Advances in Engineering Software 42 (2011) 513–528

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

A novel solution approach and protocol design for bio-telemetry applications

Musa Cibuk [a], Hasan H. Balik [b,*]

[a] Bitlis Eren University, Faculty of Engineering, Department of Computer Engineering, Turkey
[b] Istanbul Arel University, Faculty of Engineering and Architecture, Department of Electrical and Electronics Engineering, Turkey

ARTICLE INFO

Article history:
Received 3 February 2011
Received in revised form 30 March 2011
Accepted 31 March 2011
Available online 22 April 2011

Keywords:
Communication model
Protocol design
Biotelemetry
Medical networks
Telemedicine
Health monitoring
Wireless communication

ABSTRACT

In this contribution a novel model-based solution approach is introduced for medical networks and biotelemetric applications. Medical networks are communication networks that serve for the purpose of monitoring and protecting human health. These networks are designed to use biotelemetric ways to transmit the vital data to health observers such as doctors, nurses, first-aid teams, hospitals, and health agencies. These networks are also used in collective damages that may occur in situations such as flood, earthquake, war and terror and for treatments and follow-up of patients and to organize health teams more effective and efficiently. Implementations using this model presented here provides a reference design. In addition MCP (Medical Communication Protocol) and MMP (Medical Management Protocol) are designed to reveal how communications between modules designed. In this way, communication rules explained clearly on developed solution based on the model.

1. Introduction

In parallel to the rapid developments of wireless communication and networks, telemetric systems that use such technologies also show improvements. The aim of the telemetric systems in the field of health is doctors to be more effective in the treatment processes. In terms of patients, it aims to raise their quality of life and reduce dependencies to hospitals. Thus, because of treatment and follow-up were made in their normal lives, the satisfaction of patients is increasing. On the other hand, in terms of hospitals, it is possible to prevent lack of bed and excessive number of patients with the biotelemetric applications. Treatments and follow-up can be made without going to the hospital, in this way concentration in hospitals would be prevented. Furthermore infectious diseases in these environments can be prevented by providing an effective solution for combating diseases and transmitted infection in hospital environment.

Not only in normal times, but also biotelemetric applications are critical in cases of large number of emergency and first aid situations that may be arised in extraordinary circumstances such as war, terrorist attack and natural disaster. This situation is also stand out when literature studies are examined [1–7]. With these systems, prioritize processes according to emergency level of injured patients also known as triage, can be done more efficiently. Vital data such as pulse points, ECG and SpO_2 are observed through the sensors attached to the patients. This data is also transmitted to both the health and coordination centers and it can be traced with the screen of first-aid personnel. Thus, interventions of first-aid teams would be more appropriate and correctly. If number of observed people is assumed to be many, customized sensor networks emerge. These types of networks, are named as medical network [8–10].

Along with the hosting to the telemetric and sensor based systems, in essence, medical networks are communication networks that serve the purpose of monitoring and protecting human health. These networks are designed to use biotelemetric ways to transmit the vital data to health observers such as doctors, nurses, first-aid teams, hospital, health agencies. These networks are also used in collective damages that may occur in situations such as flood, earthquake, war and terror and for treatments and follow-up of patients and to organize health teams more effective and efficiently. Again, medical networks are used in intensive emergency departments of hospitals and triage applications.

In this study, a new model-based solution approach is demonstrated, in reference to the model [10], proposed for medical networks and biotelemetric applications. First, the basic unit and relations of the model are privatized and components of layered modules and internal structures of these components are designed and described their functioning in detail. Then, relations, described by this model and application communication protocols (MCP, MMP) which provide that relations are designed and described in detail.

* Corresponding author.
E-mail addresses: mcibuk@gmail.com (M. Cibuk), hasanbalik@gmail.com (H.H. Balik).

0965-9978/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.advengsoft.2011.03.019

Table 2.1
The model concepts for the application of medical networks.

Terms in model	Related solution	Real world
Montiored	Patient	Patient, injured
Observer	Client	Doctor, nurse, first aider, Visitor, patient's relative, 3. Person
Management center	Central manager	System admin, hospital management, ministry of health, authorised persons

2. Model-based solution approach

TCP/IP networks were selected as platform for solution approach. The most important factor in choosing TCP/IP, is internet. Internet is one of the most common environments for sharing information and communication. Due to interoperability with the other information and communication platforms, Internet was considered the most appropriate choice for solution approach in this study. Layered structure of the solution and equivalents in TCP/IP model are shown in Fig. 2.1.

As can be seen in the Fig. 2.1, there are layers of both models to meet each other. Network Access, Internet and Transport layers of TCP/IP model correspond to access layer of referenced model [10]. Application layer is divided into three separate layers which are Interaction, Management and Communication layers. Hence any communication network, that is based on TCP/IP and can carry out the functions of first three layers, can easily be used for the network access. Access layer of the provided model has no defined any binding layered unit so it offers users a flexible choice. Other layered structures are solved with the developed application-level modules.

2.1. Basic concepts and principles

When the basic structure of the model [10] examined, it can be seen that, there are three different communication points communicating each other over a central access environment and transmission among themself appear to be capable of two-way communication. The concept of the communication point which is defined in the model [10], corresponds to the patient, injured people, medical staffs, visitors, patient's relatives, administrative people and institutions. These concepts and their equivalents in this model are shown in Table 2.1.

Medical networks are structures that serve in the field of health. Therefore, there must be a service provider and service receiver on this structure. Functioning ground of this service is the infrastructure of the network. For this reason, service providers should have privileges to change and intervention within the mechanism. Management of the network and rules of the interventions are determined by the service providers. Service receivers are the parts of this mechanism and have no right to any intervention. Service receivers must follow the rules, determined by the service providers. Layered positions of concepts and access environment are shown in Fig. 2.2.

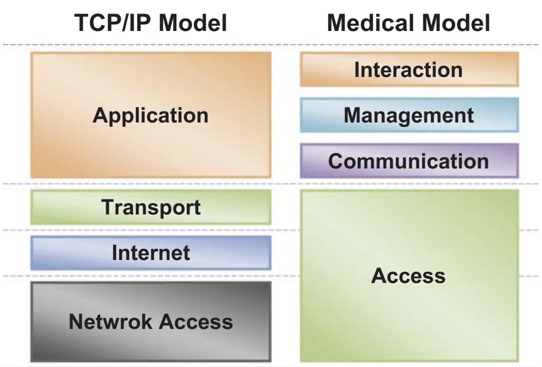

Fig. 2.1. Comparison of Medical Model and TCP/IP model.

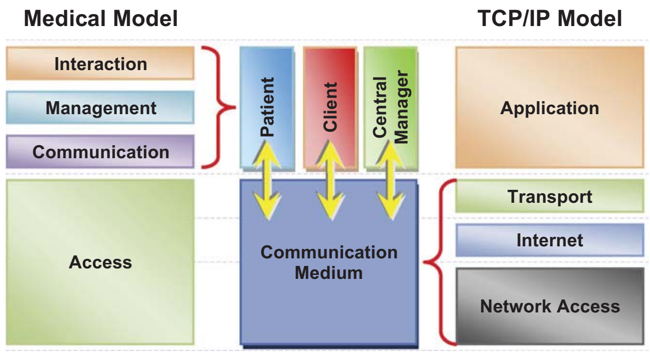

Fig. 2.2. Layered positions of Medical Model.

The basic principles of the model-based solution presented in this paper and modular structure of the solution were determined as follows;

- Node elements of communication (CN's); Patient, Client and the Central Administration. Where CN is communication unit.
- Each one node, must have at least one layered unit for Interaction, Management and Communication layer (CN = DU + MU + CU). Where DU, MU and CU are data unit, management unit and communicate unit respectively.
- Each CN is represented by a unique identifier within the medical network environment.
- At least one patient must be defined in the network.
- A network can have only one central administration.
- Within a network, it can be only one central administration. If there is a central administration in the network, it is responsible for the operations (such as rule definition, information storage, permission and authorization processes) of the network.
- There must be at least one client or central administration on a network.
- If a network has no central administration, communications between the patient and clients on this network occurs with the mutual authentication.

Patient is the side that is observed. Information collected with the sensors or other means from the ends is transmitted to the observer or central administration. The patient is considered to be independent of location and space. Communication nodes, used in the patient side, is named Patient Communication Node – PCN. These are hardware devices running on a PC, PDA and smart phone or specially design hardware with the appropriate service and protocols. A PCN must measure at least one or more vital information. Measurements can be made by PCN or independent sensors communicating with PCN. A PCN based sensor network can be created on a patient by using more than one sensor. PCN plays a role as a service receiver in medical networks. Therefore, PCN follows the rules and mechanism, defined on network system.

The client is the side that follows measured location and quantities information of patient. They see information from the patient, review and intervene if necessary. Furthermore, they can communicate with patient, make voice or video talk. With the confirmation of the patient or central administration, they can observe patient information provided by PCN. They can make real-time monitoring and may request query and statistical analysis under the privileges provided by central administration. Communication nodes, used in the client side, named Client Communication Node – CCN. These are the devices, such as PC, PDA, portable computers or smart phone and specially designed hardware on which appropriate service and protocols are running. CCN can be a service provider or receiver in medical network. Authority and responsibilities of a CCN user determine whether this user is a service receiver or provider. In general, health workers are service providers, visitors and relatives of the patients are service receivers.

Central Manager, providing coordination between the patients and clients, determines the operation and rules of the network. It manages the medical network and services offered on the network. It organizes the relations and communications between the patient and client. On the other hand, it hosts retrospective vital status information on its own. Thus, it provides opportunities to query and analysis information of all patients in the network. Communication nodes, used by central administration, named Administrative Communication Node – ACN. These are the server-based devices which have high data storage, processing power and speed of communication. It has appropriate running services and protocols in itself. It organizes the operation and rules of the network and provides the services and reviews for patient and clients.

