Project: IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs)

Submission Title: [Mediation Device Operation]

Date Submitted: [14 May, 2001]

Source: [Ed Callaway] **Company: [**Motorola]

Address: [8000 W. Sunrise Blvd., M/S 2141, Plantation, FL 33322]

Voice: [(954) 723-8341]**, FAX:** [(954) 723-3712]**,**

E-mail: [ed.callaway@motorola.com]

Re: [WPAN-802.15.4 Proposal support]

Abstract: [This document describes the operation of the Mediation Device, proposed by Motorola for the 802.15.4 standard, including its operation in a dedicated device and as a function distributed among network devices. Operation in small networks is also described.]

Purpose: [Supporting information to Motorola 802.15.4 proposal]

Mediation Device Operation

Rev. 0; 4/2001 (Qicai Shi, Ed Callaway)
Rev. 1; 5/2001 (Monique Bourgeois)

Motorola Labs

1. Introduction

IEEE 802.15.4 is a standard that places more emphasis on long battery life and low cost and less emphasis on message latency.

To lower the power consumption, the duty cycle must be reduced. However, for an asynchronized system, low duty cycle means there is a poor chance for the devices to synchronize with each other. For instance, device A tries to make contact with device B, but device B is likely to be sleeping while device A is talking due to their low duty cycles.

Low cost crystals and on-chip Micro Electro-Mechanical System (MEMS) resonators are considered key reference frequency technologies for reducing cost. The issue with these technologies is poor frequency stability, which compounds the device synchronization problem.

This paper introduces a new protocol that enables low duty cycle devices to communicate with each other without requiring a high accuracy synchronization reference, thus overcoming the issue of poor frequency stability.

2. Low duty cycle frame structure

To lower power consumption, the duty cycle has to be reduced. Figure 1 gives an example of low duty cycle frame structures.

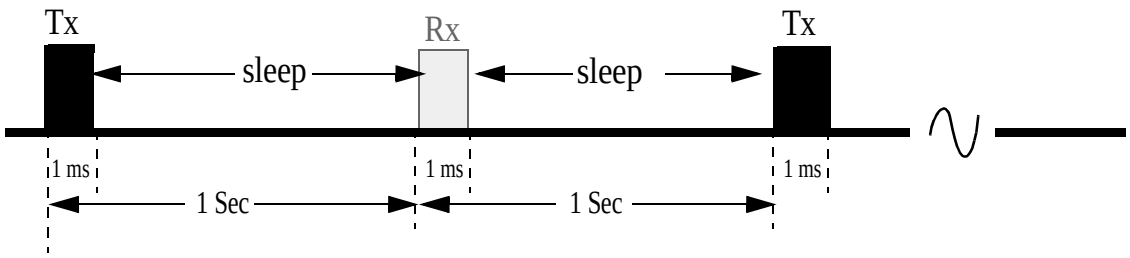

Figure 1: Low Duty Cycle Frame Structure

Both the transmit and receive time slots are 1 ms long; the device goes into a deep sleep after transmitting or receiving. After 1 second, it wakes up and repeats the cycle. The duty cycle is about 0.1% and, therefore, very energy efficient.

3. Issue with low duty cycle schema

Take a look at the example shown in Figure 2 :

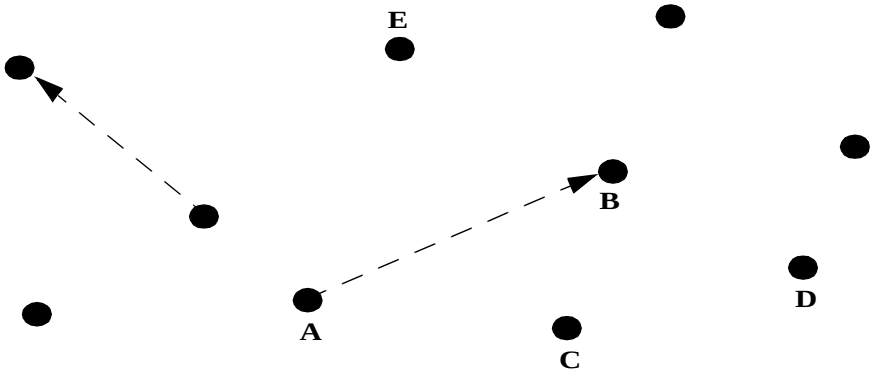

Figure 2: Low Duty Cycle Issue

Device A tries to talk to device B. Since both A and B are awake only 0.1% of the time and A and B don't know each other's time schedule, the probability for A to synchronize with B is very low.

4. Mediation Device (MD)

In order to overcome this issue, a 'Mediation Device' (MD) is introduced here. A MD can record and replay a message; it functions as an "answering machine".

A MD may or may not have a relatively high duty cycle. An possible frame structure is shown in Figure 3 :

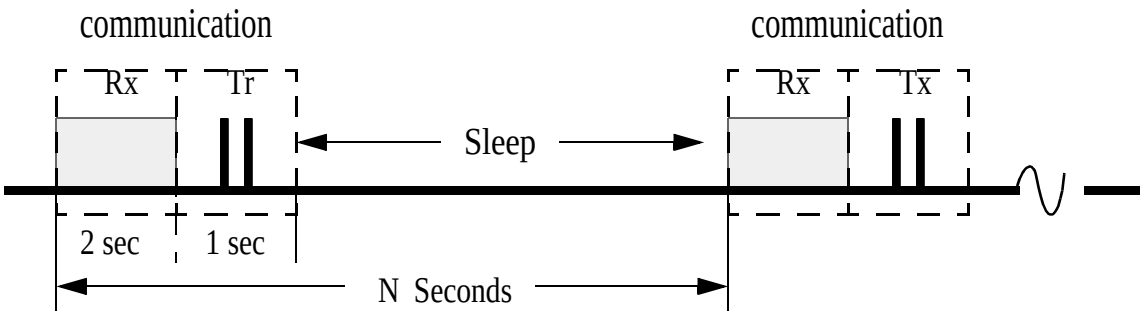

Figure 3: Mediation Device Frame Structure

The MD allows 2 seconds for receiving messages. After the receive period is over, the MD spends up to 1 second sending 1ms Ack messages to the devices from which it heard. If the MD did not hear from any devices, then it does not send any Ack messages. The MD has a receive period which is longer than the time it takes for a device in normal mode to complete two consecutive transmissions. This is necessary to guarantee that the MD can receive messages from all in-range devices.

