1 One Hop IoT Network over IEEE 802.15.4 (Level 2)

1.1 Introduction

The concept of Cyber Physical Systems (CPS) over the Internet of Things (IoT) was explained in Experiment -19: Cyber physical systems (CPS) and IoT. In most situations, due to the practical difficulty of laying copper or optical cables to connect sensors and actuators, digital wireless communication has to be used. In such applications, since the energy available in the sensor and actuator devices is small, there is a need for keeping costs low, and the communication performance requirement (in terms of throughput and delay is limited) several wireless technologies have been developed, with the IEEE 802.15.4 standard being one of the early ones.

The IEEE Standard 802.15.4 defines PHY and MAC protocols for low-data-rate, low-power, and low-complexity short-range radio frequency (RF) digital transmissions.

In this experiment, we will study the simplest IEEE 802.15.4 network with one wireless node transmitting packets to an IEEE 802.15.4 receiver, from where the packets are carried over a high-speed wireline network to a compute server (where the sensor data would be analyzed).

1.1.1 The IEEE 802.15.4 PHY and MAC

We will study the IEEE 802.15.4 standard that works in the 2.4 GHz ISM band, in which there is an 80 MHz band on which 16 channels are defined, each of 2 MHz, with a channel separation of 5 MHz. Each IEEE 802.15.4 network works in one of these 2 MHz channels, utilizing spread spectrum communication over a chip-stream of 2 million chips per second. In this chip-stream, 32 successive chips constitute one symbol, thereby yielding 62,500 symbols per second (62.5 Ksps; $\frac{(2\times10^6)}{32} = 62,500$). Here, we observe that a symbol duration is $32 \times \frac{1}{2\times10^6} = 16\ \mu$sec. Binary signaling (OQPSK) is used over the chips, yielding 2^{32} possible sequences over a 32 chip symbol. Of these sequences, 16 are selected to encode 4 bits ($2^4 = 16$). The sequences are selected so as to increase the probability of decoding in spite of symbol error. Thus, with 62.5 Ksps and 4 bits per symbol, the IEEE 802.15.4 PHY provides a raw bit rate of $62.5 \times 4 = 250$ Kbps.

Having described the IEEE 802.15.4 PHY, we now turn to the MAC, i.e., the protocol for sharing the bit rate of an IEEE 802.15.4 shared digital link. A version of the CSMA/CA mechanism is used for multiple access. When a node has a data packet to send, it initiates a random back-off with the first back-off period being sampled uniformly from 0 to $(2^{macminBE} - 1)$, where $macminBE$ is a standard parameter. The back-off period is in slots, where a slot equals 20 symbol times, or $20 \times 16 = 320\,\mu\text{sec}$. The node then performs a *Clear Channel Assessment* (CCA) to determine whether the channel is idle. A CCA essentially involves the node listening over 8 symbols times and integrating its received power. If the result exceeds a threshold, it is concluded that the channel is busy and CCA fails. If the CCA succeeds, the node does a Rx-to-Tx turn-around, which takes 12 symbol times and starts transmitting on the channel. The failure of the CCA starts a new back-off process with the back-off exponent raised by one, i.e., to macminBE+1, provided it is less than the maximum back-off value, macmaxBE. The maximum number of successive CCA failures for the same packet is governed by macMaxCSMABackoffs; if this limit is exceeded the packet is discarded at the MAC layer. The standard allows the inclusion of acknowledgements (ACKs) which are sent by the intended receivers on a successful packet reception. Once the packet is received, the receiver performs a Rx-to-Tx turnaround, which is again 12 symbol times, and sends a 22-symbol fixed size ACK packet. A successful transmission is followed by an InterFrameSpace (IFS) before sending another packet.