Communication Medium is the connection element, creates the infrastructure of the medical network. It provides the communication between PCN, CCN and ACN. This environment may include physical or logical structures. It performs necessary operations such as transport, routing and connectivity to provide end-to-end transmission of packages. TCP/IP networks are selected as access environment in the solution approach. Layered position within TCP/IP networks and model of this access environment is shown in Fig. 2.2. As can be seen in the Fig. 2.2, communication medium is defined in network access layer in the Medical Model and it is defined in network access, internet and transport layers in the TCP/IP Model. Hence it can easily be said that the access environment includes physical and logical technologies to perform environment, access, routing and logical couplings.

3. Layered units

There are basic structures in the model definitions for each layer that undertake specific functions. These structures come together to form communication units (CN s) [10]. Each basic structure is expressed as a module in the solution approach. Basic structure of presented model and modular equivalents are shown in Fig. 3.1.

Communication units have orthogonal structure that are carried out as combination of CN = DU + MU + CU. The modules can be software or hardware depending on the purpose. The modules and their functionings are explained in solution approach in below.

3.1. Sensor control module (SCM)

This module samples measured value from sensor. Sensor's descriptive informations are also kept on this module. This information can be measuring type of sensor, measuring range, threshold values and identifier informations. A SCM can not perform a direct transfer of information. Sampling is done by according to commands comes from PMM, then results are sent to PMM. SCM software is a module that works on the physical sensors. Vango [30] and SenQ [31] approaches are used for structure and function. Accordingly, SCM basically consists of three components. These are sensor driver, sampling and pre-processing. SCM structure and function are shown in Fig. 3.2.

Sensor drivers provide interaction with sensor-detect hardware. This makes it more efficient to read the information by creating a standard interface. The sensor driver at the same time gives information about the status of the sensor to sampler. Sampler understands, whether the sensor is ready or not, through state information (is Present). The sensor drivers, as well as in operation [31,32] is divided into three different types. These are Event Sensors, Split Phase Sensors and Instant Sensors. Informer provides variables such as, sensor identifier, type, operation type and status.

Sampling component, depending on requests from query processor collects data from sensors. Samples can be different for each query. This provides to response the same sensor on different demands. Querier may request to create a new sampling process (create), to destroy an existing sampling (destroy), to initiate (start), stop (stop) or parameter adjustment (config) from sampler, Sampling component has submodules. These are Sampler Sensor Module and Data Collector module.

The Sampler Sensor Module, sampler performs the reading of information in accordance with the requests over sensor driver. Sampling process requests and organizes the timings for different reading. Data Collector, especially engaged in instant communications and in necessary conditions, sampled data is transmitted directly without any processing. This method is used for especially audio, video or real-time monitoring. Obtained raw data is also transmitted to central management module, through this module.

Data Preprocessor is the third component of the SCM. This component evaluates the sampler derived data according to specific procedures and criteria. Sensor calculates data coming from sampler according to querying requests and this process is continued during sampling period. Then sensor transfers processed sampling data to PMM module.

Between SCM modules and management modules in the sublayer of SCM exchange information according to R_6 relation rule and its procedure. In the upper layer, sensor or equivalent systems direct interaction through the drivers (R_8).

3.2. Location finding module (LFM)

This module is used to detect their location and place. Internal structure of location finding module is actually same as the internal structure of SCM. The difference between them caused from the sensor drivers. This module drivers are location finding sensors drivers or similar device drivers. There are different applications to distributed sensor systems to detect position using the IPS (Integrated Positioning System) based [2,3,22,32] or orbits around the

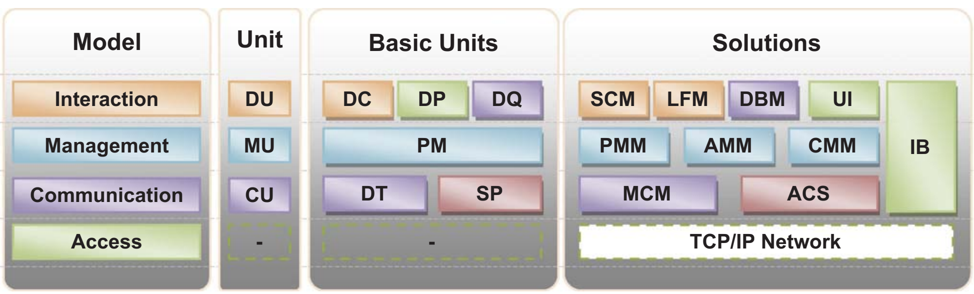

Fig. 3.1. Equivalents of the basic units of the model at the solution.

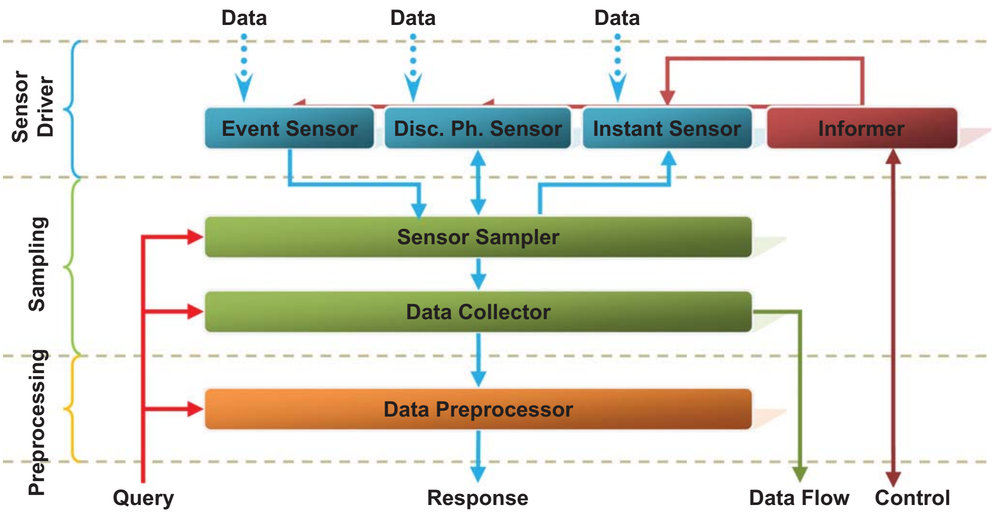

Fig. 3.2. Structure of SCM.

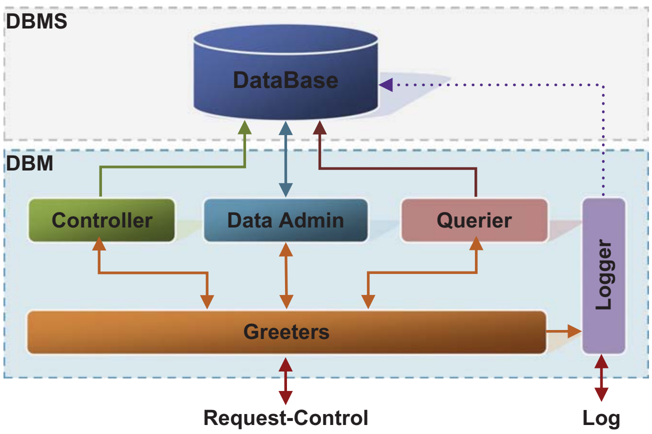

Fig. 3.3. Structure of DBM.

world placed the GPS (Global Positioning System) satellite system-based [18,23,33]. In this solution approach, the GPS technology [34] for location finding was used. Between LFM modules and management modules in the sublayer of LFM exchange information according to R_6 relation rule and its procedure. In the upper layer, sensor or equivalent systems direct interaction through the drivers (R_8).

3.3. Database module (DBM)

This module stores long-term information on the network and allows querying. So that, critical data obtained over the patient are stored in long-term period. Personal and user information, management and authorization relations are kept in users on a medical network. Internal structure and components of DBM are shown in Fig. 3.3.

Greeters meet the incoming requests coming from AMM (Administration Management Module). These consist user access, authorization, auditing, data recording and queries. Controller supervisions devices and users to the system registry (*checkRegister*), login (*checkLogin*) and authorization (*checkAuthorize*) performs the operations. DBM executes some query on related tables in the database for greeters requested audits. The controller can only read the database. Greeters takes confirmation from controller and than the greeters sends request to data admin and data admin performs that request. Data admin has the right to delete, and modify all kinds of writings on the database. Querier is a module that gives historical analysis, comparative statistics. Querier can read only database. Logger provides and keeps records about request and process of greetings. Especially it uses monitoring the system to solve the problems that may occur.

DBMS contains tables such as the system user information, personal information, authority, group, role, sensor data, PCN, CCN, log and other tables. Between greeters module, logger module and sublayers of both module exchange information according to R_6 relation rule and its procedure. DBMS is not a part of database module, it is a independent structure .In the upper layer, controller, data manager and logger modules direct interact with DBMS (R_8).

3.4. User Interface (UI)

The user interface provides the interaction of people with visual and auditory end devices. Interface consists of two component which are visual form (frontends) and controls (user interface controller). These components are shown in Fig. 3.4.

Frontend shows the data from the user to provide audio-visual items. In this way, the user would have got a desire data to see the patient side. The user passes through interface controller in order to do transactions by using frontend buttons, edit controls, etc.

There are two basic task of the interface controller. Firstly UI controller properly transfers requests that come from frontend through controller to management module, then returns responses to related control and forms. Secondly, when the user login to system, UI controller checks authorization, then UI controller shows form and controls through frontends that according to based on user's roles.

In the UI module, between UI controller module and submodules in the sublayer of UI controller exchange information according to R_6 relation rule and its procedure. In the upper layer, frontends direct interact with user (R_8).