The most important task of a MD is to record and replay simple control messages such as the following: “who is talking”, “with whom does it want to talk”, and “ what time it will talk again”, etc.

5. Dedicated MD

In this implementation, the MD is a dedicated device. For star networks, in which all devices are within range of a single device, there is one MD. The MD has larger power consumption than the regular devices.

Figure 4 shows us an example of how the system functions.

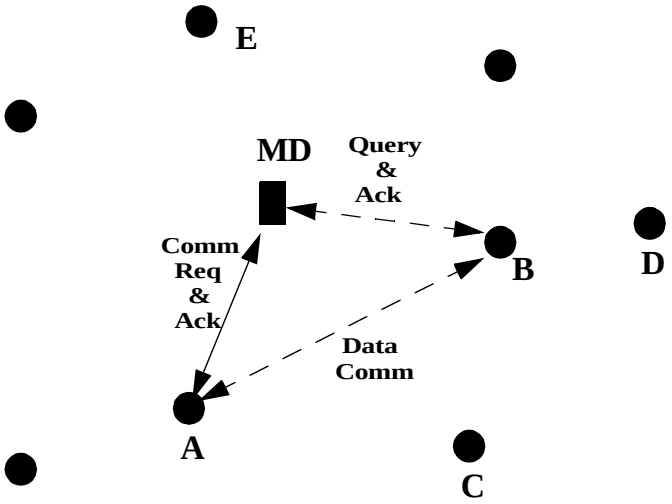

Figure 4: Dedicated MD Operation

In Figure 4,

1). Device A wants to talk to B; it sends out a Communication Request (Comm Req) message and waits for an Ack.
2). Device B is sleeping at this moment, but fortunately the MD has a long communication period. The MD intercepts A’s message which includes A’s id, B’s id and A’s communication timing schedule information.
3). When device B wakes up, B will send out a Query message that is also intercepted by the MD.
4). Once the MD receives B’s Query message, the MD sends an Ack back to B and forwards A’s request to communicate to B.
5). Device B receives the Ack from the MD, finds out that A wants to contact him and also finds out when A will try to contact him again.
6). Since B now knows A’s time schedule, B will synchronize with A. Now they can communicate.

(An alternate way : Once B knows that A is trying to talk with him, B will wake up for a long period and sync with A.)

Since the MD is a dedicated device, it can have a fixed location. Its location can be used as a reference point for a relative location determination algorithm.

This protocol is good for an indoor environment where the MD can operate on AC power.

5.1 Timing schedule for this protocol

Figure 5 shows a detailed timing schedule for this protocol. In this example, the MD receives the ***Query message later than the Comm Req***.

