CHAPTER 4

SYSTEM OVERVIEW

4.1 WBAN ARCHITECTURE

A WBAN is a 3-tier system with the first tier forming the sensor nodes located over a person's body. The second tier is the personal server containing the coordinator located nearer to the nodes. The third tier is medical server which is responsible for monitoring the health of the wearer. The sensor nodes connect to the personal server and then through the Internet to a medical server tier that resides at the top of this hierarchy. The system is not merely a distributed data logger, which in itself would provide great advantage over current systems, but provides distributed data processing and analysis functions.

Each tier in the network is intelligent and provides some form of analysis; in some cases it may be possible for on-the-spot real-time diagnosis of conditions. The third tier, centered on a medical server, is optimized to service hundreds or thousands of individual users, and encompasses a complex network of interconnected services, medical personnel and healthcare professionals. Each user wears a number of sensor nodes that are strategically placed on the body. The nodes are designed to unobtrusively sample vital signs and transfer the relevant data to a personal server through a wireless personal network implemented using ZigBee (802.15.4) or Bluetooth (802.15.1). The personal server, implemented on a home personal computer, handheld computer, smart phone, or residential gateway, controls the WBAN, performs sensor fusion, and preliminary analysis of physiological data. It provides graphical or audio interface to the user,
and transfers captured health information to the medical server through the Internet or mobile telephone networks (e.g., GPRS, 3G). Figure 4.1 shows the architecture of a WBAN.

![Figure 4.1: Architecture of WBAN](image)

### 4.1.1 Medical Server

The medical server provides a variety of differing functions to WBAN users, medical personnel and informal caregivers. The medical server stores electronic patient records in a database, provides a high availability daemon for authenticating registered WBAN users and accepting session uploads, summarizes physiological data and automatically analyzes the data to verify it is inside or outside acceptable health metrics (heart rate, blood pressure, activity) and identifies known patterns of health risks. It is the responsibility of the medical server to interface the electronic patient records and insert new session data, generate alerts to the physician and
emergency health care professionals when abnormal conditions are detected, and provide physician and informal caregiver portals via the Internet for retrieving health summary reports remotely. This is especially powerful for the physician who can access the data at a convenient time to determine whether the patient is responding to a prescribed medication or exercise and make updates to those prescriptions and forward them electronically back to the patient where the user's personal server is responsible for delivering such changes to the user.

The large amount of data collected through these services can also be utilized for knowledge discovery through data mining. Integration of the collected data into research databases along with quantitative analysis of conditions and patterns could prove invaluable to researchers trying to link symptoms and diagnoses with historical changes in health status, physiological data, or other parameters (e.g., gender, age, weight). In a similar way this infrastructure could significantly contribute to monitoring and studying of drug therapy effects.

4.1.2 Personal Server

The personal server, at the second tier, is responsible for

- Interfacing with the medical server via the Internet
- Interfacing the WBAN sensors and fusing sensor data
- Providing an intuitive graphical and/or audio interface to the end user.

The personal server application can run on a variety of platforms with a variety of wide area network (WAN) access possibilities for Internet access. Platform selection is system specific and should be selected to minimize complexity for a given user. For in-home monitoring of elderly patients, a stationary residential gateway or personal computer might be the ideal platform. But
for high mobility users, it may be necessary to use a smart phone or handheld computer with GPRS capabilities.

The personal server requires ZigBee or Bluetooth capability for communications within the WBAN. Depending on the platform this may be integrated in the device or provided as a separate plug-in network coordinator (NC). The NC is responsible for coordinating WBAN communications and managing aspects such as time synchronization, timeslot assignment, and channel sharing. In addition, the personal server is responsible for sensor configuration including node registration (type and number of sensors), initialization (e.g., specify sampling frequency and mode of operation), customization (e.g., run user-specific calibration or user-specific signal processing procedure upload) and setup of a secure communication (key exchange). Once the sensor nodes are configured, the personal server fuses sensor data into personalized session files. Based on synergy of information from the multiple medical sensors, the PS application should determine the user's state and his or her health status providing user feedback through a friendly and intuitive graphical or audio user interface.