3.5. Patient management module (PMM)

This module is responsible for all managerial procedures and decision mechanisms in the PCN. PMM like a brain of PCN. PMM collects and reviews sensor informations coming from the top of modules such as SCM and LFM, then PPM transfers to related unit through MCM. From this perspective, it is like a sink nodes [35,36]

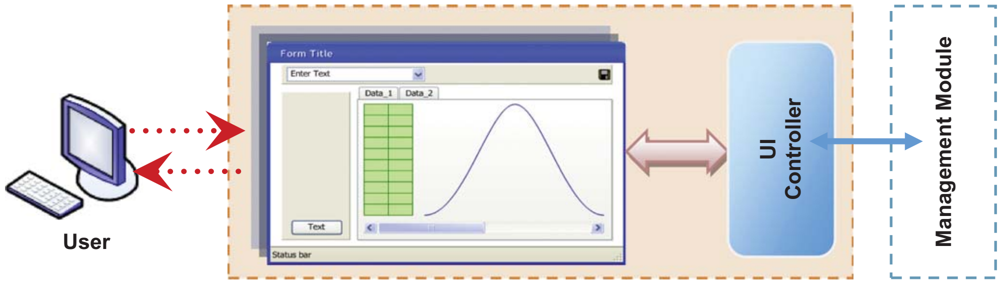

Fig. 3.4. Structure of user interface.

in sensor networks. This module manages the top-layer modules to information and control flow with central management, response to requests, direct communication with clients.

PMM has different operation modes. These provides to take on different missions of PMM in different system solutions. Therefore, PMM works more compatible for different configurations. PMM has three different operation modes. These are: *Sense*, *Server* and *Hybrid* mode. Who used PCN or central management unit connected to PMM can be changed current mode of PMM. Code bits of the package are checked to understand, in which mode PPM runs. Code bits are determined within the scope of R_3 relation. Components and relations of PMM are shown in Fig. 3.5.

SCM transports command and parameters of the sampling requests to the components and reporters on the SCM according to the queries and operations transmitted by process controller. SCM wants to the creation of a new sampling, start, stop and terminate existing sampling.

SCM transmits data to reporter module then reporter module transfers incoming data to the message sender during reporting period. The most important tasks of the reporter module is to provide regularly transferring data and preventing redundant data traffic over the network. The pre-processor of SCM works with sampling period and reporter works with reporting period. This is the most important difference between pre-processor of SCM and reporter. These are different from each other.

Process Controller evaluates incoming wishes and requests through messages. Process controller performs operations such as information requests, connection, creating session, monitoring session, control of sub and upper layer module, interaction to user interface. In addition, the Process Controller keeps and updates session, connection, user and sensor information tables.

Message Receiver performs the necessary control and analysis of messages of lower layer by MCM or the ACS modules. Message Receiver, assigns object or variable from incoming message according to type of messages. Then, Message Receiver transmits that to process controller.

Message Sender encodes data which comes from process controller, reporter, or data collector in the upper layer then it creates a package to send transmitting module in lower layer. Message Sender also sets to priority level for instant communications. Process Controller is informed transmitting status by message sender.

Message Receiver, Message Sender, ve Process Controller have R_4 relation to lower layer. Querier, reporter and process controller have R_6 relation to upper layer.

3.6. Administrative management module (AMM)

This module is responsible for management procedure and decision mechanisms in ACN and medical network . AMM performs management of upper layer module in fulfill of request, controls data flow with patient and clients, directs connection between clients and patients. AMM manages DBM to store historical patient and sensor data for long-term. There are three main components of the central management module. These are shown in Fig. 3.6.

Process controller analysis request and wishes that comes through message. AMM performs connection requests, login requests, and registering requests coming from patients and clients AMM controls lower and upper layer modules, and it performs such as managing interaction of user interface. In addition, it tries to response historical and statistical query from clients by using DBM. It transmits event log, performed process to DBM module which stores login file.

Message receiver performs the necessary control and analysis of messages of lower layer by MCM or the ACS modules. Message receiver, assigns object or variable from incoming message according

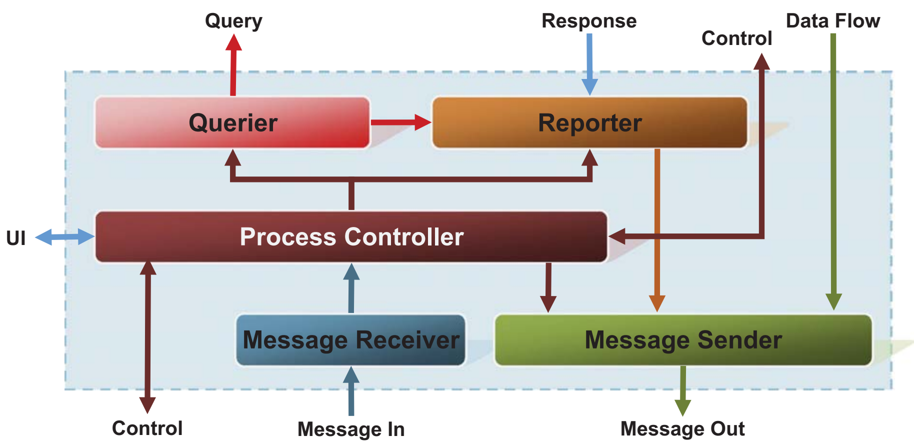

Fig. 3.5. Structure of PMM.

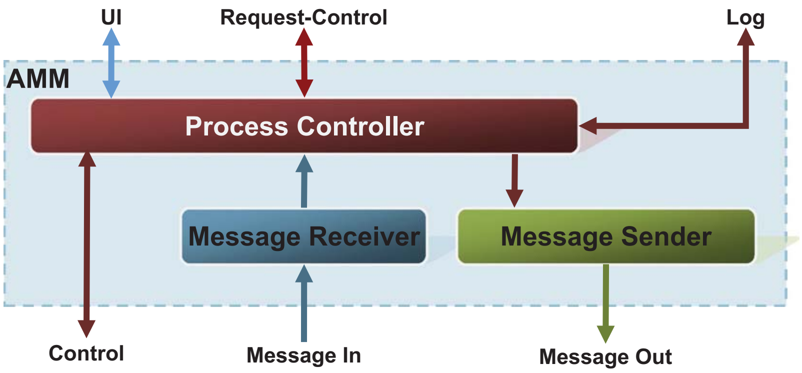

Fig. 3.6. Structure of AMM.

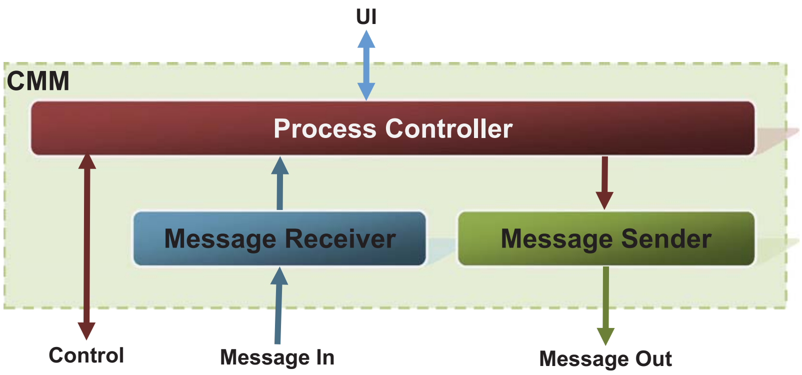

Fig. 3.7. Structure of CMM.

to type of messages. Then, message receiver, transmits that to process controller.

Message sender encodes data which comes from process controller then it creates a package to send transmitting module in lower layer. Process controller is informed transmitting status by message sender.

Process controller has R_6 relation to upper layer. Message receiver, message sender, and process controller, have R_4 relation to lower layer.

3.7. Client management module (CMM)

This module is responsible for all managerial processes and decision mechanisms in CCN. Requests from the UI module are transferred to the related points through the MCM. Information and control flow with central management, response to requests, direct communication with PCNs are under this module responsibility. CMM module structure shown in Fig.3.7.

Process controller analysis requests and wishes that comes through message. CMM transmits connection with patients, login requests, and registering requests to related point. CMM controls lower layer modules and it performs such as managing interaction of user interface. In addition, it transmits request of historical and statistical query from clients to central management for monitoring and evaluating.

Message receiver component performs the necessary control and analysis of messages of lower layer by MCM and the ACS modules. Then, Message receiver, transmits that to process controller. Message sender encodes data which comes from process controller then it creates a package to send transmitting module in lower layer. Process controller is informed transmission status by message sender.

Process controller has R_6 relation to upper layer. Message receiver, message sender, and process controller, have R_4 relation to lower layer.

3.8. Medical communication module (MCM)

This module performs communication between medical networks such as PCN, ACN or CCN and transmits data flow to media access. Also it informs status of media access to upper layer. It checks incoming medical package for conformity of source and target. If the package passes control, MCM analyze the package and then it transmits to control unit in the upper layer, if the package does not pass conformity control, the package will be droped. There are four basic component of medical communication module. There are shown in Fig. 3.8.

Communication controller, is responsible for logical operation and control of the MCM. It checks incoming message from message receiver for conformity of source and target. If package is not suitable or has a different target, it is dropped by communication controller. CC prepares package from upper layers with suitable parameters then if access medium is available, Communication controller transmits package to package sender.

Medium controller informs to communication controller about operation of platform which works in access medium. So that physically and logically connections are ready or not, status of network connections are learned. Package receiver listens the medium and it takes incoming package form other node then it transmits this packages to communication controller. Package sender transmits MP package to medium according to given data and parameters from communication controller.

Communication controller has R_4 relation to upper layer. Message receiver, message sender, and medium controller components

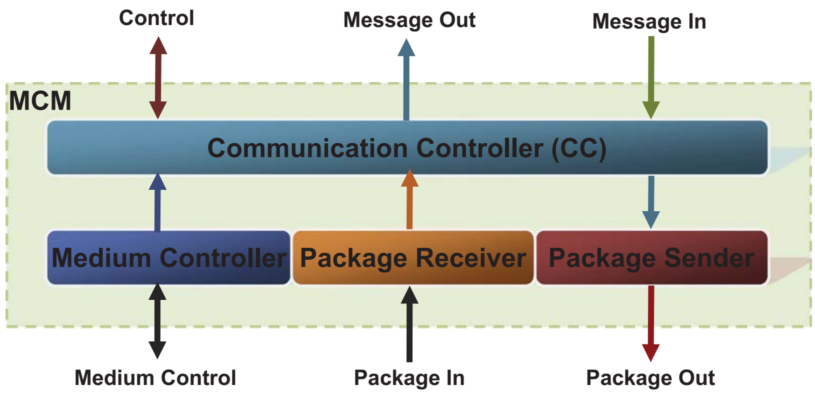

Fig. 3.8. Basic components of medical communication module.

