# INTEGRATION OF MOBILE PHONES INTO LAN ENVIRONMENT

Assis. Prof. Hamid M. Ali & Mohammed Dhiyaa Al-Qassar
Computer Engineering Dept., College of Engineering, Baghdad University

**ABSTRACT**
The aim of this paper is to integrate mobile phones into LANs by connecting the two parties and providing the mobile with a transparent access to the services and resources hosted by the LAN. Quick assessment of the typical related approaches for accessing LAN (like WLAN and Bluetooth BNEP) shows that they are inappropriate for mobile phones. The paper further inspects the LAN access approach based on Bluetooth PAN Profile and marking up its potential drawbacks. A modified approach is proposed and implemented by altering the access nature or model from peer-to-peer communication to a client/server service providing, and switching from BNEP to RFCOMM protocol.

**الخلاصة**
يهدف البحث لتكامل الهاتف النقال مع شبكات الحاسوب المحلية LAN عن طريق ربطهما معا وتمكين الهاتف النقال من الوصول بشفافية للخدمات والموارد التي توفرها وتستضيفها الشبكة المحلية. ان التقييم السريع للطرق المتبعة في الربط مع الشبكة المحلية (مثل WLAN و Bluetooth BNEP) يظهر عدم ملائمتها للهاتف النقال. يعمد البحث الى تحليل طريقة الربط المعتمدة على Bluetooth PAN Profile ومن ثم يؤشر النقاط التي تحول دون ملائمتها للهاتف النقال. يلي ذلك طرح تعديلات مقترحة على الطريقة المذكورة وذلك من خلال استبدال نموذج ند-لند بنموذج خادم\عميل واستبدال بروتوكول BNEP ببروتوكول RFCOMM وتصميم وتطبيق نضام يعتمد هذه التعديلات.

**KEYWORDS**

**Mobile Phone, LAN Environment, Bluetooth, PAN, Symbian OS, S60**

## INTRODUCTION

The objective of current telecommunication environment is the availability of all communication services anytime, anywhere, to anyone, by a single identity number (address) and a pocket-sized communication terminal. Achieving these goals means that the communication system must support mobile clients, which requires that the communication terminals (that clients hold) are mobile too. [WINCH1998]

Although originally targeted for different scopes of applications, those two environments (of mobile phones and computer networks) can be joined together to come out with new horizons, tools and approaches for the current classical applications domain. Mobile phones will gain access to enormous volume of resources (which they lack) hosted by computer networks. On the other hand, computer applications can reach out and extend to the very large market segment of mobile phones (according to International Telecommunication Union the number of mobile subscribers is 2,137,069,400 subscribers, about 32% of the world population have mobile phones). [ITU2006]

The connection (integration) of mobile phones with computer networks is not a new aspect. It had already been well established in the enterprise (WAN) and micro (PAN) scales. The integration trend has not (widely) commercially targeted the in-between local (LAN) scale yet. Due to several reasons (discussed later in the paper), the current standards and technologies used successfully to connect other types of mobile devices (like PDAs and laptops) to the LAN are usually impractical to be used with mobile phones. This problem is addressed by the paper by proposing an approach to connect mobile phones to LAN and integrating the two environments by providing the mobile phone with a proper and transparent access to the resources and services hosted by the LAN.

- **MOBILE PHONE CONNECTIVITY**

Mobile phone's connectivity to computer networks can be categorized into three models according to the scale of the geographic area covered by the connected network. Each model (or category) has its unique architecture and applications. [PAN 2003]

**Wide and Metropolitan Area Network (WAN/MAN) Model**

Mobile phones use data carrier services (embedded within the cellular mobile phone network and usually provided by the network operator), like GPRS and EDGE, to connect to WAN/MAN networks. Since these services are part of the mobile network, and since mobile networks are interconnected all over the world with other mobile networks and computer networks (like the Internet), then WAN/MAN networks provide virtually a global coverage.

An illustration of this model is shown in Fig. 1. Mobile phone sends/receives data packets over GPRS, EDGE or similar data carrier services. All the data is directed to/from data network gateway, which links the mobile and data networks together. The gateway translates between data carrier service format and the destined data network format.