The IEEE 802.15.4 can operate either in a beacon enabled or a nonbeacon enabled mode. In the beacon enabled mode, the PAN coordinator works with time slots defined through a superframe structure (see Figure 1-1). This permits a synchronous operation of the network. Each superframe has active and inactive portions. The PAN Coordinator interacts with the network only during the active portion. The active portion is composed of three parts: a beacon, a contention access period (CAP), and a contention free period (CFP). The active portion starts with the transmission of a beacon and a CAP commences immediately after the beacon. All frames, except acknowledgment frames and any data frame that immediately follows the acknowledgment of a data request command (as would happen following a data request from a node to the PAN coordinator), transmitted in the CAP, must use a slotted CSMA/CA mechanism to access the channel.
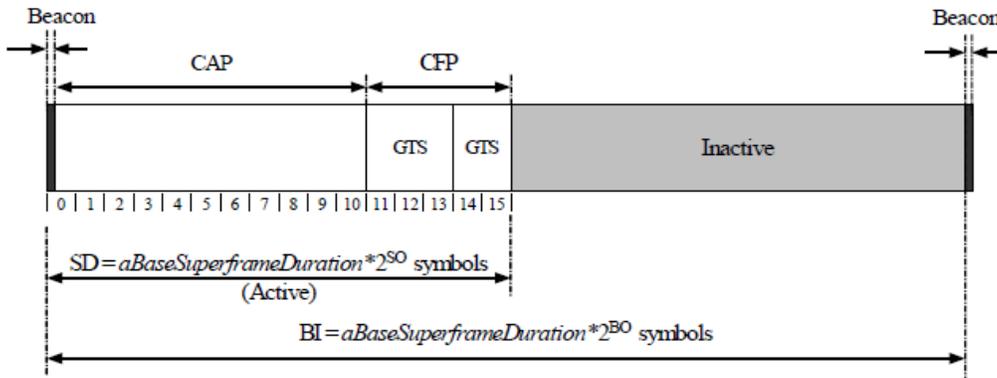
Figure 1-1: The IEEE 802.15.4 Frame Structure

When a transmitted packet collides or is corrupted by the PHY layer noise, the ACK packet is not generated which the transmitter interprets as packet delivery failure. The node reattempts the same packet for a maximum of a Max FrameRetries times before discarding it at the MAC layer. After transmitting a packet, the node turns to Rx-mode and waits for the ACK. The macAckWaitDuration determines the maximum amount of time a node must wait to receive the ACK before declaring that the packet (or the ACK) has collided. The default values of macminBE, macmaxBE, macMaxCSMABackoffs, and a Max FrameRetries are 3, 5, 4, and 3.

1.1.2 Objectives of the Experiment

In Section 1.1.1, we saw that the IEEE 802.15.4 PHY provides a bit rate of 250 Kbps, which has to be shared among the nodes sharing the 2 MHz channel on which this network runs. In the simulation experiment, the packets will have an effective length of 109 bytes (109 B = 872 bits). Thus, over a 250 Kbps link, the maximum packet transmission rate is $\frac{250 \times 1000}{872} = 286.70$ packets per second. We notice, however, from the protocol description in Section 1.1.1, that due to the medium access control, before each packet is transmitted the nodes must contend for the transmission opportunity. This will reduce the actual packet transmission rate well below 286.7.

In this experiment, just one node will send packets to a receiver. Since there is no contention (there being only one transmitter) there is no need for medium access control, and packets could be sent back-to-back. However, the MAC protocol is always present, even with one node, and we would like to study the maximum possible rate at which a node can send back-to-back packets, when it is the only transmitter in the network. Evidently, since there is no uncertainty due to contention from other nodes, the overhead between the packets can be calculated from the protocol description in Section 1.1.1. This has been done in Section 1.1.5.

This analysis will provide the maximum possible rate at which a node can send packets over the IEEE 802.15.4 channel. Then in Section 1.1.6, we compare the throughput obtained from the simulation with that obtained from the analysis. In the simulation, in order to ensure that the node sends at the maximum possible rate, the packet queue at the transmitting node never empties out. This is ensured by inserting packets into the transmitting node queue at a rate higher than the node can possibly transmit.