For interface to the medical server, the personal server requires some wireless wide area network (WWAN) or wireless local area network (WLAN) access such as GPRS or 802.11 respectively. In the case of a static residential gateway or home personal computer implementation, the personal server may be connected directly to a broadband Internet link. The personal server holds patient authentication information and is configured with IP address or domain name of the medical server so that it can access services over the Internet. The PS schedules upload of health monitoring session files at periodic intervals or defers transmission in the event an Internet connection is unavailable. In such cases, the personal server may be unable
to propagate indicators of serious changes in health status. Because processing is performed on
the personal server and on sensor nodes, the system should be capable of recognizing
abnormalities and alerting the user to potential threatening physiological conditions.

4.1.3 Sensor Nodes

For every personal server, a network of intelligent sensor nodes captures various
physiological signals of medical interest. Each node is capable of sensing, sampling, processing,
and communicating physiological signals. For example, an ECG sensor can be used for
monitoring heart activity, an EMG sensor for monitoring muscle activity, an EEG sensor for
monitoring brain electrical activity, a blood pressure sensor for monitoring blood pressure, a tilt
sensor for monitoring trunk position, a breathing sensor for monitoring respiration, while the
motion sensors can be used to discriminate the user's status and estimate her or his level of
activity. Each sensor node receives initialization commands and responds to queries from the
personal server. WBAN nodes must satisfy requirements for minimal weight, miniature form-
factor, low-power consumption to permit prolonged ubiquitous monitoring, seamless integration
into a WBAN, standards based interface protocols, and patient-specific calibration, tuning, and
customization. With further development of the technology, the wireless network nodes can be
implemented as tiny patches or incorporated into the user's clothes. The network nodes
continuously collect and process raw information, store them locally and send processed event
notifications to the personal server. The type and nature of a healthcare application will
determine the frequency of relevant events (sampling, processing, storing, and communicating).

Ideally, sensors process data on-sensor, minimizing the number of data transmissions,
therefore significantly reducing power consumption and extending battery life. When local
analysis of data is inconclusive or indicates an emergency situation, the node can transfer raw signals to the next tier of the network for further processing. Patient privacy, an outstanding issue and a requirement by law, must be addressed at all tiers in the healthcare system. Data transfers between a user's personal server and the medical server require encryption of all sensitive information related to the personal health. Before possible integration of the data into research databases, all records must be stripped of all information that can tie it to a particular user. The limited range of wireless WBAN communications partially addresses security. In addition, the messages can and should be encrypted using either software or hardware techniques. Some wireless sensor platforms have already provided a low power hardware encryption solution for ZigBee communications.

4.2 SPECIFICATIONS AND STANDARDS

The three major standards or specifications used in a WBAN are Zigbee, GPRS/GSM and GPS. Zigbee is used for short range communication between the sensors and also between the sensors and the personal server. GPRS/GSM is for sending messages over the telephone line. If the personal server is a cell phone or if a message needs to be sent to the doctors phone GPRS/GSM are the technologies used. GPS technology is for finding the correct location of the patient. The location information is encrypted and sent to the server as it would compromise the privacy of the patient if it is captured by an adversary. In the following sections we explain these three technologies in detail.
4.2.1 Zigbee

ZigBee is a worldwide standard specification of a suite of communication protocols which has been developed by the ZigBee Alliance [Zigbee]. The ZigBee Alliance is an association of international companies that develops the standard. The suite of communication protocols is based on the IEE 802.15.4-2003 standard for wireless personal area networks (WPAN). The goal with the ZigBee wireless standard is to provide a short-range, low-power, low-rate, low-cost and secure networking technology. One feature that supports the low-power consumption of the technology is that ZigBee can spend time in sleep mode most of the time in some applications. A device can enter sleep mode when there is no data to receive or transmit. The wake up delay - going from sleep mode to wake mode - is very small in ZigBee, i.e., 15mS. Particularly compared to Bluetooth which has a wake up delay around three seconds. IEE 802.15.4-2003 which ZigBee has as foundation uses three different frequency bands which together are divided into 26 channels. One of the operating frequency bands is 2.4GHz and is used worldwide. The other two have lower frequencies and are specific for Europe, America and Australia.