This model is suitable for such applications as web browsing, emails and other public Internet services. It also can be used with private enterprise applications where a connection to a private computer network is required. This connection is achieved either through the Internet by using VPN approach or by making a direct connection between the private computer network and the data network gateway.

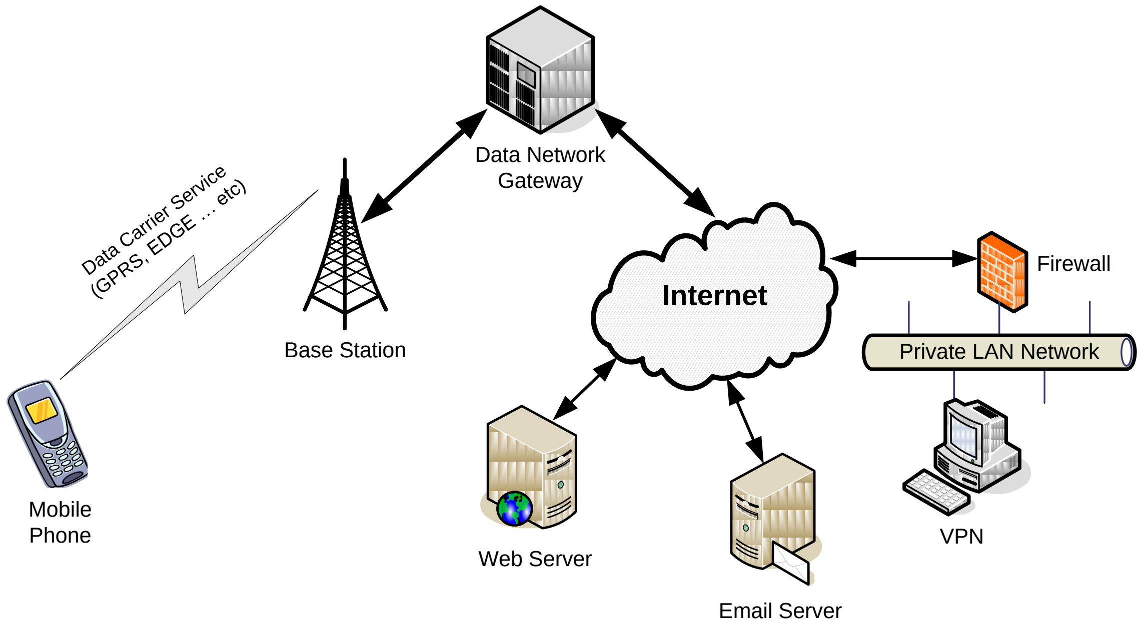


**Fig. 1** – WAN/MAN Connectivity

## Local Area Network (LAN) Model

Mobile phones use wireless LAN (WLAN) standards, like IEEE 802.11, to connect to wireless computer networks, or to conventional wired computer networks through a wireless access point. Such wireless networks cover a typical range of 100m (usually the whole network is located within single building).

An illustration of this model is shown in Fig. 2. This model can be used for private custom applications where a high end mobile phone with advanced specifications (large memory, fast processor, high resolution display and wireless network interface card) is used as a very light hand held smart terminal (the same function as PDA).

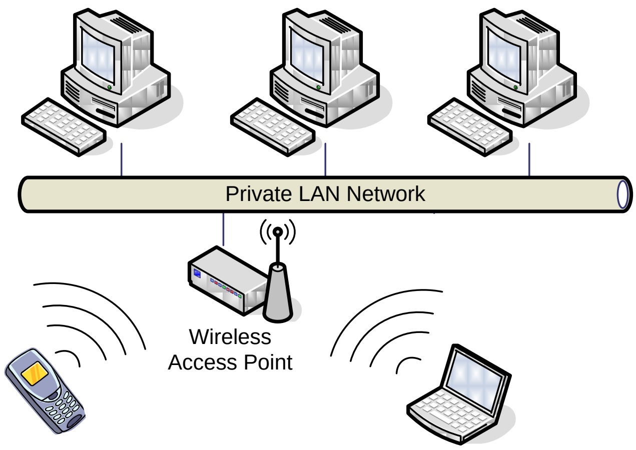


**Fig. 2 –** LAN Connectivity

## Personal Area Network (PAN) Model

Mobile phones use short-range wireless technologies (like infrared or Bluetooth) or short distance wired standards (like RS-232 or USB) to form or to join a PAN network. Such networks cover a typical range of 10m (usually the networked devices are located in the vicinity of a single person).