1.1.3 NetSim Simulation Setup

Open NetSim and click on **Experiments>IOT-WSN> One Hop IoT Network over IEEE 802.15.4** click on the tile in the middle panel to load the example as shown in below Figure 1-2.
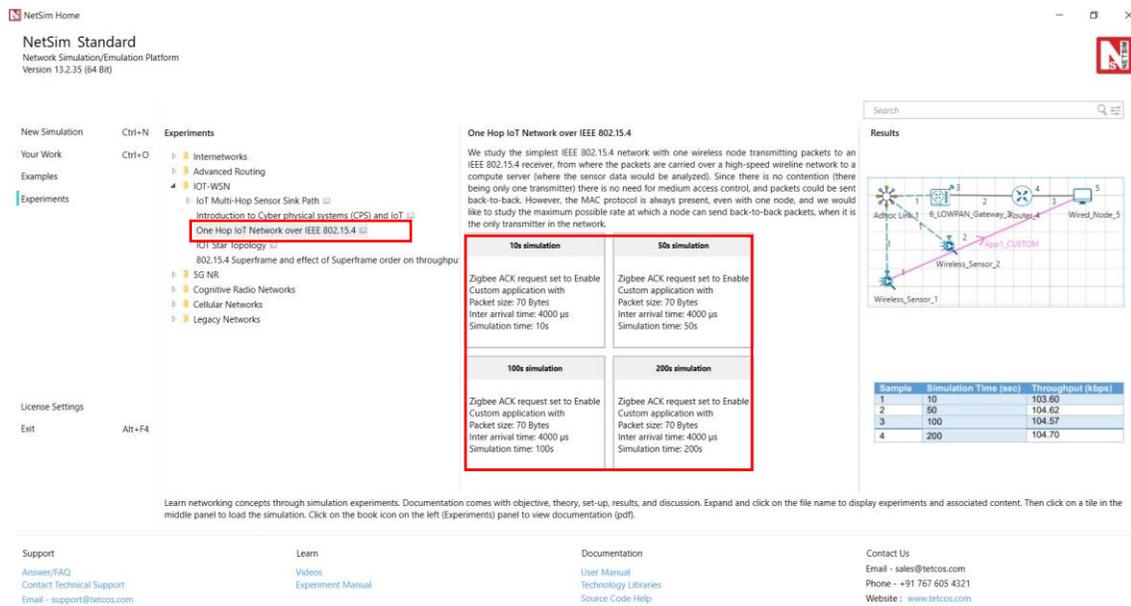
Figure 1-2: List of scenarios for the example of One Hop IoT Network over IEEE 802.15.4

10s Simulation sample

NetSim UI displays the configuration file corresponding to this experiment as shown below Figure 1-3.
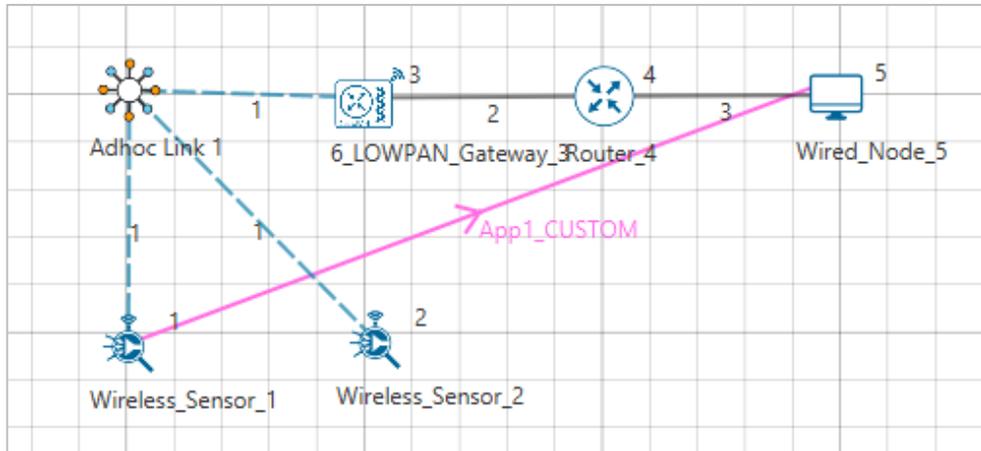
Figure 1-3: Network set up for studying the One Hop IoT Network over IEEE 802.15.4

1.1.4 Simulation Procedure

The following set of procedures were done to generate this sample:

Step 1: A network scenario is designed in NetSim GUI comprising of 2 Wireless Sensors, a 6 LOWPAN Gateway, 1 Router, and 1 Wired Node.

Step 2: In the Interface Zigbee > Data Link Layer of Wireless Sensor 1, **Ack Request** is set to Enable and **Max Frame Retries** is set to 7. It will automatically be set for Wireless Sensor 2, since the above parameters are Global.

Step 3: In the Interface Zigbee > Data Link Layer of 6 LOWPAN Gateway, **Beacon Mode** is set to Disable by default.

Step 4: The Ad hoc link properties are set to **NO PATHLOSS** for the channel characteristics.