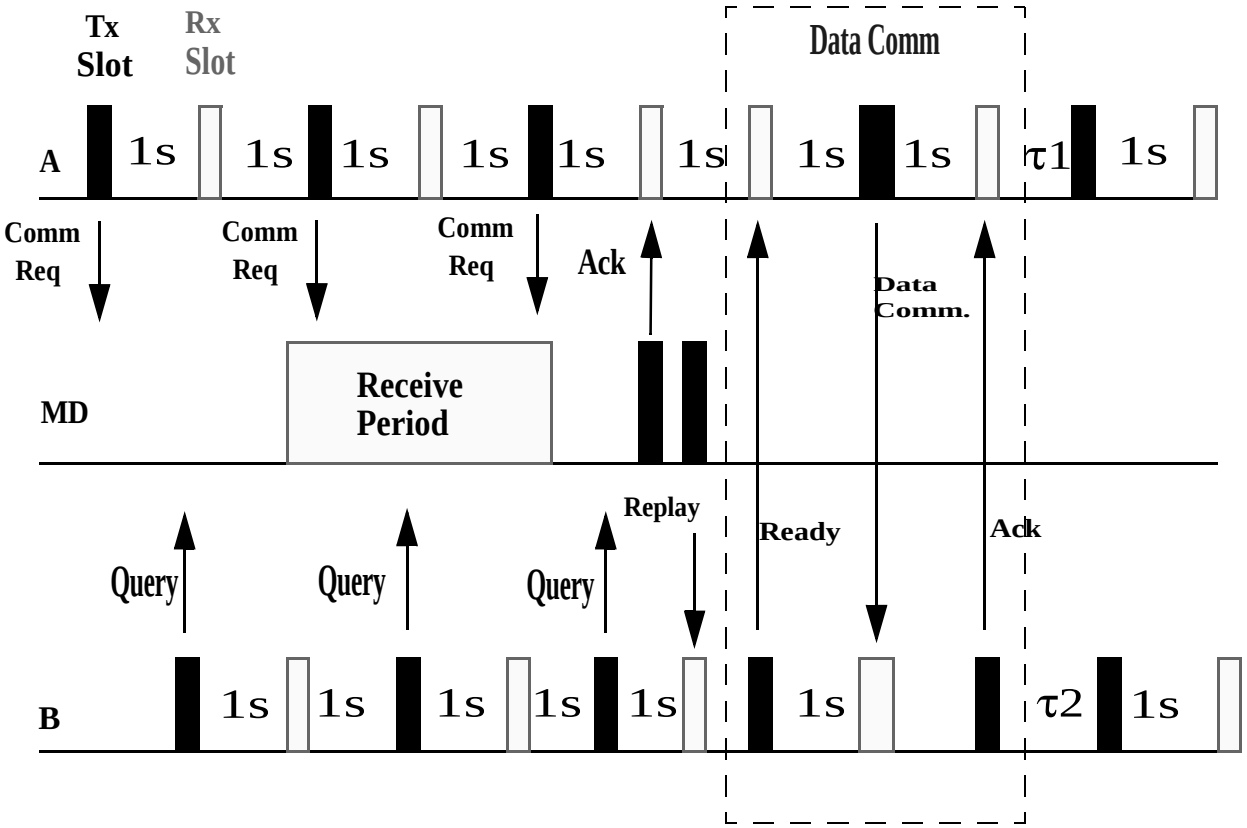

Figure 5: Dedicated MD Timing

Timing schedule for device A in Figure 5:

In this example, device A wants to communicate with device B. A's transmit/ receive window is 1ms, A's sleep period is 1 second, and A has a 0.1% duty cycle.

1). Device A sends out a Comm Req message. However, no device receives this message.
2). Device A sleeps for 1 second then wakes up and listens. Since no device receive A's Comm Req message, A will not receive an Ack message.
3). Device A sleeps for 1 second and sends out the Comm Req message again. This time the MD receives A's message.
4). Device A sleeps for 1 second then wakes up and listens. Because the MD is still in the

receive period, the MD will not send back an Ack to A. So Device A will not receive anything.
5). Device A sleeps for 1 second and sends out the Comm Req message again. The MD receives A's message again.
6). Device A sleeps for 1 second then wakes up and listens. This time A receives an Ack from the MD.
7). Device A knows that his message has been picked by the MD, and A will not send out the Comm Req message again. A sleeps for 1 second, wakes up and listens. This time, A receives B's Ready message. A now knows that B is synchronized with him.
8). A sleeps for 1 second and transmits Data to B.
9). A sleeps for 1 second and receives an Ack message from B. At this point, A has sucessfully finished the task of sending data to B.
10). Instead of sleeping the regular 1 second period, A sleeps a random time t1 (0<t1<2 Sec) to randomize its time schedule with respect to B's. Then A wakes up and sends out a query message.

.....
So on and so forth.

Timing schedule for the MD in Figure 5:

In this example, the MD is the device that mediates the communication between device A and device B. It has a long receive period (2002 ms), so it is guaranteed that the MD can receive messages for all other devices within the MD's communication range.

1). Initially, the MD is in sleep mode. The sleep period can be several seconds.
2). After a while, the MD wakes up and listens. During the receive period, the MD receives a Comm Req message from A and records this message. Then, the MD receives a query message from B and records this message. The MD receives a third message which is the same message as the first message from A; the MD refreshes this message.
3). Since the MD has A's timing information, it syncs with A and sends an Ack message to A.
4). The MD also has B's timing information. It then syncs with B and replays A's contact information to B. The contact information includes A's timing information.
5). The MD goes back to sleep.

.....
So on and so forth.

Timing schedule for device B in Figure 5:

In this example, B is the device with which A tries to talk. B's transmit / receive window is 1 ms, B's sleep period is 1 second, and B has a 0.1% duty cycle.

1). B sends out a Query message to the MD to see if there any messages intended for B, but the MD is sleeping and does not receive this message.
2). B sleeps for 1 second, wakes up and listens. Since no device receives B's Query message, B will not receive an Ack message.
3). B sleeps for 1 second and sends out the Query message again. This time, the MD receives B's message.
4). B sleeps for 1 second, wakes up and listens. However, the MD is still in the receive period, so the MD will not send an Ack to B.

5). B sleeps for 1 second and sends out the Query message again. The MD is no longer in the receive period, so the MD cannot receive B's message. This is not a problem since the MD already has B's Query message.
6). B sleeps for 1 second, then wakes up and listens. B receives the replay message, which includes A's timing information, from the MD.
7). B now knows A's timing information and will adjust its timing to synchronize with A. B then sends a Ready message to A.
8). B sleeps for 1 second and listens. B successfully receives the data from A.
9). B sleeps for 1 second and sends an Ack message to A.
10). Instead of sleeping the regular 1 second period, B sleeps a random time t2 (0<t2< 2 Sec) to randomize its time schedule with respect to A's. Then B wakes up and sends a Query message.

.....

So on and so forth.

What if the Query is earlier than the Comm Req ?

This is not a problem, because the MD stores all the messages and processes them after the ***Receive Period***.
At the end of the receive period, the MD has A's timing information. Then the MD can pass this information to B, and B will send a Ready message to A. Based on A's timing information, B will change its own timing and sync with A.

5.2 Handling multiple requests for access to the same device

What if two (or more) devices want to communicate with the same device?

The MD is capable of handling multiple Comm Reqs for the same device. Figure 6 illustrates this scenario. Devices B and C both want to talk with A. Device A will receive both B's and C's timing information and will adjust its duty cycle in order to synchronize with both B and C. In other words, the MD will tell A that both B and C want to make contact, and it will instruct A to change its Tx / Rx times to synchronize with B and add intermediate Tx / Rx slots to communicate with C. All the devices will wait a random time t (0<t<2 seconds) before resuming normal operations.