An illustration of this model is shown in Fig. 3. Possible applications of this model would be connecting the mobile phone to a computer in order to transfer files (ring tones, wallpapers … etc.) between the two devices, synchronizing and backing up important data (address book, schedule, calendar … etc.) or using the mobile phone as a modem by the computer. All of these applications are based on exchanging data between the mobile phone and the computer that it is directly connected to; therefore, PAN connectivity is used as a cable replacement by mobile phones (and similar mobile devices).

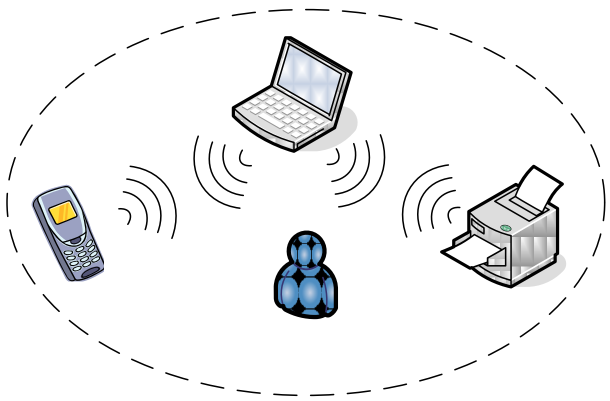

**Fig. 3 –** PAN Connectivity

• **ACCESSING LAN THROUGH PAN**

It is possible to use PAN to access a remote network (LAN or WAN/MAN) through a Network Access Point (NAP), Fig. 4 shows this approach. NAP is a device that is connected to both PAN and the remote network. The NAP can be a dedicated specialized device or just a conventional networked device like desktop computer. The main job of NAP is to act as a link through which other devices in the PAN can reach the remote network. NAP may operate at different levels (bridge, router, gateway or proxy) depending on the differences between the two networks it is connected to. [PAN2003]

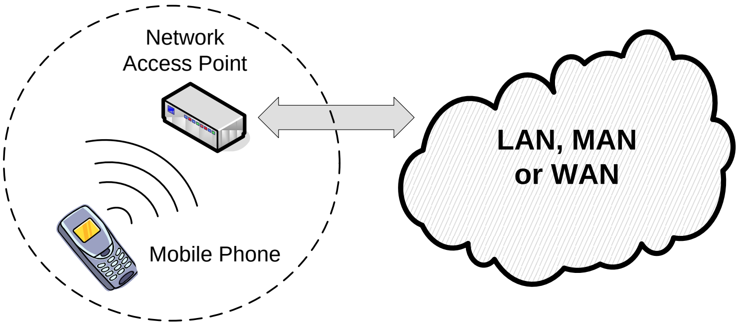


**Fig. 4 –** Connecting to Remote Networks via PAN

LAN-through-PAN can be important for mobile phones (as compared to using PAN to access WAN/MAN) for the following reasons:

- WLAN is hardly ever supported by mobile phones for two main reasons. First, WLAN consumes a lot of power, which means the typical mobile phone battery will be drained quickly. Second, mobile phones with WLAN are very expensive. Therefore, the number of mobile phone models, their prices and their sales volume make the LAN connectivity model virtually impractical.
- WLAN is a new aspect in the mobile phone industry as compared to PAN (infrared, USB and Bluetooth) which is a well-known feature that is offered by many models previously and most of the models currently. Hence, by using PAN to access LAN there is no need to replace the current mobile phone devices with newer ones in order to achieve LAN connectivity.
- Although private LANs can be accessed through WAN/MAN model, but this requires the approval of a third party represented usually by the mobile network operator to use the data service provided by the network. This scheme also requires either connecting the private LAN to the Internet and using VPN, or having a direct access to the mobile network data service. All these requirements will increase the cost, security risks and management complexity.