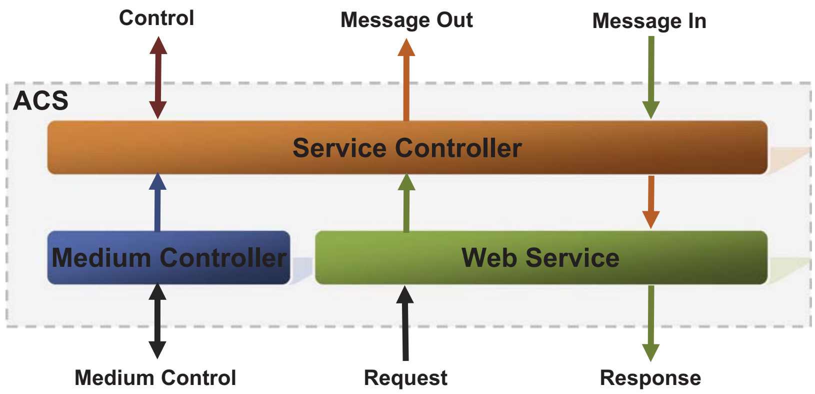

Fig. 3.9. Structure of ACS.

have R_2 relation to access medium and that bound to rules of access medium.

3.9. Application communication service (ACS)

This module purposes to serve service to third party application in access medium. Therefore, applications from outside are able to join the medical network system. Application in the communication service of this solution approach is intended to http based service such as a web service. Therefore many of internet bowsers and other applications connect to medical network easily for monitoring and querying. Basic structure of application communication module is shown in Fig. 3.9.

Service controller converts to message such as incoming commands and requests from web service component and then it transmits to upper layer or if messages comes form upper layer, it transmits to web service. It is kind of a converter. Therefore this component transmits status and control data comes from access medium controller to upper layer. Medium controller gives information about operating and status of platform to service controller which runs in access medium. Connection of medium, that is ready or not, and status of access medium are learned by this way.

Web service component transmits requests to service controller which comes from web-based client. This component can be thought as IIS or Apache web server for better understanding. This applications are used for web service component. Other application can also be used to do same task.

Service controller has R_4 relation to upper layers. Web service and medium Controller components, have R_2 relation to access medium and that bound to rules of access medium.

3.10. Internet browser (IB)

There is not any match in the network for this module. It can be considered as third party applications. Http platform, that uses web browser, has been preferred in this solution approach. Any web browser transmits user requests through web to application communication service.

4. Horizontal relations

Relations, that is intended speaking and distinguishing between equivalent nodes (CN's) in the same layer, is defined as horizontal relations [10]. Horizontal relations are rule makers and they express themselves by rules. In other words they explain how to do a thing rather than making it. This solution approach is consist of different relations which are called R_1, R_3, R_5 and R_7.

4.1. R_1 horizontal relation

This relation is providing data transfer between end nodes on media layer. Information packages are presented to the media are directing, delivering to target, and flow control occurs around this relation's rules. This relation is determined by selected media rather than determined by the model. In this study it has been acted by rules according to TCP/IP's rule set rather than making a new definition.

Communication media is corresponded to network, internet and transport layer of TCP/IP is seen on Fig. 2.2. Then physical standards, media access techniques and protocols which are determining relation and used by these layers of TCP/IP create content of R_1

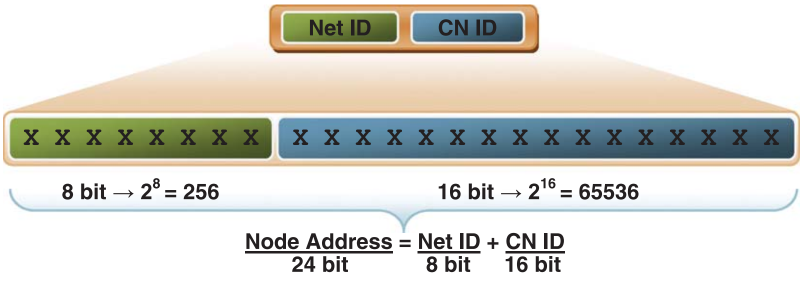

Fig. 4.1. CN addressing in MCP.

relation. Because of contents of these protocols are standardized and certain, there is no need to extra explanation about them.

4.2. R_3 horizontal relation (MCP)

This relation sets out rules and the exchange of data between CUs. These rules are carried out by, addressing which is providing CUs to communicate each other and laying down the flow of message and by PDU structure. R_3 relation developed in this study is called Medical Communication Protocol – MCP. MCP is so critical on communication and package transferring processes. This protocol especially determines end point addressing, how hierarchical organization should be, medical package transferring between ends and rules and methods which are determining the course of communication.

In this context, MCP protocol defines three main functions. These are, Identification (addressing), PDU format and communication course.

4.2.1. Identification (addressing)

According to MCP protocol, each CN should have a node address within a medical network. Nodes address structure illustrated in Fig. 4.1.

Each node in medical network consists of 24 bits addressing. Eight bits are Net ID and 16 bits are CN ID of 24 bits. Net ID is an identifier that is used for medical networks. Medical network having CN is understood by this identifier. Another feature of Net ID is allowing to find different medical networks in the same physical network media. CN ID is descriptive that separating nodes inside same medical network. Node addressing has a hierarchical structure. Addressing is according to this hierarchical structure. According to hierarchy, first and last addresses of Net ID and CN ID are special addresses. These addresses carry a special meaning. First addresses are representative and last addresses refer to everyone under hierarchical sub groups. MCP addressing hierarchy and meanings of them have shown in Table 4.1.

MCP structure developed in the present study has flexible and improvable features. CNs come together and creates medical networks. It has to be a representative on each medical network. Representatives of different medical networks come together and create councils. Each council must have a root representative. Different councils come together and create federations; different federations come together and create confederations. Structure continues open expanding. A CN have addresses more than one provided that not show the same medical network. It is able to directly communicate within contained network members and can communicate through representative of other network members.

Representatives are usually nodes which have central management position within network. Each network representative amounts addresses of their and other representative's council in the Node Address Table (NAT). Representative registers itself to root representative when it is included any council. Root representative updates its own NAT table after accepted that representative and send this updated table to other council members. So address forwarding between medical networks would be operated independently from communication media.

Table 4.1
MCP addressing hierarchy.

NetID	CNID	Definition
0	0	Root representative
0	1	All of representative (broadcast to representatives)
0	X	Any representative
1	0	All of network representative
1	1	Everyone in all networks (broadcast to all)
1	X	Any X node in all networks
X	0	Delegate of any networks
X	1	Everyone in any network (broadcast to nodes)
X	X	Any node in any network

4.2.2. PDU format

MCP, uses unique protocol data unit to fulfill its set of rules and functions. Structure of that PDU has shown in Fig. 4.2.

Starting delimter used to determine whether there is MCP package or not. This delimiter has 8 bits and it is a special sequence of bits which is shaped as 10101111. Source and destination node addresses come after starting delimiter. 24-bits node addresses placed in these fields about source and destination node. So, source–destination relation is introduced. After source and destination node addresses, code bits field which includes 8-bits comes. Any nodes role (operating mode) into network is determined by setting the field to 1. Those bits are shown in Fig. 4.3.

There are 8-bits tip bits after the code bits which indicate type of medical packages. Because of these bits it is learned that how to take data and which layer will take them only one bit can be set according to rule. Type bits and meaning of them has shown in Fig. 4.4.

Type bits consist of two fours including communication and message. Communication bits specify that communication layer handles incoming package and message bits specify that management layer handles incoming package as well. Communication bits are *request*, *response*, *update* and *TickTock* respectively.

- *Request*: These packages wait responses return. There should be response for each request. When sender sends a request, it waits about *reg_time_out*. It repeats about *req_retry* if couldn't get any response. After all sends a warning message to whom requested these and it ends the process.
- *Response*: on target side it sends as an answer to request packages. These packages don't require any response return.
- *Update*: these packages don't require any response return and they send directly to target. They used for updating NAT tables and parameters which are necessary for communication.
- *TickTock*: It is used to learn working condition of any CN.

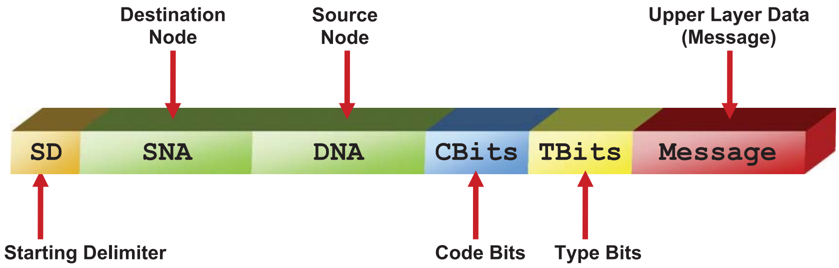

Fig. 4.2. PDU structure of MCP.

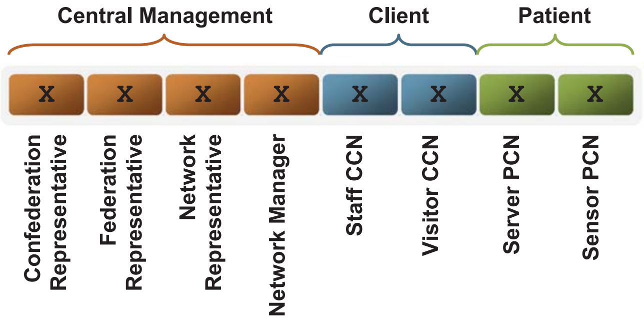

Fig. 4.3. Code bits for MCP packages.