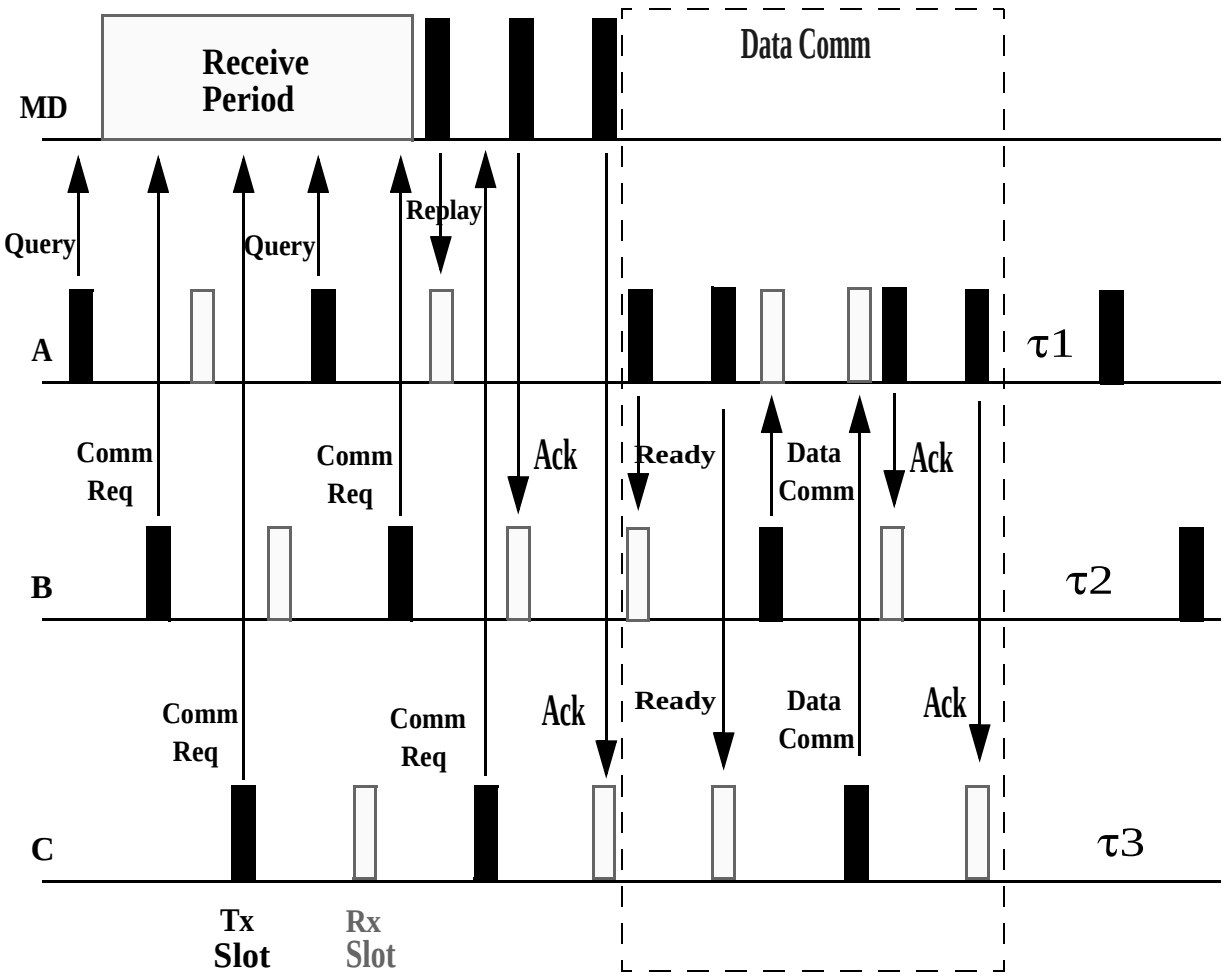

Figure 6: Dedicated MD Scenario When Devices B and C Both Request to Communicate with Device A

5.3 Handling multiple requests from the same device

What if a single device wants to communicate with two (or more) devices?

In this scenario, A wants to talk with both B and C. Obviously, the MD cannot simply give them both A's timing information, because B and C would collide while trying to contact A. Device B will receive A's timing information, but device C will receive a multiple of the timing information. C will receive instruction from the MD to contact A during A's next Rx slot following A's communication exchange with B. After each device has finished talking, it will wait a random time t1, t2 or t3, where (0<t<2 seconds) before resuming normal operation.

See Figure 7 for an illustration of how this works.