• **BLUETOOTH**

As mentioned before, PAN can use different communication technologies, but the most popular and suitable is Bluetooth especially regarding mobile phones. Bluetooth is a wireless communication standard for cable replacement between digital devices by using simple, low cost and low power communication modules with non-directional short range, frequency hopping and ad-hoc radio links. Therefore, it meets all the requirements imposed by the nature and purpose of PAN (in general) and limitations of mobile phones (in particular). Beside that, several older phone models and almost all current models are supplied with Bluetooth. Hence, Bluetooth will be a logical choice to implement a LAN-through-PAN solution.

## Bluetooth System Architecture

Bluetooth system architecture is segmented into several layers. These layers, some of which are shown in Fig. 5, form the Bluetooth protocol stack that can be logically partitioned into the following groups: [GANGULI2002]

- **Transport protocol group**: facilitates the identification of other Bluetooth devices. It also configures and manages the physical and logical links, allowing the higher protocol layers to transmit data through these layers. The layers of the transport protocol group work together and form a virtual pipe that is used to transport data from one device to another. This group includes radio, baseband, Link Manager Protocol (LMP), Host Controller Interface (HCI) and the Logical Link Control and Adaptation Protocol (L2CAP) layers.

- **Middleware protocol group**: provides additional protocols that help new and existing applications to operate over Bluetooth links. It consists of both third party and industry standard (usually Internet related) protocols and the protocols defined by the Bluetooth Special Interest Group (SIG) for wireless communication in Bluetooth devices. The third party and industry standard protocols include protocols such as Point-to-Point Protocol (PPP), Internet Protocol (IP) and Transmission Control Protocol (TCP) while the protocols defend by the SIG include protocols such as Radio Frequency Communication (RFCOMM), Bluetooth Network Encapsulation Protocol (BNEP) and Service Discovery Protocol (SDP).

- **Application Group**. The application group includes the actual applications that use Bluetooth links. These applications are not necessarily aware of the Bluetooth wireless communication. An example of such applications is a Web browsing client.

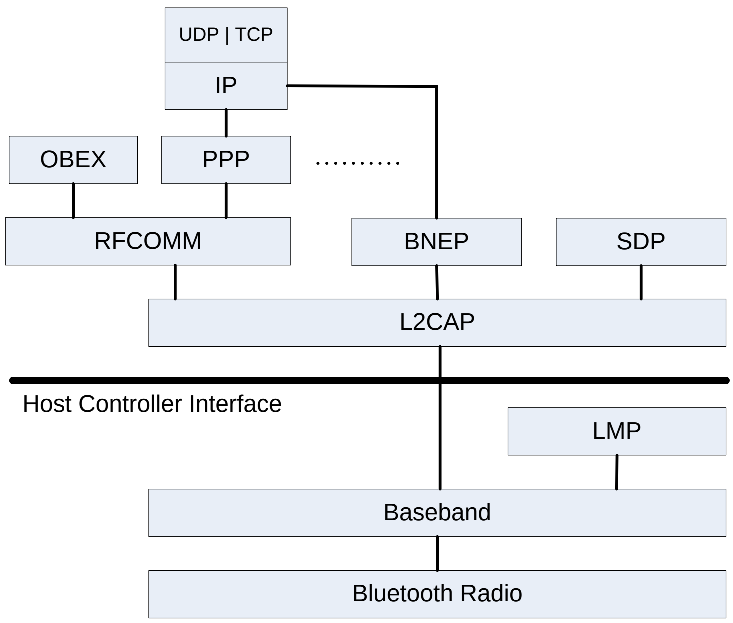


**Fig. 5 –** Bluetooth Protocol Stack

## Bluetooth Protocols

The core Bluetooth protocols defined by the Bluetooth specification, and shown in Fig. 5 are: [GANGULI2002]

<u>Bluetooth Radio</u> helps to transmit and receive data over the air. The frequency band used by Bluetooth devices is the globally available unlicensed ISM band ranging from 2400 MHz to 2483.5

MHz. Within the band, Bluetooth has 79 channels spaced 1 MHz apart. The choice of this band takes into consideration the size and power limitations of most mobile devices.