In a ZigBee wireless network there can co-exist three different nodes named ZigBee Coordinator (ZC), ZigBee Router (ZR) and ZigBee End Device (ZED). There can only be one ZC in each network. The ZC can relay messages between devices in the network, but it is also responsible for startup and initialization of the network, select appropriate frequency channel, allow other nodes to join the network and handle security management. In comparison, the ZR has no other responsibilities but relaying messages between other devices in the network. As implicated from the name, the ZED is located at the end (or edge) of the network and it only
transmits and receives messages to and from routers and the coordinator. The real
communication coverage distance for ZigBee varies depending on the application. Some factors
that affect the maximum communication distance are; type of antenna, antenna gain, transmit
power and background noise. The ZigBee standard support three different network topologies,
these are:

- **Star Topology**

  In this configuration, which is the simplest and most limited, nodes use the
  coordinator as a central hub to exchange messages. Thus, there is only one coordinator
  that is responsible of relaying messages between communicating nodes.

- **Tree Topology**

  In this topology, the coordinator operates at the root of the tree as a parent and can
  have many nodes (Routers/End Devices) as children. At lower levels of the tree hierarchy
  routers and end devices can be found, where only routers can act as parents and have
  children of their own further down in the hierarchy. Each child node can only
  communicate with its parent node which, in turn, can communicate with its parent node.

- **Mesh Topology**

  This topology builds upon the tree topology to add more flexibility where nodes
  on the same level of the tree or within range can communicate directly, without the need
  to exchange messages through the closest ancestor. The Mesh topology gives rise to more
efficient message propagation, and means that alternative routes can be found if a link fails or there is congestion.

4.2.2 GSM/GPRS

GSM is originally the abbreviation of Group Special Mobile but is nowadays the abbreviation of Global System for Mobile Communications. It is the most popular standard for mobile communications with over 3 billion users across the world and is referred to as the second generation (2G) mobile phone system. The main difference between the second and the first generation of mobile phone systems is that in the 2G system signaling and voice channels are digital compared to being analog in 1G. GPRS is the abbreviation of General Packet Radio Service and is a later added service to the GSM standard. GPRS is a packet based service compared GSM which only uses circuit switched services like voice calls and circuit switched data. A difference between circuit switched and packet based data traffic is which type of billing that is used. Packet based traffic is typically charged per megabyte of transferred data while circuit switched traffic is charged per minute of connection time. Another difference between the two technologies is that in GPRS there is a best-effort service while GSM guarantee a certain quality of service (QoS).

There are three different classes of devices supporting GPRS, these are:

- **Class A**: Has the ability to simultaneously connect to both GSM and GPRS services.
- **Class B**: Has the ability to use both GSM and GPRS services but only one at a time. During a GSM connection GPRS is suspended and later resumed again automatically.
- **Class C**: As class B, but the switching between GPRS and GSM must be done manually.
Furthermore, a GPRS connection is established by specifying an access point name (APN).

4.2.3 GPS System

The first GPS satellite was launched in 1978 and the GPS system was operational in 1993. The system consists of over 24 GPS satellites with a medium earth orbit (MEO) of 20 000 km. There are six orbital planes with at least four satellites in each plane. The idea is that each satellite will pass over the same point two times every 24 hours and that at least four satellites will be in the line of sight (LOS) over the horizon wherever you are on the planet. This is almost true since the system has a real-time coverage of 99.9%. Each satellite broadcasts two different L-band frequencies which are used when the distance to a GPS receiver is calculated by using the simple formula distance = (speed of light)*time. Four atomic clocks are used on each satellite to give an as precise time estimation as possible. To be able to triangulate a GPS receiver and obtain the correct coordinates there has to be four satellites within reach of the receiver, which will be the case 99.9% of the time as long as the receiver resides outdoor in LOS to the satellites.