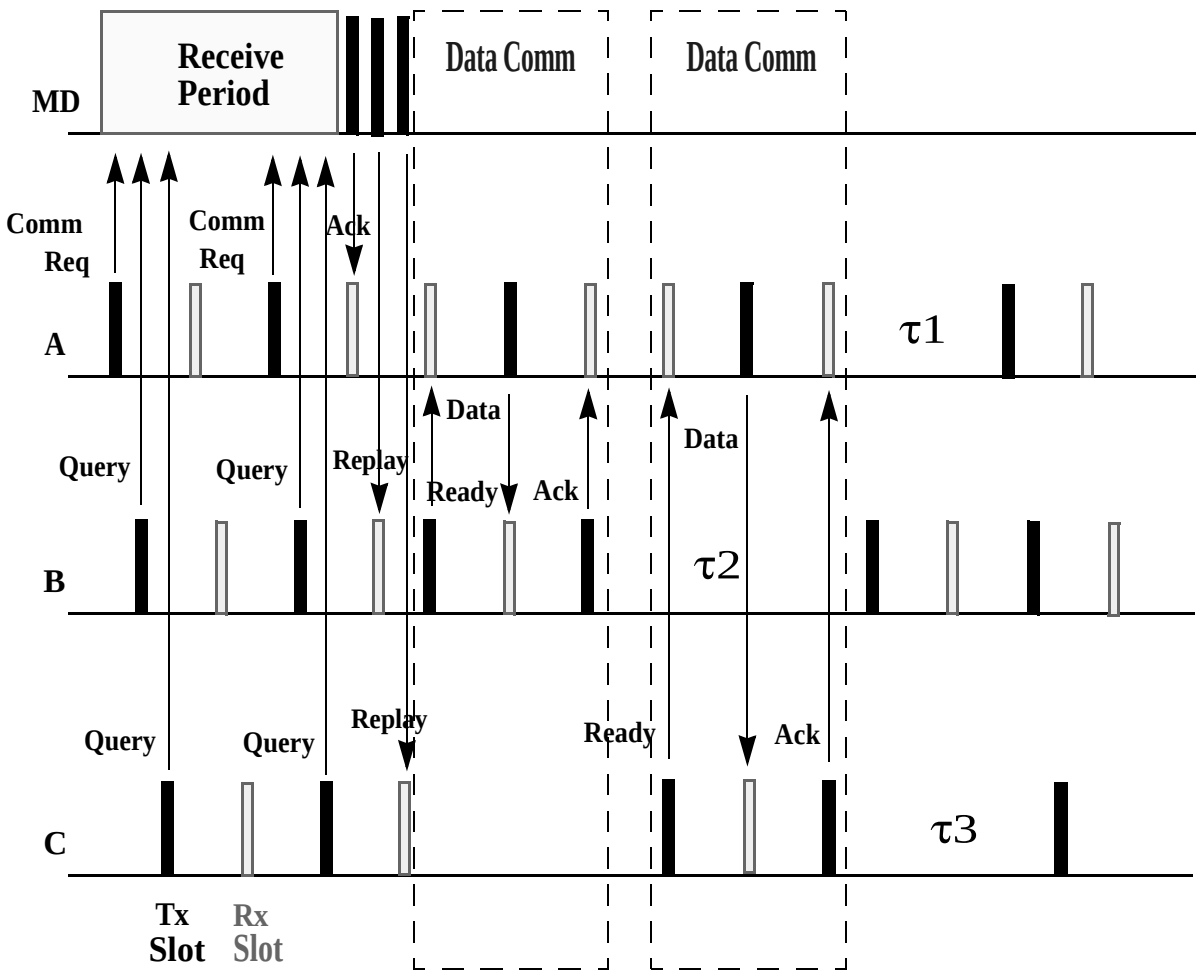

Figure 7: Dedicated MD Scenario When Devices A Requests to Communicate with Devices B and C

6. Distributed MD

In this implementation, the MD is not a dedicated device. It is instead part of the functionality of every device in the network. All devices within the network have the responsibility to function as a MD at certain points in time (just as in a bicycle race, each rider has to be the leader once in a while).

A device becomes a MD randomly. Once it is an MD, it functions exactly as a dedicated MD as described earlier. After one MD period, the device goes back to the normal model. This pattern repeats throughout the lifetime of the device.

A timing schedule example for this protocol is shown in Figure 8:

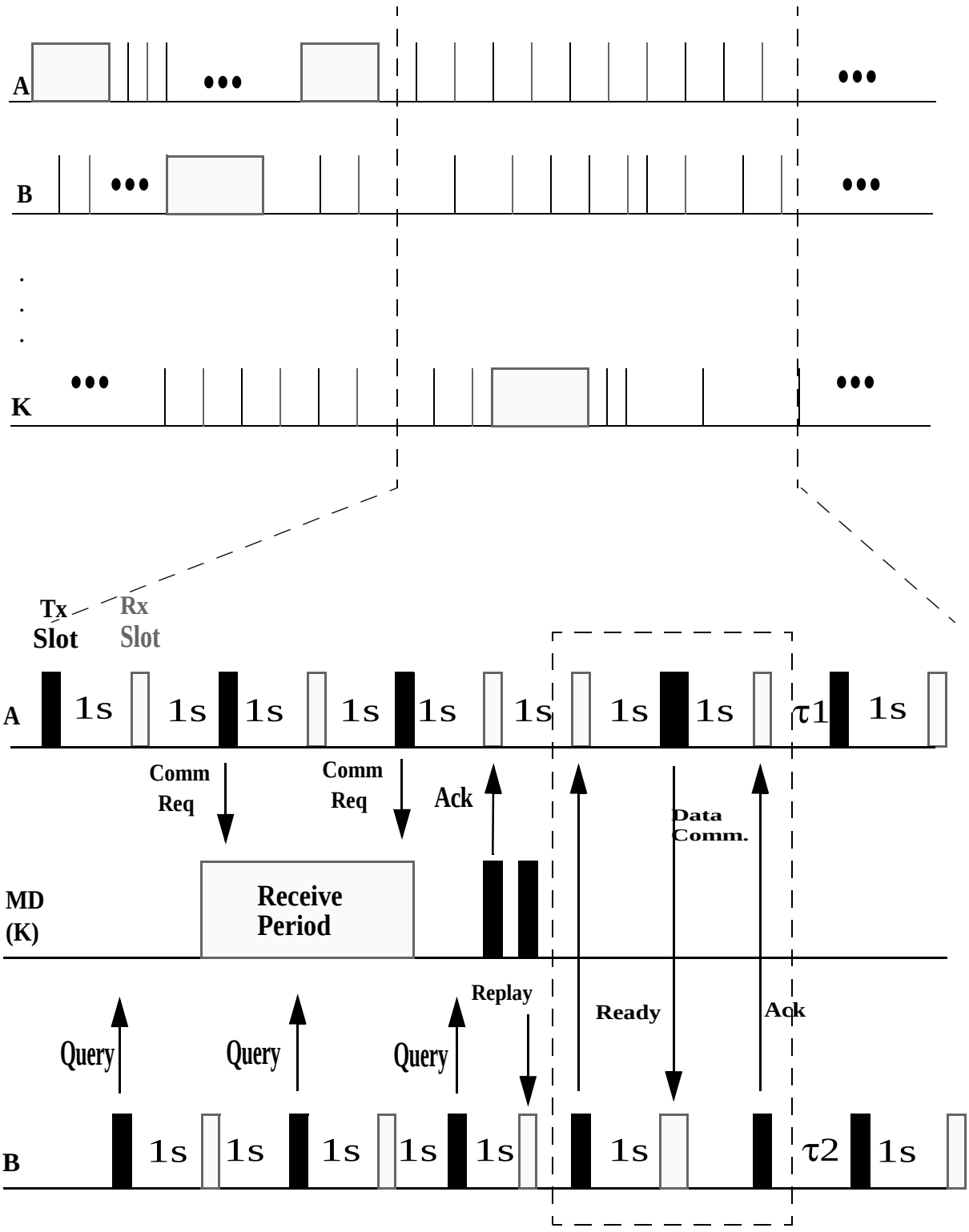

Figure 8: Distributed MD Timing for a Network Cluster

The top part of Figure 8 shows the big picture of the system timing schedule for devices A through K. The bottom part of the figure shows a zoomed-in picture of the selected time window.

From the big picture, we can see that all the devices function as a MD at some point in time, each one according to its own time schedule. After completing its duties as a MD, the device goes back to normal mode.

From the detailed picture, we can see that once the device becomes a MD, it functions exactly the same way as a dedicated MD.

Figure 9 takes a closer look at one specific device.

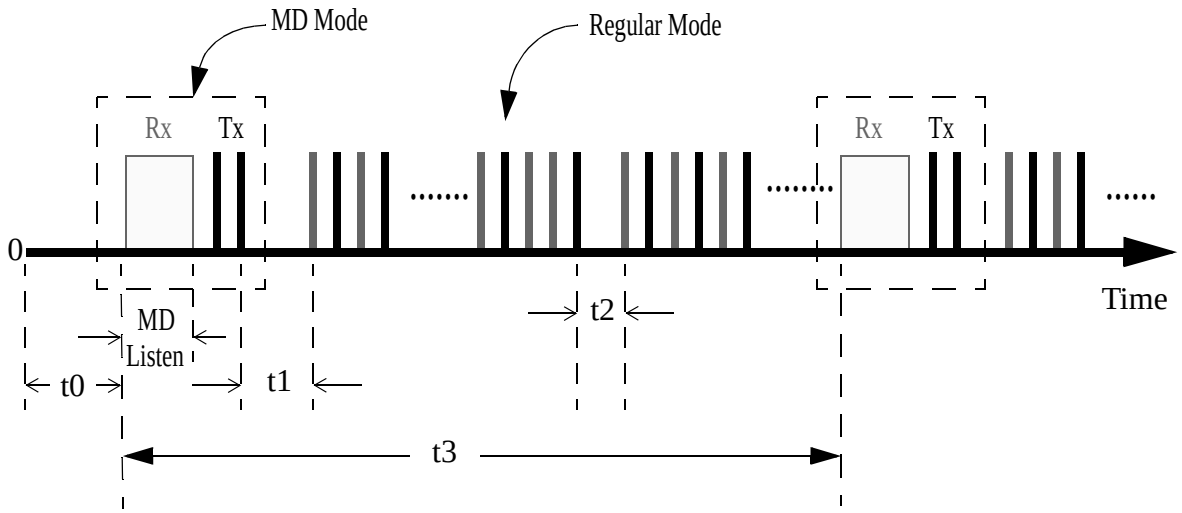

Figure 9: Distributed MD Timing for a Single Device

In Figure 9,

1). In MD mode, the receive period is 2002 ms, and each transmit period is 1 ms. The delay (sleep) periods between transmissions depend on the time schedules of the devices that sent messages to the MD during its receive period.
2). In normal mode, the transmit and receive periods are 1 ms, and the sleep period is 1 second.
3). During the communication period, the sleep time needed to synchronize the devices is the same as discussed in Section 5, Dedicated MD.
4). To randomize the time schedules of the devices, every device waits its own time t0 ($0 < t0 < 2$ seconds) after power up (or reset) before beginning normal operation.
5). Time t1 is the delay between exiting MD mode and re-entering normal mode for a given device, where t1 is a random delay time ($0 < t1 < 2$ seconds).
6). A random delay of t2 ($0 < t2 < 2$ seconds) is introduced after a successful communication sequence between two devices. Each device waits its own time t2 before continuing normal operation, in order to randomize the time schedules of the two devices that were synchronized.
7). Time t3 is the random time (1500 seconds $< t3 <$ 2500 seconds) measured from when a device enters MD mode, goes back into normal mode and then re-enters MD mode. t3 must be

chosen such that the MD functionality is spread efficiently among all nodes in the network, while maintaining a low duty cycle for each individual device.

6.1 How to avoid collisions?

Collision is an issue when using the distributed MD method. *What if more than one device enters MD mode simultaneously?* There is no problem for the case of more than one device listening at the same time, but the devices may interfere with each other if they talk at the same time.

To avoid collisions, a device sends out an Alarm message at the end of its MD receive period, as shown in Figure 10. If any other device is also in MD mode at this moment, it will receive the Alarm message and exit MD mode immediately. It will then wait a random period of time and return to normal mode.