Baseband is responsible for the determination and instantiation of the air interface by defining processes to identify other devices and establish connectivity between them. The key functions performed by the baseband include connection creation, selection of a frequency hopping sequence, timing hops, operations related to power control and security, packet processing and the selection of link types. The baseband layer and the radio layer can be functionally correlated to the physical layer of the Open Systems Interconnection (OSI) reference model.

LMP performs all the functions related to link management. These functions include for example link setup, link security and link configuration. In addition, LMP layer offer other management services like time management, mode control management and power management. The LMP uses the underlying services of the baseband layer to manage the links; therefore, it does not perform operations related to information delivery.

L2CAP is specifically concerned with functions such as protocol multiplexing (sharing of the air interface between multiple protocols and applications), segmentation and reassembly of data packets and negotiating an acceptable level of service.

RFCOMM provides the protocol stack with the support of serial communication similar to that in cable technology. RFCOMM provides a Protocol Data Unit (PDU) structure to emulate the RS-232 control and data signals over the baseband layer. The upper level services are thereby provided with capabilities to transport both control and data signals by using the serial line transmissions.

SDP is a simple protocol that allows devices to locate and gather information about the services of other devices. SDP involves communication between an SDP server and an SDP client. The server maintains a list of service records that describe the characteristics of services associated with the server. Each service record contains information about a single service. A client may retrieve information from a service record maintained by the SDP server by issuing an SDP request. SDP uses a request/response model where each transaction consists of one request PDU and one response PDU.

BNEP is used to transfer both control and data packets over Bluetooth to provide networking capabilities for Bluetooth devices. BNEP encapsulates packets from various networking protocols, which are transported directly over the Bluetooth L2CAP. [BNEP2003]

## Bluetooth Network Topology

The connection between Bluetooth devices can be either point-to-point (between two Bluetooth devices) or point-to-multipoint (several Bluetooth devices connected to each other). Either way, a Bluetooth connection creates a network known as a Piconet. Each piconet consists of one master controlling unit and up to seven active slave units. All Bluetooth units participating in the piconet are time- and hop-synchronized to the same physical channel. The master unit's system clock and Bluetooth address determines the hopping phase and sequence of the piconet physical channel. The software and hardware specifications of a master and slave unit are the same. However, the unit that establishes a piconet becomes the master unit controlling all traffic within the piconet.

The spectrum of connectivity in a piconet is increased by formation of multiple piconets in the same area. Overlapping piconets form a Scatternet. As a result, a Bluetooth device can also participate in

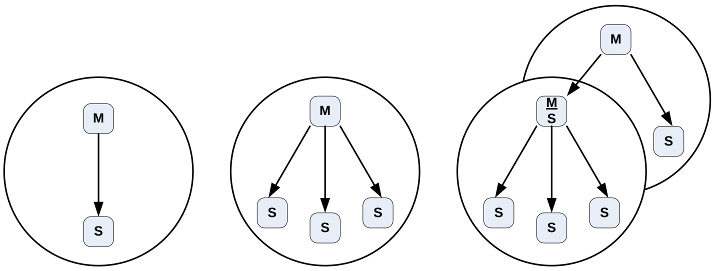


**Fig. 6 –** Bluetooth Piconet and Scatternet

multiple piconets at the same time (but not as a master in both of them). A piconet and scatternet are shown in Fig. 6. [GANGULI2002]

## * BLUETOOTH PAN PROFILE

The Bluetooth PAN Profile describes how two or more Bluetooth enabled devices can form an ad-hoc network and how the same mechanism can be used to access a remote network through a NAP.
Bluetooth PAN Profile is based on BNEP, which is used to provide networking capabilities for Bluetooth devices. Several device roles are defined by the profile to organize cooperation and interaction between PAN devices. [PAN2003]

### BNEP Protocol

If two applications run on two different devices within the Bluetooth network but each application uses a different network layer protocol (for example IP and IPX over Bluetooth), then in order for the two applications to interoperate and exchange information, a common packet format needs to be defined to encapsulate network layer protocols over the Bluetooth media.