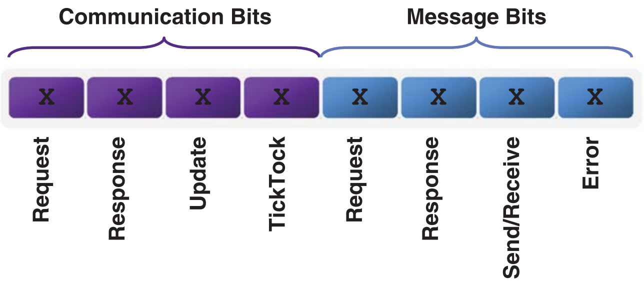

Fig. 4.4. Type bits for MCP packages.

Communication bits are *request*, *response*, *send/receive* and *error* respectively. Package that message bits are set transferred directly to management layer after resolved by communication layer.

- *Request*: These packages wait responses return. There should be response for each request. Include of the package and answer reply durations is determined by management layer.
- *Response*: on target side it sends as an answer to request packages. These packages don't require any response return. Include of the package and dealing is determined by management layer.
- *Send/receive*: these packages don't require any response return and they send directly to target. They usually used on instant data transfer.
- *Error*: They are used to inform other side about erroneous packages or processes of any CN. Inside those packages there is information about cause of that error.

After type bits message field is placed on MCP PDU. This layer represents the upper layer data. Length of this is not constant and varies of contents of message. If tip bits of incoming message indicates that message belongs to upper layer then directly transferred to upper layer member after message taking inside it. MCP doesn't concern about message structure and content. Their contents and structure is determined and processed by R_5 horizontal relation.

4.3. R_5 horizontal relation

This relation sets out rules about information exchange and how it take place between MU's. These rules take places through a protocol inside R_5 relation. This protocol will be named as Medical Management Protocol – MMP.

4.3.1. Medical Management Protocol (MMP)

MCP medical networks are responsible for addressing between ends and package switching. On the other side MMP is responsible of operations and processes. MMP determines rules and methods about logical organizations and talking processes between ends. MMP protocol uses a PDU format to fulfill its responsibilities. This PDU's is named as "Message" and a sample message structure shown in Fig. 4.5.

PDU of MMP has five parts and first of them is activity field. This field shows transaction process of message. Transaction processes are four types and these are handled by managers of different process. MMP processes handled is detailed on Section 4.3.2. Activity field consists of 8-bits. Order and meaning of these bits is shown in Fig. 4.6.

First four of activity bits show process. Just one bit can be set at the same time. Then two bits action comes. Action determines behavior of incident and its direction of interaction in process. There are two types of actions on MMP. These are *"Set"* and *"Get"* actions. Last two bits of action bits show result of action. There are four different results. *"Not"* is meaning of there is no result, *"Error"* is meaning of result has come back with error, *"Okay"* is meaning of result of process is positive and *"Prerequisite"* is meaning of some pre-processes should be done.

There is Command-Data field after activity bits. This field has variable lengths and contains different parameters and rules for each process. According to process of message, this field is handled by different process managers. Each process has significant rule sets that perform some operations inside. These rules can be considered as methods and functions which are fulfilling some parts of tasks determined by the process. There are variables and data types accepted by each rule.

After Command-Data field message, queue number comes which has 32-bits. A number is assigned to each message before it is sent and this number is put into message number field of PDU. Message numbers is used for MU that distinguish sent messages from each other during active process. Message numbers take values ⁎⁎from 1 to 2^{32}-1. The process is finished on the message transfer process when message numbers comes to the end.

After message, number field related message number which has 32-bits comes. This field shows response to which message corresponds to incoming message packages before it. Every bit set to zero on this field if there is no related message which is corresponded any message before. So it is reported to other side that the message is first. Process of messaging between two messages and changed reference numbers and messages is shown in Fig. 4.7.

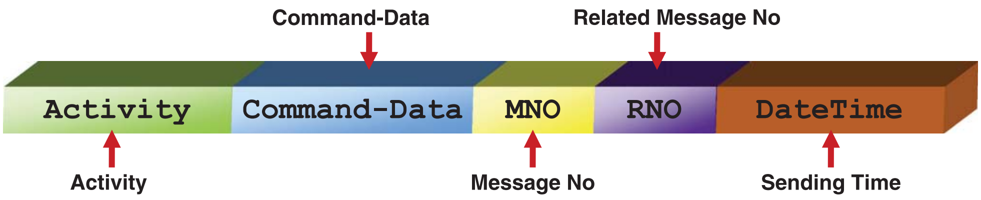

Fig. 4.5. PDU structure for MMP.

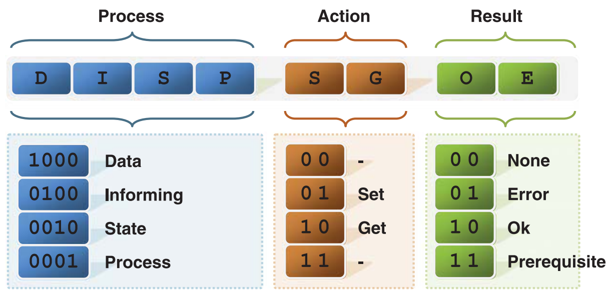

Fig. 4.6. Activity bits for MMP Messages.

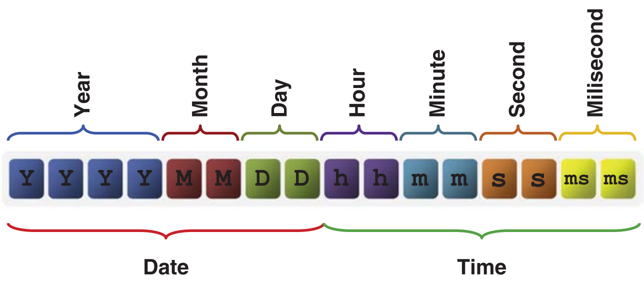

Fig. 4.8. Datetime format for MMP Messages.

When source CN wants to send a message to destination, it changes reference number to zero and sends a message with X number which is its own specification. If RNO is zero then it shows new process is started. All bits in MNO fields must be set to one to finish process. So finish of the process is reported to the other side. During the process any node could last the process at any time.

Processes are for one message package. There is no need to process during the information transfer of one message. MMP solves this itself. On these type of cases every bits of MNO field is set to 1 and every bits of RNO field is set to 0. Process is started with RNO value and finished with MNO value on such a message. So, process is completed its life into one message package. This type of processes called Single Shot Process (SSP).

Date and time field is the last field of MMP PDU. It has 16 characters and indicates messages sent date and time. Receiving side looks this field and understand when the package is sent. Date and time format and its order is shown in Fig. 4.8.

4.3.2. MMP operational process

MMP determines rules of processes and operational processes. There are two basic actions on operational processes. These are " *set*" and "*get*" actions. Actions determine the direction of interaction and behavior of events on processes. "*Set*" action is target side interaction because it is caused change or update. "*Get*" action is source side interaction because target moves according to the result of "*get*" action. Actions are not significant alone. They are significant with processes and rules in MMP PDU. MMP which is developed in this study defines four main operational processes into it. These processes are;

- *Data*: sets out procedures of questioning, reporting and saving information which is read from patient.
- *Information*: provides operations of reading and saving descriptive information about node, sensor and people within medical network.
- *State*: sets out patients and CNs activity and vital status changing, reading, and declaring.
- *Operation*: necessary actions set and fulfill for keeping nodes and users functioning within medical network.

MMP processes transactions required to fulfill the functions of each process, defines a team called in the script. All operational processes are performed by these commands. Each command processes given information or parameters in accordance with the action specified in the message or itself that provides the requested information or parameters. MMP protocol is defined processes, commands defined during processes and their description is displayed in Table 4.2.

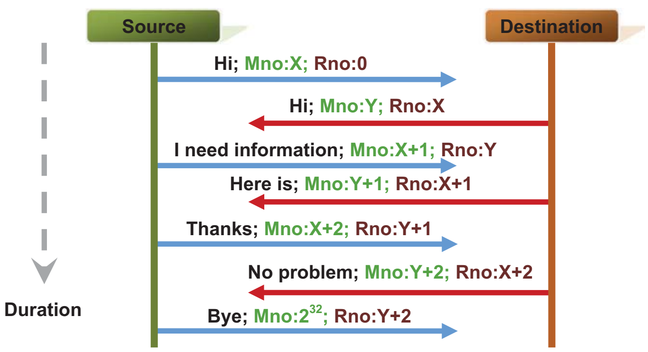

Fig. 4.7. Flow process and numbering for messages between CN.

4.4. R_7 horizontal relation

This relation reveals communication and information flow between DUs working on interaction layer. It is considered at least one DU in each CN. However, units of DU are not in direct communication with each other, they are in interaction lower or higher than structural units. Possible operation and functions on R_7 relation appears processes between R_5 relations contained in the MMP. Therefore functioning of MMP processes already covers R_7 relation.

As a result there is no description or rule for R_7. It is adopted that this subject is solved in MMP. But it doesn't mean completely ignore the R_7. Because R_7 is not the part of solution approach, is part of a model [10] and it is defined in model. Therefore it is possible to be addressed in solution approaches based on model.

Table 4.2
MMP processes and process commands.

Process	Command (Function)	Comment
Data	data_Query()	From PCN or ACN vital or other information questioning by CN is used to collect statistical information
	data_Value()	Is used for reading and recording obtained from the sensor data of patient
	data_Report()	Is used to reporting and management of these processes of vital data on PCN
Information	info_Person()	Is used to read and record personal information of medical network users
	info_Sensor()	Sensor information and recording their capabilities and retrieval all of these within PCN
	info_Node()	Is used to learn the hardware knowledge and abilities of CNs
State	state_Active()	Is used to launch, stop, pause and restart the CN or any sub-units
	state_Emergency()	Is used to sent and processing of emergency signals over PCN
	state_Triage()	Used for health care prioritization of patient according to the pre-defined settings on PCN
Operation	proc_Register()	Is used for recording and removal of a CN to medical network
	proc_Authenticate()	Is used to system logging for registered CN on medical network
	proc_Session()	Is used to log on a session or managing that session of which CN is logged in another CN on that medical network
	proc_Synchronize()	Is used in time synchronization operations of a CN for healthy flow of information to be
	proc_Encryption()	Encryption methods for secure transfer of information between CN and to specify how to use them
	proc_Connection()	I used for making a connection and termination between CNs

5. Vertical relations

Service between relations, state and information transformations are possible on this kind of relations. Vertical relations shall be carried out with binding or services. Bindings are consists of physical structures like cable and connectors and services are software. Conduct of relations is provided within method, function and parameters within defined rules [10].