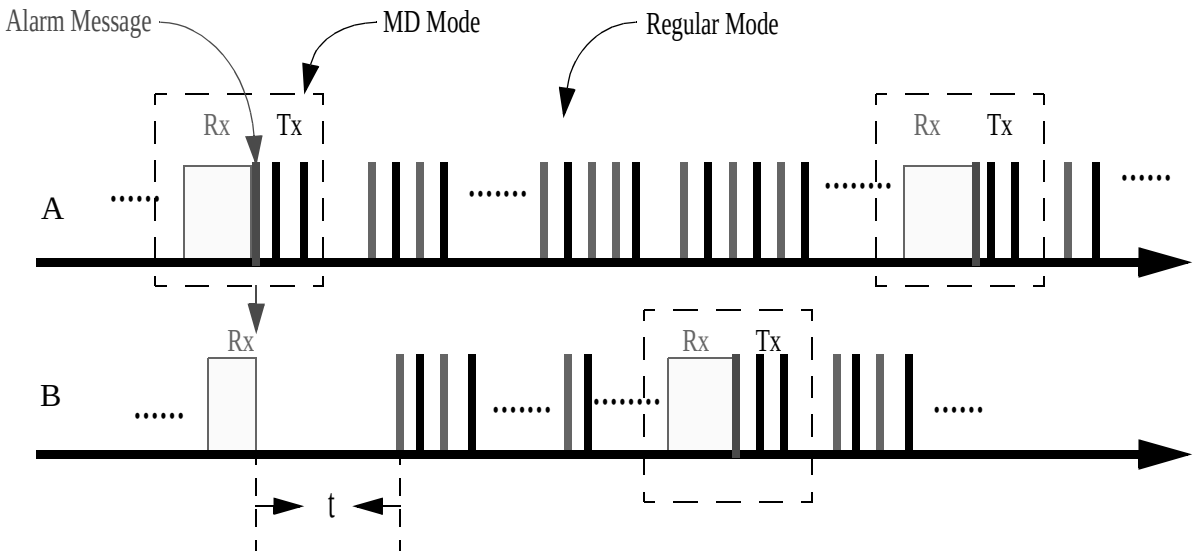

Figure 10: Collision Avoidance for Distributed MD

In Figure 10,

1). Device A enters MD mode.
2). Then, less than 2 seconds later, B enters MD mode.
3). At the end of A's MD receiving period, A sends out an ALARM message to ask other MD devices to return to normal mode.
4). During B's MD receiving period, B receives the ALARM from A. Once B receives this Alarm, it will exit MD mode, sleep a random time t to avoid future collisions with A, and re-enter normal mode.

7.0 MD for small network

The distributed MD protocol works well for large networks having more than ten nodes. However, for a small network of less than five nodes, message latency may become very high. For instance, for the Distributed MD method, the average latency is about 2000 / 5 = 400 Seconds when there are five nodes in the network.

How to reduce the latency for a small network?

For a small network, the communication window can be reduced. For instance, all nodes can be required to communicate within a 100 ms window, as illustrated in Figure 11.