The same thing applies for the scenario when one of the devices is part of a Bluetooth network and the other is part of another type of network (like Ethernet). BNEP encapsulates packets from various networking protocols, which are transported directly over the Bluetooth L2CAP.

BNEP is implemented using connection oriented L2CAP channels. The Bluetooth is considered as a transmission media in the same OSI layer as Ethernet, Token Ring, etc. L2CAP is considered as the Bluetooth MAC layer. The Bluetooth device address space is administrated by the IEEE, and is assigned from the Ethernet address space. This means that it is possible to build a Bluetooth network access point as a bridge between Bluetooth devices and an Ethernet network.

The use of the BNEP for transporting an Ethernet packet is shown in Fig. 7. BNEP removes and replaces the Ethernet header with the BNEP header. Finally, both the BNEP header and the

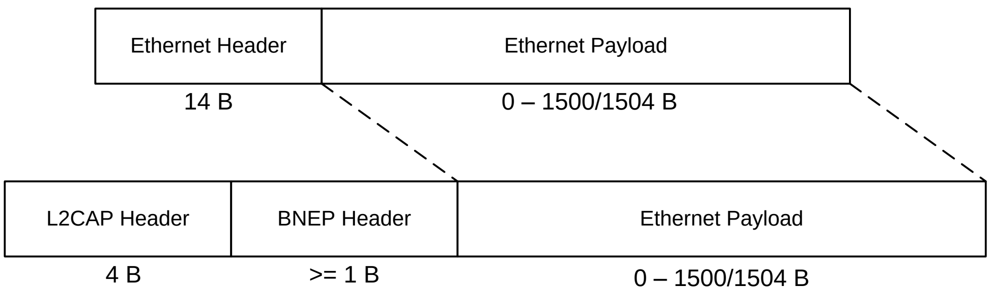


**Fig. 7 –** Using BNEP to Transport Ethernet Packets

Ethernet payload is encapsulated by L2CAP and sent over the Bluetooth media. [BNEP2003]

**PAN Profile Device Roles**

Bluetooth PAN Profile defines three roles or services:

- Network Access Point (NAP): A Bluetooth device that supports the NAP service is a Bluetooth device that provides some of the features of an Ethernet bridge. The device with the NAP service is simply referred to as NAP. The NAP and the PAN User exchange data using BNEP. NAP has an additional network connection to a different network media in which the Ethernet packets are either exchanged via Layer 2 (Data Link) bridging or Layer 3 (Network) routing mechanism.

- Group Ad-hoc Network (GN): A Bluetooth device that supports the GN service is able to forward Ethernet packets to each of the connected Bluetooth devices (PAN Users) as needed. The GN and the PAN User exchange data using BNEP. GN does not provide access to any additional networks. Instead, a GN is intended to allow a group of devices to form temporary network and exchange information.

- PAN User (PANU): This is the Bluetooth device that uses either the NAP or the GN service. PANU supports the client role for both the NAP and GN role.

The NAP/GN performs Ethernet Bridge functions to forward Ethernet packets from one PANU to another PANU or from a PANU to another network. NAP/GN regards each established Bluetooth BNEP connection as a valid Bridge Port. Thereby the NAP/GN shall perform bridging between all of the BNEP connections. In addition, NAP regards the optional Ethernet connection as a valid Bridge Port too (if the NAP is acting as a bridge and not as a router, otherwise a normal routing is performed by the NAP). Fig. 8 shows the protocols used in each of the three roles and their interaction with each other. [PAN2003]

