks, Architecture and Protocols*. (Prentice Hall, Upper Saddle River, NJ, 2004).
8. C. L. Fullmer and J. J. Garcia-Luna-Aceves. Floor acquisition multiple access (FAMA)

for packet-radioi networks. In *ACM SIGCOMM '95*, pp. 262–273, Cambridge, MA (September, 1995).
9. V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. MACAW: a media access protocol for wireless LAN's. In *ACM SIGCOMM '94*, pp. 212–225, London, UK (August, 1994).
10. Standard for wireless LAN medium access control (MAC) and physical layer (PHY) specifications. IEEE standard 802.11, IEEE, New York, NY (May, 1997).
11. C. R. Lin and M. Gerla. MACA/PR: an asynchronous multimedia multihop wireless network. In *IEEE INFOCOM '97*, pp. 118–125, Kobe, Japan (March, 1997).
12. B. S. Manoj and C. S. Ram Murthy. Real-time support for ad hoc wireless networks. In *IEEE ICON '02*, pp. 335–340, Grand Copthorne Waterfront, Singapore (August, 2002).
13. Y. Xiao, MAC layer issues and throughput analysis for the IEEE 802.15.3a UWB, *Dynamics of Continuous, Discrete and Impulsive Systems, Series B: Applications & Algorithms*. **12**, 443–462 (June, 2005).
14. *IEEE 802.15 WPAN high rate alternative PHY Task Group 3a (TG3a)*. (Available online at `http://www.ieee802.org/15/pub/TG3a.html`, August 2006).
15. *Artifex v.4.4.2*. (RSoft Design, Inc., San Jose, CA, 2003).
16. S. Delurgio, *Forecasting – Principles, and Application*. (McGraw-Hill/Irwin, New York, 1998).