In this study basic methods to be followed on execution and implementation of vertical relations shown in Fig. 5.1. Arrows are bi-directional, if information flows between upper and lower units.

Exchange of information between neighboring layers is performed with state-process-queue board trio. State board is just readable for the other side and other layer can not intervene posts on this layer. Process board is used for intentions and actions between neighboring layer units. This unit understands what its neighbor wants to do and after fulfilling the request, inform it from this board. This board is readable and writeable from neighbor units. Queue is a carrier box which is included information about intentions and actions. This box is filling with information by sender and receiver sends them out after processing.

Perform services on software side is provided by appropriate function, method and variables. For the action and intention described in vertical relation, appropriate functions and methods are written to perform these operations. Numerical and logical variable expressions are also used for states. Functions are considered as code snippets that passing parameters and processes in queue. Bidder should also be aware of results of the processes done by each function. Each function can fulfill one process and there are functions fulfilling many processes as well.

Another aim of vertical relations are encapsulation and decapsulation processes [37]. With this approach which is developed based on model, there are four different relations put forward R_2, R_4, R_6, R_8.

5.1. R_2 vertical relation

This relation is defined services and operation between media and communication layer. R_2 relation performs the function of encapsulation R_3 protocol data inside R_1 protocol. This process is shown in Fig. 5.2.

Duties and responsibilities of R_2 are performed by 5 basic functions. These are `create_Package()`, `send_Package()`, `read_`

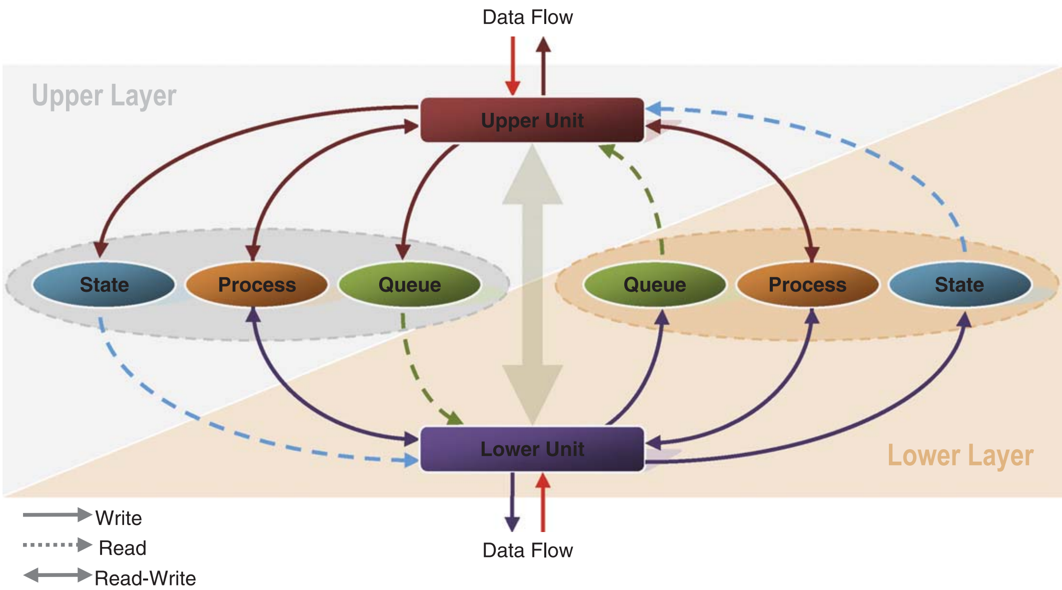

Fig. 5.1. Cross-layer vertical relations and communication form for a solution.

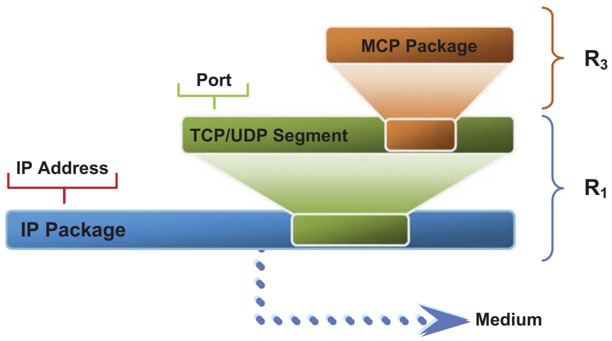

Fig. 5.2. R_2 encapsulation process.

`Package()`, `process_Package()` and `check_CommStatus()` functions.

```
function create_Package(DNA,TBits,CBits,
Message,Return){};
function
send_Package(DIP,DPort,MCPackage,Return){};
function
read_Package(rMPackage,MCPackage,Return){};
function process_Package(rMCPackage,Return){};
function check_CommStatus(Return){};
```

`create_Package()` function creates and prepares MCP packages and send back about state information. `send_Package()` function makes access environment encapsulation to MCP packages and pass it to the media. Transferring and preparing package information is sent back. `read_Package()` function is reading packages coming from access medium and makes accuracy control. Reading of package and accuracy control information is sent back. If the package is right and suitable MCP package then transfer it to the right variable otherwise sent error package to the back. `process_Package()` function is processed conformity of `read_Package()` and transferred data package to `rMCP_Package()` within R_3 relation rules. This function primarily examines of source and target node relation of package and after that looks tip bits of MCP Package and properly process if the processing is necessary on CU. If it is interested to upper-layer information then makes necessary board and queue processing for taking it by upper layer. Also processing of package and state information is sent back by this function. `check_CommStatus()` function includes lower functions that checking source IP, port, medical node, etc. parameters set or not. These parameters are necessary for functionality of first four functions. All functions performed within R_2 apply this function which is developed inside them. So they get information and shape their behaviors according to media and other necessary parameters.

R_2 relation independently process all described these functions inside it. To determine the R_2 relation exactly we need to define how these functions have interaction between themselves. So how the processes take place is grasped better.

In this study interaction between functions are also named communication services. Services bring together functions and duties defined by vertical relation in a logical flow. R_2 service structure and how functions interact between themselves by using flow diagram are shown in Fig. 5.3.

5.1.1. R_4 vertical relation

This relation provides coordination and communication between units which are working communication and management layers into developed model approach. These units shall exchange information with MMP. R_4 vertical relation determine how to communication layer units provide service which are called MU and inside of management layer. Message performs encapsulation and decapsulation processes of PDUs. This process is performed by developed `create_Message()` function on R_4 relation. Basic structure and parameters of this function is;

```
function create_Message(Activity,Command_Data,
RNo,Return){}
```

`Activity` parameters inside function specify message activity bits. `Command_Data` includes command of sending message. This command varies according to demand and process. `RNo` parameter sends back associated message number and `Return` parameter sends back state information of function result.

Generated message packages must be transferred communication low-layer which is serving for sending. This process is fulfilled by `send_Message()` function into R_4 relation. Basic structure and parameters of this function is given;

```
function send_Message(Message,Return){};
```

`Message` parameter inside function hosts send message. Message is put into queue board and `have_send` variable is set on process board. `Return` parameter send back state information about the result of function.

Some functions have taken place inside R_4 relation for taking and processing message. `read_Message()` function is used for taking message from serving communication low-layer if there is. Basic structure and parameters of this function are given;

```
function read_Message(rMessage,Return){};
```

This function continuously monitors `have_message` logical variable in process board of low-layer. If this variable has set to 1 it understands there is an incoming message and read queued message then send it to `rMessage` parameter. Then read message is checked from accuracy. If the message pass these controls it is sent to concerned function for processing. `Return` parameter send back state information about the result of the function.

Messages which are tested for compliance are transferred to `process_Message()` function for processing. This function opens and looks incoming message activity bits and sends it to appropriate process managers. Basic structure and parameters of this function is given;

```
function process_Message(rMessage,Return){};
```

rMessage parameter is included incoming MMP message. Return parameter send back state information about result of the function.

Finally there is `check_MngStatus()` function in R_4 relation. This function checks necessary parameters are set or not for functionality of first four functions. So, behavior is shaped accordingly management processes and other parameters by learning information about them. Basic structure and parameters of this function is given;

```
function check_MngStatus(Status,Return){};
```

`check_MngStatus` function has two parameters. `Status` specifies parameter required to be checked and `Return` is used to send back information. If `Return` variable is different from 0 there is an error. Service structure of R_4 and interaction between functions is shown in Fig. 5.4.