Step 5: Right click on the Application Flow **App1 CUSTOM** and select Properties or click on the Application icon present in the top ribbon/toolbar.

A Custom Application is set from Wireless Sensor 1 i.e., Source to Wired Node 5 i.e., Destination. Transport Protocol is set to **UDP** with Packet Size set to 70 Bytes and Inter Arrival Time set to 4000 µs. The Packet Size and Inter Arrival Time parameters are set such that the Generation Rate equals 140 Kbps. Generation Rate can be calculated using the formula:

$$Generation\ Rate\ (Mbps) = Packet\ Size\ (Bytes) * 8/Interarrival\ time\ (µs)$$

NOTE: *If the size of the packet at the Physical layer is greater than 127 bytes, the packet gets fragmented. Taking into account the various overheads added at different layers (which are mentioned below), the packet size at the application layer should be less than 80 bytes.*

Step 6: Plots are enabled in NetSim GUI. Run simulation for 10 Seconds and note down the throughput.

Similarly, do the other samples by increasing the simulation time to 50, 100, and 200 Seconds respectively and note down the throughputs.

1.1.5 Analysis of Maximum Throughput

We have set the Application layer payload as 70 bytes in the Packet Size and when the packet reaches the Physical Layer, various other headers get added like Table 1-1.

App layer Payload	70 bytes
Transport Layer Header	8 bytes
Network Layer Header	20 bytes
MAC Header	5 bytes
PHY Header (includes Preamble, and Start Packet Delimiter)	6 bytes
Packet Size	109 bytes

Table 1-1: Overheads added to a packet as it flows down the network stack

By default, NetSim uses Unslotted CSMA/CA and so, the packet transmission happens after a Random Back Off, CCA, and Turn-Around-Time and is followed by Turn-Around-Time and ACK Packet and each of them occupies specific time set by the IEEE 802.15.4 standard as per the timing diagram shown below Figure 1-4.
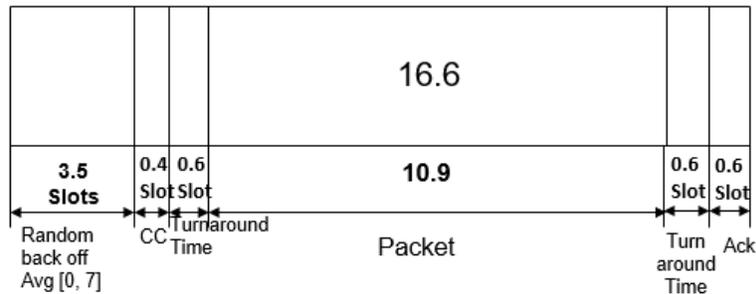
Figure 1-4: Zigbee timing diagram

From IEEE standard, each slot has 20 Symbols in it and each symbol takes 16µs for transmission.

Symbol Time	T_s	16 µs
Slot Time	$20 * T_s$	0.32
Random Backoff	$3.5 * Slots$	1.12

CCA	0.4 * $Slots$	0.128
Turn-around-Time	0.6 * $Slots$	0.192
Packet Transmission	10.9 * $Slots$	3.488
Turn-around-Time	0.6 * $Slots$	0.192
ACK Packet Time	0.6 * $Slots$	0.192
Total Time	16.6 * $Slots$	5.312

Table 1-2: Symbol times as per IEEE standard

$$Analytical\ Application\ Throughput\ = \frac{70(bytes)inApplayer * 8}{5.312\ ms} = 105.42\ kbps$$

1.1.6 Comparison of Simulation and Calculation

Throughput	from	104.74
Throughput	from	105.42

Table 1-3: Results comparison form simulation and theoretical analysis

Throughput from theoretical analysis matches the results of NetSim's discrete event simulation. The slight difference in throughput is due to two facts that

- The average of random numbers generated for backoff need not be exactly 3.5 as the simulation is run for short time.
- In the packet trace one can notice that there are OSPF and AODV control packets (required for the route setup process) that sent over the network. The data transmissions occur only after the control packet transmissions are completed.

As we go on increasing the simulation time, the throughput value obtained from simulation approaches the theoretical value as can be seen from the table below Table 1-4.

Sample	Simulation Time (sec)	Throughput (kbps)
1	10	103.60
2	50	104.62
3	100	104.57
4	200	104.70

Table 1-4: Throughput comparison with different simulation times