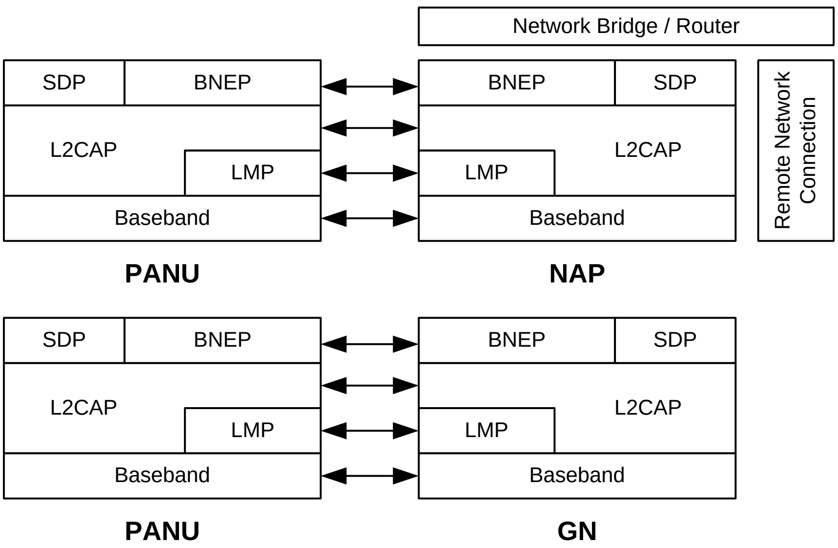


**Fig. 8** – NAP, GN and PANU Device Roles

- **PROPOSED MODIFICATIONS TO BLUETOOTH'S PAN PROFILE**

Bluetooth SIG engineered the Bluetooth protocol stack from ground up to support PAN. Furthermore a specific protocol, BNEP, and device role are designed to enable a Bluetooth device act as a NAP allowing other devices in the PAN to reach for LAN or WAN/MAN through it. However, using Bluetooth PAN Profile and BNEP to implement a LAN-through-PAN solution for integrating mobile phones with computer network services has the following downsides:

- The Bluetooth PAN Profile and BNEP were designed from the beginning so that more capable computing devices like desktop or laptop computers can benefit from them. That doesn't mean that these standards will not work with the less capable devices like mobile phones, but it means that no commercial mobile phone product has supported the PAN and BNEP yet (i.e. no current mobile phone model has a built-in support for PAN or BNEP). Therefore, in order to implement a LAN-through-PAN solution, one needs first to implement a BNEP protocol and a PAN profile.
- BNEP is a Bluetooth based protocol. It will not work with different networks like infrared or future short-range wireless standards. That means if a NAP is built based upon BNEP and Bluetooth PAN Profile, the only way to connect to this PAN is through Bluetooth. Although Bluetooth is very promising and efficient but future upgrades and network standards spectrum served by such a NAP would be limited to Bluetooth technology only.
- In order to make use of the BNEP and Bluetooth PAN Profile network, transport and application layers must be available on and supported by the Bluetooth device (PANU). As mentioned before, BNEP and Bluetooth PAN Profile were designed with minds set to desktops, laptops and similar computing devices as main targeted devices. Such rich resourced devices can easily afford the complex structure of multilayered network protocol stack, especially for the upper and intermediate layers. On the other hand, mobile phones are limited in resources and a simpler networking structure is preferred.

Fig. 9 shows the proposed approach to implement a LAN-through-PAN solution. The following modifications are made to the PANU and NAP architecture, shown previously in Fig. 8:

- Bluetooth technology is still used between PANU and NAP; however RFCOMM is used instead of BNEP since, unlike BNEP, virtually every Bluetooth device supports RFCOMM.

- The NAP does not forward packets from PANU to PANU (i.e. does not perform GN role). It is unlikely that networking is required between the PANUs (mobile phones) and if it is, then they can connect to each others using separate one-to-one Bluetooth connections.

- The network bridge/router is replaced by a network services proxy at the application layer level. The proxy is loosely coupled with the lower networking layers so that the networking technologies and standards used can be upgraded and/or changed with no or minimum alteration to the proxy implementation itself. This makes the NAP more flexible and network type independent. In addition to that, the proxy acts on behalf the mobile phone to access the required network services and obtaining the results, relieving the mobile phone from carrying out network housekeeping tasks.