When the position of the receiver is calculated, two different techniques can be used which are called pseudo-range and carrier-phase measurements. The most commonly used is the carrier-phase measurement which is using the two unmodulated carriers of the two frequencies that are transmitted from the GPS satellites. It is the phase difference between the signal(s) transmitted from the GPS satellites and an internally generated signal in the receiver that is measured. The phase difference is used to decide the distance from the receiver to each of the GPS satellites and when that is done the position, or coordinates of the receiver, can be triangulated. The signal generated internally in the receiver is an exact copy of the carrier wave transmitted from the GPS satellite. Thereby in an ideal world, there should be no phase
difference between the two signals. But there is a difference due to the Doppler Effect which Doppler shifts the signal when it propagates to the receiver. Most handheld GPS devices (receivers) around the world has a position accuracy of up to 10 meters horizontally and 15 - 20 meters vertically which is the case when absolute measurement is used, meaning one receiver and four satellites. The accuracy can be increased by doing relative measurement which is when one extra fixed receiver with a known position is used as a reference. From the difference between the position of the fixed receiver and the handheld receiver a baseline is computed. By doing this almost all error sources can be eliminated and thereby increase the accuracy.

The data transmitted between the GPS receiver and another technical unit, such as a computer or a mobile phone, is communicated by using the standard NMEA 0183 or NMEA 2000 communication protocols. Most GPS receivers support the NMEA 0183 protocol. NMEA is the abbreviation of National Marine Electronics Association, which is an association that defines communication protocols used in marine communication technologies. There are several different NMEA messages transmitted from GPS receivers. In the NMEA 0183 standard, pre-defined ASCII sentences are used in a way that makes it simple to parse the content accurately. Some of the most popular sentences can be found in Table 4.1.

<table>
<thead>
<tr>
<th>Sentence Name</th>
<th>Sentence Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>$GPRMC</td>
<td>Recommended Minimum Specific GPS/TRANSIT Data</td>
</tr>
<tr>
<td>$GPGGA</td>
<td>Global Positioning System Fix Data</td>
</tr>
<tr>
<td>$GPZDA</td>
<td>UTC Date/Time and Local Time Zone Offset</td>
</tr>
<tr>
<td>$GPGLL</td>
<td>Geographic Position, Latitude and Longitude</td>
</tr>
<tr>
<td>$GPVTG</td>
<td>Course Over Ground and Ground Speed</td>
</tr>
<tr>
<td>$GPGSV</td>
<td>GPS Satellites in View</td>
</tr>
</tbody>
</table>

Table 4.1: Some of the sentences used in the NMEA 0183 standard
The most widely used NMEA sentence in Table 4.1 is $GPRMC$, which contains most of the useful data such as; speed, geographical position and UTC time. UTC is the abbreviation of Coordinated Universal Time which is used for civil time keeping and is a time standard based on the international atomic time (TAI). The geographical position is given in longitude and latitude and the speed in meters per second.

4.3 TINYOS

The OS used in sensor nodes is TinyOS [Phi05]. TinyOS is a tiny (fewer than 400 bytes), flexible operating system built from a set of reusable components that are assembled into an application-specific system. TinyOS supports an event-driven concurrency model based on split-phase interfaces, asynchronous events, and deferred computation called tasks. TinyOS is implemented in the NesC language [Gay03], which supports the TinyOS component and concurrency model as well as extensive cross-component optimizations and compile-time race detection. TinyOS has enabled both innovation in sensor network systems and a wide variety of applications. TinyOS has been under development for several years and is currently in its third generation involving several iterations of hardware, radio stacks, and programming tools. Over one hundred groups worldwide use it, including several companies within their products.