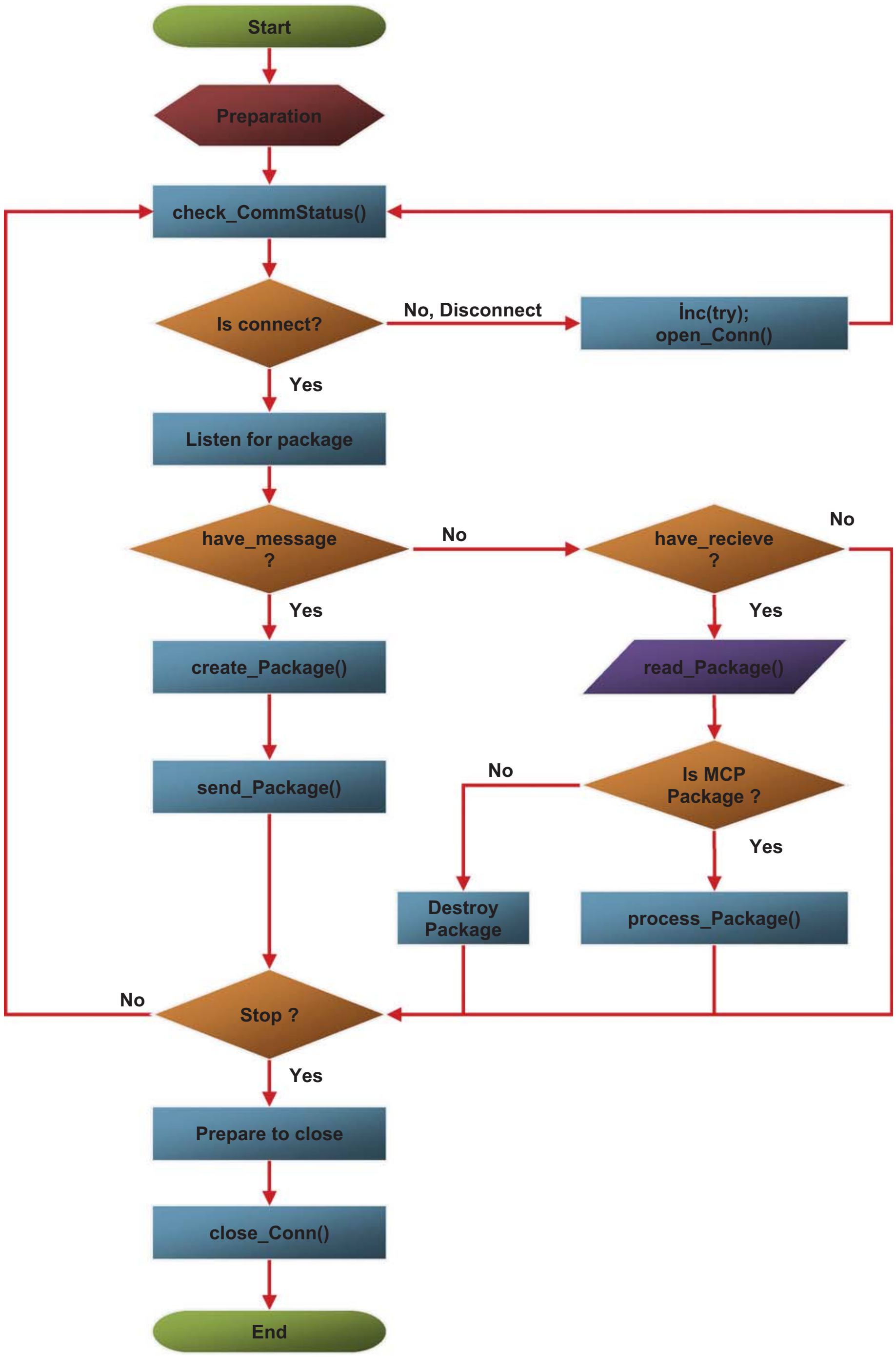

Fig. 5.3. Interaction of R_2 functions and service structure.

5.1.2. R_6 vertical relation

This relation sets out information transfer and communication between MU unit on management layer and DU unit on interaction layer. R_6 relation fulfills rules and formats set out by R_7 horizontal relation which is on the interaction layer. But, independent R_7 relation is defined and operational process rules are valid on this approach. R_7 relation does not define any rule but R_6 define rules by operational processes in MMP protocol. Each process on MMP has rules into it. R_6 relation offers appropriate method and functions to realize these commands. Each command process parameters according to action stated into message or bring information requested from it. Processes and commands are form of speech into DUs and between MUs.

set_Command and *get_Command* derivatives have developed for each command because commands for "*set*" and "*get*" action behave on different manner. Process commands performed within the scope of R_6 relation shown in Table 5.1.

There is a process controller for each process that understanding and taking into action these commands on concerning unit. Process managers are kind of interpreters. They interpret commands analyzed and sent by `process_Message()`. Then it provides processing them with concerning units. Overall function of process managers shown in Fig. 5.5.

Any process controller learns action is Do or Bring by looking activity bits of incoming MMP message. Then finds which process command matches in R_6 by solving command-data part of the

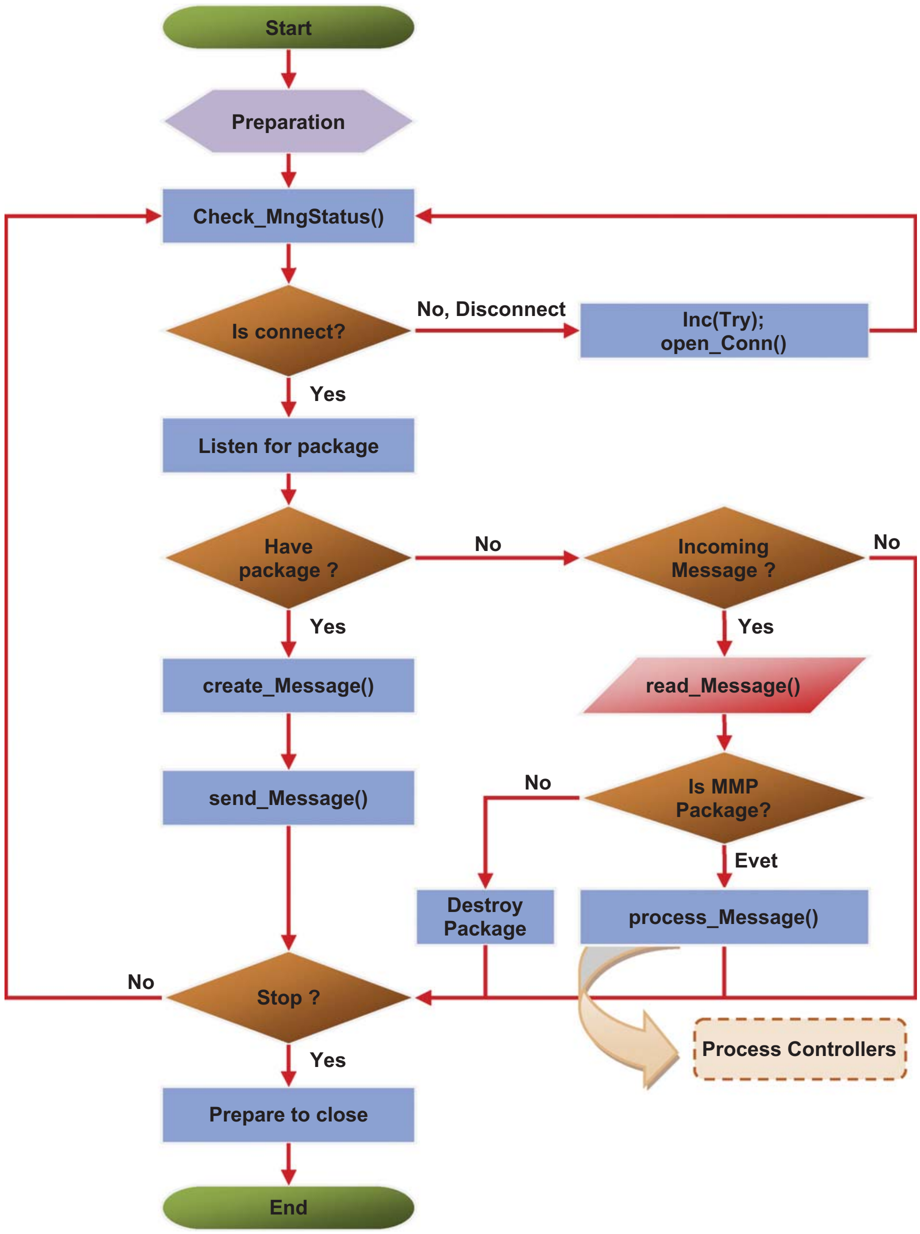

Fig. 5.4. Interaction of R_4 functions and service structure.

message. Then calls concerning command function and provides processing request by appropriate layer unit. Command-data fields into MMP message get string tip information.

5.1.3. R_8 vertical relations

R_8 relation is for other applications integration with solutions based on this model or directly interaction for people. Due to existence of this relation and function of it may vary to applications or people. Therefore there is no exact rule or functions defined about this relation. Because each integrated application gives a shape to this relation within their own rules and approaches.

6. Results and conclusion

Today, main outstanding issues for given better health service; with patients is observed and treated in their own vital circle by kept them away from hospitals as much as possible, intervene

Table 5.1
MMP processes and commands for R_6 relation.

Duration	set_Command ()	get_Command ()
Data	*set_data_Query()*	*get_data_Query()*
	set_data_Value()	*get_data_Value()*
	set_data_Report()	*get_data_Report()*
Information	*set_info_Person()*	*get_info_Person()*
	set_info_Sensor()	*get_info_Sensor()*
	set_info_Node()	*get_info_Node()*
State	*set_state_Active()*	*get_state_Active()*
	set_state_Emergency()	*get_state_Emergency()*
	set_state_Triage()	*get_state_Triage()*
Process	*set_proc_Register()*	*get_proc_Register()*
	set_proc_Authenticate()	*get_proc_Authenticate()*
	set_proc_Session()	*get_proc_Session()*
	set_proc_Synchronize()	*get_proc_Synchronize()*
	set_proc_Encryption()	*get_proc_Encryption()*
	set_proc_Connection()	*get_proc_Connection()*

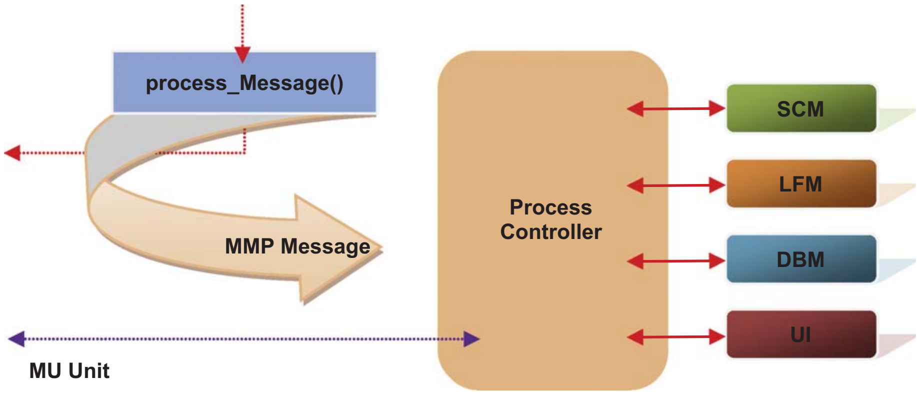

Fig. 5.5. Overall operational scheme fort he process controller.

cases occurring in the external environment with rapid and appropriate decisions. Many of the studies [1–29] done recently are progressing in that direction. On researches studies, development efforts for most effective and rapid systems are continuing about biotelemetry and medical network applications. Model [10] is presented in the previous study, contribute to these efforts. Developed model is effective and feasible approach for such networks. In addition it is thought that this solution is displacing many works done before and can be utilized for the future studies by this model.