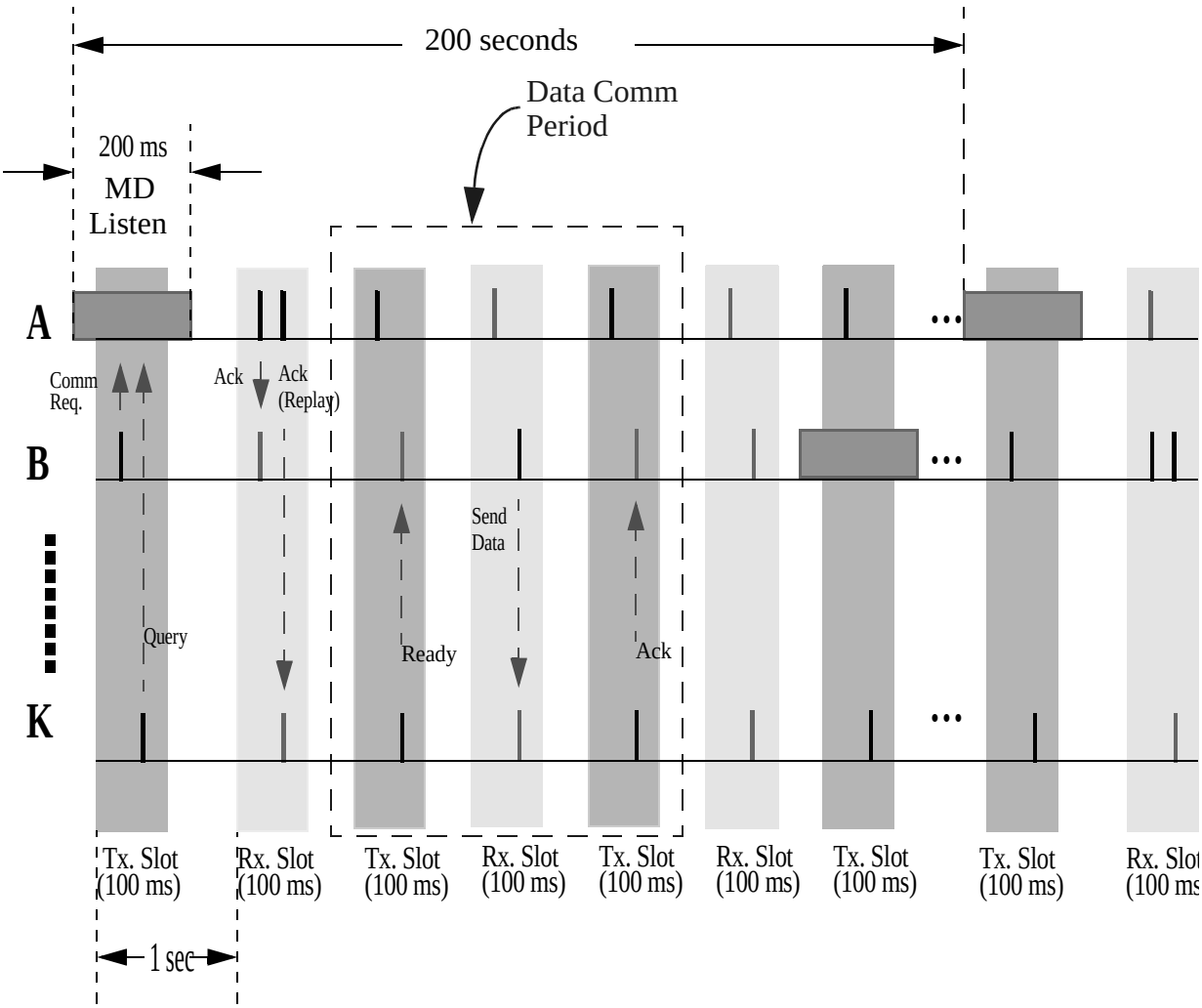

Figure 11: Timing Option to Reduce Latency in a Small Network

In Figure 11,

1.) As shown, all communications occur within special time slots that are 100 ms long. The slots are alternately defined as Transmit (Tx) Slot and Receive (Rx) Slot, and the gap between each slot is 900ms. The slots are named from the point of view of a device querying the MD.

2). The MD can ***only*** listen during the Tx Slot as shown above. Therefore, the MD listening period must be 200 ms to guarantee covering the ***whole*** Tx Slot. A device re-enters MD mode about every 200 seconds.

3). The routine Query and Comm Req can only occur during the Tx Slot. Both message types have a duration of 1 ms.

4). The Ack and Replay from the MD to the other devices occur during the next Rx Slot.

5). Usually the data communication period includes a Ready for data message, a Send Data message, and an Ack. These three messages are sent / received in consecutive time slots. The Ready message is sent from device K to B during a Tx slot. Then B responds with the Send Data message during the following Rx slot. The sequence concludes with B replying with an Ack during the next Tx slot.

6). After the data communication, all the devices go back to the routine Tx-Rx-Tx-Rx... modes.

In a synchronous scheme, the exact timing information of all the nodes is known. In contrast, nothing is known about timing in an asynchronous scheme. The scheme described above is a new multiple access scheme for which the exact timing is not known, but the timing within a 100 ms range is known.

The issues with this scheme are:
1). Nodes get lost (node transmissions drift out of the appropriate 100 ms communication window)
2). New nodes join the network (new nodes are not synchronized to the 100 ms slots)

To overcome these issues, a long MD mode (2002 ms listening period) is needed once in a while. For instance, every 2000 seconds the MD receive period can be 2002 ms long.

An overall timing schedule for any device of a small network is illustrated in Figure 12:

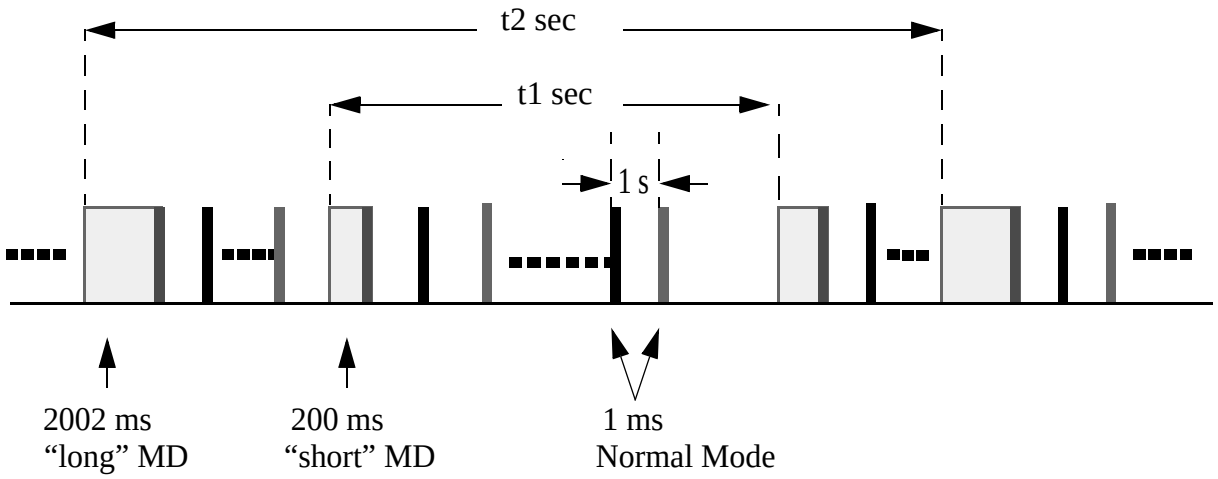

Figure 12: Timing Option for a Single Device in a Small Network

A reduced latency, small network has
1). a 1 ms communication (Tx or Rx) every second,
2). a 200 ms short MD mode every t1 seconds (t1 is a random time, 150 < t1 < 250),
3). a 2002 ms long MD mode every t2 seconds (t2 is a random time, 1500 < t2 < 2500).

The overall duty cycle is about 0.25%. The average latency is about 200 / N seconds, where N is the number of the devices in the network. For example, the latency is 40 seconds for N=5.