TinyOS has a component-based programming model, codified by the NesC language, a dialect of C. TinyOS is not an OS in the traditional sense; it is a programming framework for embedded systems and set of components that enable building an application-specific OS into each application. A typical application is about 15K in size, of which the base OS is about 400 bytes; the largest application, a database-like query system, is about 64K bytes.
A TinyOS program is a graph of components, each of which is an independent computational entity that exposes one or more interfaces. Components have three computational abstractions: commands, events, and tasks. Commands and events are mechanisms for inter-component communication, while tasks are used to express intra-component concurrency. A command is typically a request to a component to perform some service, such as initiating a sensor reading, while an event signals the completion of that service. Events may also be signaled asynchronously, for example, due to hardware interrupts or message arrival. From a traditional OS perspective, commands are analogous to downcalls and events to upcalls. Commands and events cannot block: rather, a request for service is split phase in that the request for service (the command) and the completion signal (the corresponding event) are decoupled. The command returns immediately and the event signals completion at a later time. Rather than performing a computation immediately, commands and event handlers may post a task, a function executed by the TinyOS scheduler at a later time. This allows commands and events to be responsive, returning immediately while deferring extensive computation to tasks. While tasks may perform significant computation, their basic execution model is run-to-completion, rather than to run indefinitely; this allows tasks to be much lighter-weight than threads. Tasks represent internal concurrency within a component and may only access state within that component. The standard TinyOS task scheduler uses a non-preemptive, FIFO scheduling policy.

TinyOS abstracts all hardware resources as components. For example, calling the getData() command on a sensor component will cause it to later signal a dataReady() event when the hardware interrupt fires. While many components are entirely software-based, the combination of split-phase operations and tasks makes this distinction transparent to the programmer. For example, consider a component that encrypts a buffer of data. In a hardware
implementation, the command would instruct the encryption hardware to perform the operation, while a software implementation would post a task to encrypt the data on the CPU. In both cases an event signals that the encryption operation is complete.

The current version of TinyOS provides a large number of components to application developers, including abstractions for sensors, single-hop networking, ad-hoc routing, power management, timers, and non-volatile storage. A developer composes an application by writing components and wiring them to TinyOS components that provide implementations of the required services. Many different components may implement a given interface.

4.4 WBAN PROTOCOLS

4.4.1 IEEE 802.15.4 PHY/MAC

This standard defines the physical (PHY) and medium access control (MAC) layers for low data-rate, shortrange wireless communication. The 802.15.4 standard supports raw data throughput of 250 kbps and can transmit point-to-point anywhere from tens to hundreds of metres, depending on the output power and receive sensitivity of the transceiver. These chips can come as transceivers, system-on-chips, or in a network processor (NP) form factor with a pre-programmed network protocol. The 802.15.4 PHY/MAC is the underlying protocol for ZigBee, 6LoWPAN and RF4CE. Among other things, it defines basic network start-up, device discovery and joining, security, and acknowledged unicast and broadcast communication.
4.4.1.1 IEEE 802.15.4 PHY

The PHY provides two services: the PHY data service and PHY management service interfacing to the physical layer management entity (PLME). The PHY data service enables the transmission and reception of PHY protocol data units (PPDU) across the physical radio channel. The features of the PHY are activation and deactivation of the radio transceiver, energy detection (ED), link quality indication (LQI), channel selection, clear channel assessment (CCA) and

<table>
<thead>
<tr>
<th>PHY (MHz)</th>
<th>Frequency band (MHz)</th>
<th>Spreading parameters</th>
<th>Data parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Chip rate (kchips/s)</td>
<td>Modulation</td>
</tr>
<tr>
<td>868/915</td>
<td>868-868.6</td>
<td>300</td>
<td>BPSK</td>
</tr>
<tr>
<td></td>
<td>902-928</td>
<td>600</td>
<td>BPSK</td>
</tr>
<tr>
<td>2450</td>
<td>2400-2483.5</td>
<td>2000</td>
<td>O-QPSK</td>
</tr>
</tbody>
</table>