In this study a solution approach has put forward which is applicable for biometrical systems based on application model [10] that basic concept and principles determined before. In this context, model-driven solution approach is explained and all basic unit and relations have been privatized by solution approach model on the concept of observer and central management described by this model. CU, MU and DU layer unit modules and their internal structures are designed and detailed. On the other hand, relations defined by this model and MCP and MMP protocols that providing these relations, designed. Their internal operations and PDU structures have been determined in detail.

It has been shed light on performing designed model [10] for biomedical networks by developed solution approach. Thus, implementations using this model provided a reference design. In addition MCP and MMP protocols are designed to reveal how communications between modules designed. In this way, communication rules explained clearly on developed solution based on the model.

In conclusion, with this study new and effective solution approach is exhibited on biomedical networks field which is developed based on model.

References

[1] Anliker U, Ward JA, Lukowicz P, Troster G, Dolveck F, Baer M. AMON: a wearable multiparameter medical monitoring and alert system. IEEE Trans Inform Technol Biomed 2004;8(4):415–27.
[2] Curtis DW, Pino EJ, Bailey JM, Shih EI, Waterman J, Vinterbo SA. SMART – an integrated wireless system for monitoring unattended patients. J Am Med Inform Assoc 2008;15(1):44–53.
[3] Malan D, Fulford-Jones T, Welsh M, Moulton S. CodeBlue: an ad hoc sensor network infrastructure for emergency medical care. In: MobiSys 2004 workshop on applications of mobile embedded systems (WAMES 2004); 2004.
[4] Chakravorty R. Proceedings of the fourth annual IEEE international conference on pervasive computing and communications workshops (PERCOMW'06); 2006. p. 532–6.
[5] Dembeyiotis S, Konnis G, Koutsouris D. Integrating legacy medical data sensors in a wireless network infrastructure. In: Proceedings of the 2005 IEEE, engineering in medicine and biology 27th annual conference, Shanghai, China; 2005. p. 2232–5.
[6] Arisoylu M, Mishra R, Rao R, Lenert LA. 802.11 Wireless infrastructure to enhance medical response to disasters. In: Proceedings of american medical informatics association 2005 annual symposium (AMIA'05), Washington; 2005. p. 1–5.
[7] Lenert LA, Palmer DA, Chan TC, Rao R. An intelligent 802.11 triage tag for medical response to disasters. In: Procceedings of american medical informatics association 2005 annual symposium (amia'05); 2005. p. 440–4.
[8] Chang C-K. A mobile-IP based mobility system for wireless metropolitan area networks. In: Proceedings of the 2005 international conference on parallel processing workshops (ICPPW'05); 2005.
[9] Pollard JK, Rohman S, Fry ME, 2001. A web-based mobile medical monitoring system. In: International workshop on intelligent data acquisition and advanced computing systems: technology and applications. Foros, Ukraine; 2001. p. 32–35.
[10] Çıbuk M, Balık Hasan H. Biomedikal Ağlar için Yeni Bir İletişim Uygulama Modeli, Fırat Univ. J Eng 2010;22(1):95–109. Elazığ.
[11] Baran A, Kılağız Y, Akaltun Y., ve Orman K. IEEE 802.11b Kablosuz TCP/IP Ağlarında Mikro denetleyici ve Tümleşik Web Sunucu Kullanılarak Bir Biyotelemetri Sistemi Geliştirilmesi, Ağ ve Bilgi Güvenliği Ulusal Sempozyumu ve Sergisi; 2005. p. 78–85s [İstanbul].
[12] Jurik AJ, Weaver AC. Remote medical monitoring, computer. IEEE Comput Soc 2008:96–9.
[13] Tarin Sauer C, Traver Sebastia L, Santamaria Gomez JF, Rocafull P.M., Cardona Marcet N. Bluetooth-3G wireless transmission system for neural signal telemetry. In: Wireless Telecommunications Symposium, WTS 2007; 2007. p. 1–6.
[14] Jason WP, Lo Ng, Wells BPL, Sloman O, Yang M, Peters G-Z et al. Ubiquitous monitoring environment for wearable and implantable sensors (UbiMon). In: Proceedings of the 6th international conference on ubiquitous computing (UBICOMP'04), Nottingham, UK; 2004.
[15] Jea D, Srivastava M. A remote medical monitoring and interacting system. In: Proceedings of the 4th international conference of mobile systems, application and services (MobiSys'06), Upsala, Sweden; 2006.
[16] Baran A, Kılağız Y. A Biotelemetry system with MicroController and Integrated Web Server in Wireless IEEE 802.11b TCPIP Network. Erciyes Üniversitesi Fen Bilimleri Enstitüsü Dergisi 2006;22(1–2):1–10.
[17] Fidan U, ve Güler NF. 4 Kanallı Biyotelemetri Cihazı Tasarımı, Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, Cilt 2007;22(1):7–12 [Sayfa].
[18] Krohn M, Kopp H, Tavangarian D. A wireless architecture for telemedicine. In: 4th Workshop on positioning, navigation and communication 2007. (WPNC'07), Hannover, Germany; 2007. p. 109–11.
[19] Braunstein B, Trimble T, Mishra R, Manoj BS, Rao R, Lenert L. Feasibility of using distributed wireless mesh networks for medical emergency response. In: AMIA 2006 symposium proceedings; 2006. p. 86–90.
[20] Farshchi S, Pesterev A, Nuyujukian PH, Mody I, Judy JW. Bi-Fi: an embedded sensor/system architecture for remote biological monitoring. IEEE Trans Inform Technol Biomed 2007;11(6):611–8.
[21] Arabshian K, Schulzrinne H, A SIP-based medical event monitoring system. In: Proceedings 5th international workshop on enterprise networking and computing in healthcare industry 2003 (Healthcom 2003); 2003. p. 66–70.
[22] Wood A, Virone G, Doan T, Cao Q, Selavo L, Wu Y et al. ALARM-NET: wireless sensor networks for assisted-living and residential monitoring, technical report CS-2006-01. Department of Computer Science, University of Virginia; 2006.
[23] Gao T, Greenspan D, Welsh M, Juang R, Alm A. Vital signs monitoring and patient tracking over a wireless network In: 27th annual international conference of the engineering in medicine and biology society 2005 (IEEE-EMBS 2005), Shanghai; 2005. p. 102–5.
[24] Jafari R, Dabiri F, Brisk P, Sarrafzadeh M. CustoMed: a power optimized customizable and mobile medical monitoring and analysis system. In: Proceedings of ACM HCI challenges in health assessment workshop in conjunction with proceedings of the conference on human factors in computing systems (CHI '05), Portland, Ore, USA; 2005.
[25] European mobihealth project, MobiHealth Projesi resmi web sitesi <http://www.mobihealth.org/> [26.06.09].
[26] Junker H, Stager M, Tröster G, Blttler D, Salama O. Wireless networks in context aware wearable systems. In: Proceedings of the 1st European

workshop on wireless sensor networks (EWSN '04), Berlin, Germany, January 2004. p. 37–40.

[27] Iraqi A, Barwicz A, Mermelstein PZ, Morawski R, Bock WJ. Design of a wireless communications module for telemetry in civil infrastructure monitoring. IEEE Trans Instrum Measure 2003;52(3):973–7.

[28] Ro J-h, Kim H-j, Ye S-y, Jung J-h, Jeon A-y, Kim Y-j, et al. Development of indwelling wireless pH telemetry of intraoral acidity. Int J Biol Med Sci 2008;1(1):50–4.

[29] Ren H, Meng M, Cheung C. Experimental evaluation of on-body transmission characteristics for wireless biosensors. In: Proceedings of the 2007 IEEE international conference on integration technology, Shenzhen, China; 2007. p. 745–750.

[30] Greenstein B, Mar C, Pesterev A, Farshchi S, Kohler E, Judy J et al. Capturing high-frequency phenomena using a bandwidth-limited sensor network. In: SenSys'06: proceedings of the 4th international conference on embedded networked sensor systems; 2006. p. 279–92.

[31] Wood AD, Selavoi L, Stankovic JA. SenQ: An embedded query system for streaming data in heterogeneous interactive wireless sensor networks. In: Proceedings of the 4th IEEE international conference on distributed computing in sensor systems; 2008. p. 531–43.

[32] Wood AD, Selavo L, Stankovic JA. SenQ: an extensible query system for streaming data in heterogeneous interactive wireless sensor networks. Technical report CS-2008-01, Department of Computer Science, University of Virginia; 2008.

[33] Yiğit E. GPS Teknolojisi ile Konum Tespit Sistemi Tasarımı, Yüksek Lisans Tezi, Beykent Üniversitesi, 50 sayfa; 2009

[34] Kahveci M, Yıldız F. *GPS Global Konum Belirleme Sistemi Teori ve Uygulama*, Nobel Yayın Dağıtım, 2. Baskı, 215 sayfa; 2005.

[35] Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E. A survey on wireless sensor networks. Comput Networks (Elsevier) 2002;38(4):93–422.

[36] Akyildiz IF, Melodia T, Chowdhury KR. A survey on wireless multimedia sensor networks. Computer Networks, vol. 51. Elsevier; 2007. pp. 921–960.

[37] Learn networking, How encapsulation works within the TCP/IP Model. <http://learn-networking.com/tcp-ip/how-encapsulation-works-within-the-tcpip-model> [03.06.09].