Table 4.2: IEEE 802.15.4 PHY

transmitting as well as receiving packets across the physical medium. The standard offers two PHY options based on the frequency band. Both are based on direct sequence spread spectrum (DSSS). The data rate is 250kbps at 2.4GHz, 40kbps at 915MHz and 20kbps at 868MHz. The higher data rate at 2.4GHz is attributed to a higher-order modulation scheme. Lower frequency provides longer range due to lower propagation losses. Low rate can be translated into better sensitivity and larger coverage area. Higher rate means higher throughput, lower latency or lower duty cycle. This information is summarized in table 4.2.
Both PHY layers use a common packet structure, enabling the definition of a common MAC interface. Each packet, or PHY protocol data unit (PPDU), contains a preamble, a start of packet delimiter, a packet length, and a payload field, or PHY service data unit. The 32-bit preamble is designed for acquisition of symbol and chip timing. The IEEE 802.15.4 payload length can vary from 2 to 127 bytes. This structure is shown in figure 4.2.

<table>
<thead>
<tr>
<th>PHY protocol data unit (PPDU)</th>
<th>Preamble</th>
<th>Start of packet delimiter</th>
<th>Length Field</th>
<th>PHY layer payload PHY service data unit (PSDU)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4 bytes</td>
<td>1 byte</td>
<td>1 byte</td>
<td>2-127 bytes</td>
</tr>
</tbody>
</table>

**Figure 4.2: IEEE 802.15.4 Packet Format**

There is a single channel between 868 and 868.6MHz, 10 channels between 902.0 and 928.0MHz, and 16 channels between 2.4 and 2.4835GHz. Several channels in different frequency bands enable the ability to relocate within spectrum. The standard also allows dynamic channel selection, a scan function that steps through a list of supported channels in search of beacon, receiver energy detection, link quality indication, channel switching. Receiver sensitivities are -85dBm for 2.4GHz and -92dBm for 868/915MHz. The advantage of 6-8dB comes from the advantage of lower rate. The achievable range is a function of receiver sensitivity and transmit power. The maximum transmit power shall conform to local regulations. A compliant device shall have its nominal transmit power level indicated by the PHY parameter, `phyTransmitPower`. 

79
4.4.1.2 IEEE 802.15.4 MAC

The IEEE 802 project divides the data link layer (DLL) into two sub layers, the MAC and logical link control (LLC) sub layers. The LLC is standardized in IEEE 802.2 and is common among the IEEE 802 standards. The IEEE 802.15.4 medium access control (MAC) sublayer controls the access to the radio channel employing the CSMA/CA mechanism. If upper layers detect that the communications throughput has been degraded below a determined threshold, the MAC will be instructed to perform an energy detection scan through the available channels. Based on the detected energy, the upper layers will switch to the channel with the lowest energy. The IEEE 802.15.4 performs the energy scan by the use of a clear channel assessment procedure. This can be performed by following either simple in-band energy detection above a threshold, or an IEEE 802.15.4 carrier detection or a combination of both. The 802.15.4 MAC is also responsible for flow control via acknowledged frame delivery, frame validations as well as maintaining network synchronization, controlling the association, administering device security and scheming the guaranteed time slot mechanism.

The general MAC frame format is given in Figure 4.3. Each MAC frame consists of the following basic components:

- MAC header (MHR), which comprises frame control, sequence number, and address information

- A MAC payload of variable length, which contains information specific to the frame type. Acknowledgement frames do not contain a payload.

- A MAC footer (MFR), which contains frame check sequence (FCS).
There are 4 frame structures: beacon frame, data frame, acknowledgement frame, MAC command frame. The optional use of a superframe structure is allowed for applications requiring dedicated bandwidth to guarantee communication latency. The format of the superframe is defined by the PAN coordinator, by using the network beacons which bound the superframe structure. The superframe is composed of 16 equally sized time slots grouped in two sections: the contention access period (CAP) and the contention free period (CFP). The time slots assigned for the CFP are called guaranteed time slots (GTS) and are administered by the PAN coordinator.

The IEEE 802.15.4 MAC can operate in two modes: beacon-enabled and beaconless. In the beacon-enabled mode, the PAN coordinator broadcasts a periodic beacon containing information about the PAN. The period between two consecutive beacons defines a superframe structure. A superframe is always initiated by the beacon, while the remainder may be used for data communication by means of random access, and form the so called CAP. The beacon contains information related to the PAN identification, synchronization, and superframe structure. In this case, two types of data transactions exist:
1) Transfer from a device to the coordinator - a device willing to transfer data to the coordinator uses slotted CSMA-CA. The coordinator may confirm the successful data reception with an optional acknowledgment following the data frame.

2) Transfer from the coordinator to a device - when the coordinator has data pending for a device, it announces so in the beacon. The interested device adopts slotted CSMA-CA to send a request to the coordinator, indicating that it is ready to receive the data. When the coordinator receives the data request message, it selects a free slot and sends data using slotted CSMA-CA as well. In order to support time critical data applications, the PAN coordinator can reserve one or more slots that are assigned to devices running such applications without need for contention with other devices. Such slots are referred to as GTSs, and they form the CFP of the superframe. Note that CFP cannot operate independently and is always integrated with CAP.

In the nonbeacon-enabled mode there is no explicit synchronization provided by the PAN coordinator. Since there is no superframe defined in the nonbeacon-enabled mode and no slot synchronization is available, no GTS can be reserved, and only random access is adopted for medium sharing. The CSMA-CA algorithm shall be used before the transmission of data or MAC command frames transmitted within the CAP. The IEEE 802.15.4 uses two types of CSMA-CA algorithms as follows.

In the case of a nonbeacon-enabled network, when a device needs to send data it picks a random backoff delay, defined as a multiple of a backoff time unit. When the backoff delay expires, the device performs a clear channel assessment (CCA) operation, consisting in listening to the channel in order to determine if it is idle. If the channel is idle the device immediately transmits the data packet; oppositely, if the channel is busy the device repeats the procedure by
picking a new backoff delay before trying to access the channel again. In a beacon enabled network the devices use a slotted version of the previous protocol to access the medium in the CAP portion of the superframe. The main difference compared to the unslotted version is that at the end of the random backoff delay the device performs a CCA operation at the beginning of the next backoff unit; if the channel is idle, however, the device does not transmit the data packet immediately, but repeats the CCA for a number of backoff units defined by the value of a parameter called contention window (CW). If the channel is idle for all the backoff units within the CW the device transmits the data. If during one of the units in the CW the channel is detected to be busy, the device repeats the procedure by picking a new backoff delay.

In both cases, the algorithm is implemented using units of time called backoff periods, where one backoff period shall be equal to a constant, i.e. $aUnitBackoffPeriod$. The maximum number of backoffs the CSMA-CA algorithm will attempt before declaring a channel access failure is specified as $macMaxCSMABackoffs$. Note that the CSMA-CA algorithm shall not be used for the transmission of beacon frames, acknowledgements, or data frames transmitted in the CFP.

According to the descriptions in this section, totally there are three types of channel access mechanism for IEEE 802.15.4 MAC, i.e., unslotted CSMA-CA, slotted CSMA-CA, and slotted CSMA-CA integrated with GTS. The first scheme is working in the beaconless mode and the remaining two schemes are both working in the beacon enabled mode.
4.4.2 Ad hoc On Demand Distance Vector Routing (AODV)

AODV [Char99] is a pure on-demand route acquisition algorithm: nodes that do not lie on active paths neither maintain any routing information nor participate in any periodic routing table exchanges. Further, a node does not have to discover and maintain a route to another node until the two need to communicate, unless the former node is offering services as an intermediate forwarding station to maintain connectivity between two other nodes. The primary objectives of the algorithm are to broadcast discovery packets only when necessary, to distinguish between local connectivity management and general topology maintenance and to disseminate information about changes in local connectivity to those neighboring mobile nodes that are likely to need the information. When a source node needs to communicate with another node for which it has no routing information in its table, the Path Discovery process is initiated. Every node maintains two separate counters: sequence number and broadcast id. The source node initiates path discovery by broadcasting a route request (RREQ) packet to its neighbors, which includes source addr, source sequence number, broadcast id, dest addr, dest sequence number, hop cnt. (Source sequence number is for maintaining freshness information about the reverse route whereas the destination sequence number is for maintaining freshness of the route to the destination before it can be accepted by the source.)

The pair source addr, broadcast id uniquely identifies a RREQ, where broadcast id is incremented whenever the source issues a new RREQ. When an intermediate node receives a RREQ, if it has already received a RREQ with the same broadcast id and source address, it drops the redundant RREQ and does not rebroadcast it. Otherwise, it rebroadcasts it to its own neighbors after increasing hop cnt. Each node keeps the following information: destination IP
address, source IP address, broadcast id, expiration time for reverse path route entry and source node’s sequence number. As the RREQ travels from a source to destinations, it automatically sets up the reverse path from all nodes back to the source. To set up a reverse path, a node records the address of the neighbor from which it received the first copy of RREQ. These reverse path route entries are maintained for at least enough time for the RREQ to traverse the network and produce a reply to the sender.

When the RREQ arrives at a node, possibly the destination itself, that possesses a current route to the destination, the receiving node first checks that the RREQ was received over a bi-directional link. If this node is not destination but has route to the destination, it determines whether the route is current by comparing the destination sequence number in its own route entry to the destination sequence number in the RREQ. If RREQ’s sequence number for the destination is greater than that recorded by the intermediate node, the intermediate node must not use this route to respond to the RREQ, instead rebroadcasts the RREQ. If the route has a destination sequence number that is greater than that contained in the RREQ or equal to that contained in the RREQ but a smaller hop count, it can unicasts a route reply packet (RREP) back to its neighbor from which it received the RREQ. A RREP contains the following information: source addr, dest addr, dest sequence number, hop cnt and lifetime. As the RREP travels back to the source, each node along the path sets up a forward pointer to the node from which the RREP came, updates its timeout information for route entries to the source and destination, and records the latest destination sequence number for the requested destination. Nodes that are along the path determined by the RREP will timeout after route request expiration timer and will delete the reverse pointers since they are not on the path from source to destination as shown in Figure 4.4.
Figure 4.4: AODV Forward and Reverse Path Formation

The value of this timeout time depends on the size of the ad hoc network. Also, there is the routing caching timeout that is associated with each routing entry to show the time after which the route is considered to be invalid. Each time a route entry is used to transmit data from a source toward a destination, the timeout for the entry is reset to the current time plus active-route-timeout. The source node can begin data transmission as soon as the first RREP is received, and can later update its routing information if it learns of a better route. Each routing table entry includes the following fields: destination, next hop, number of hops (metric), sequence number for the destination, active neighbors for this route, expiration time for the route table entry. For path maintenance, each node keeps the address of active neighbors through which packets for the given destination are received is maintained. This neighbor is considered active if it originates or relays at least one packet for that destination within the last active-timeout period.

Once the next hop on the path from source to the destination becomes unreachable (hello messages are not received for a certain time, hello messages also ensures that only nodes with
bidirectional connectivity are considered to be neighbors, therefore each hello message included the nodes from which the node has heard), the node upstream of the break propagates an unsolicited RREP with a fresh sequence number and hop count of 1 to all active upstream nodes. This process continues until all active source nodes are notified. Upon receiving the notification of a broken link, the source nodes can restart the discovery process if they still require a route to the destination. If it decides that it would like to rebuild the route to the destination, it sends out an RREQ with a destination sequence number of one greater than the previously known sequence number, to ensure that it builds a new, viable route and that no nodes reply if they still regard the previous route as valid.