It is much more practical and appropriate to design mobile phones network solutions as a thin client with the minimum basic functionalities and relocate all the complex tasks to a more capable device that will act as a fat server (which in our case can be the NAP, a dedicated server or both of them).

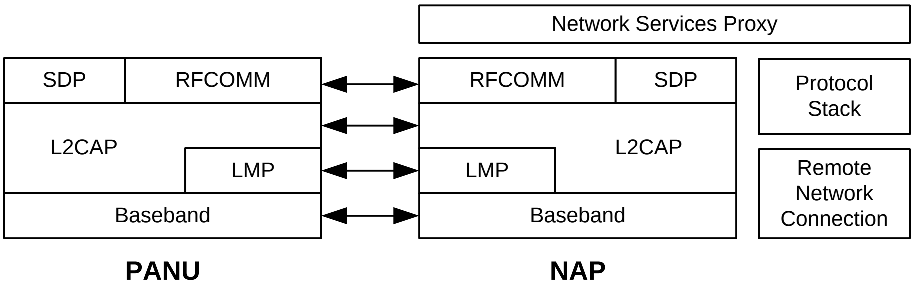


**Fig. 9 –** Modified NAP and PANU Architecture

## * PROPOSED SYSTEM DESIGN

This section briefly demonstrates the design of a system that implements LAN-through-PAN model based on the modified Bluetooth PAN profile that is proposed by this paper.

### System Components

Fig. 10 shows the (high level) components of the system. The mobile phone is hooked to the computer network by a modified network node (NAP). NAP acts like a gateway that connects to Bluetooth network at one side, and to a wired or wireless computer network at the other side. The mobile phone, then, can select a Service Access Point (SAP) to talk to. The SAP is another modified network node whose job is to act as a window for the mobile phone from which it can view the available resources and services in the network. The mobile phone issues the request for a resource or a service to the SAP and the later relays this request as if it was issued by the SAP itself. The SAP then relays the received response back to the mobile phone. Notice that the resources and services can be hosted by any node in the network including the NAP and the SAP. The functions of the NAP and SAP can be joined as a single function and hosted by a single node, but splitting them into two different identities (so that they can be hosted by two or more nodes) enhances the mobility and reduces the redundancy.

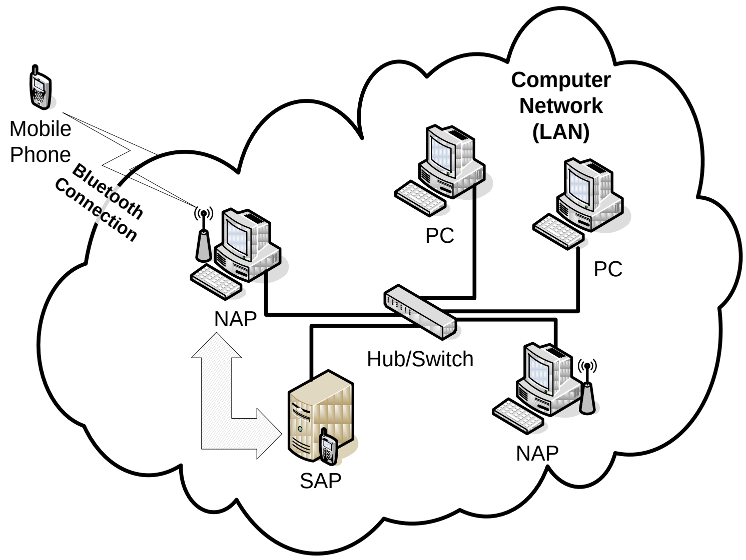


**Fig. 10** – System Components

### System Architecture

Fig. 11 shows the system architecture. The interaction between system components must follow a certain protocol. The proposed protocol is divided logically into three layers, the lower three layers in Fig. 11, each has a well defined function and each is independent from the others'

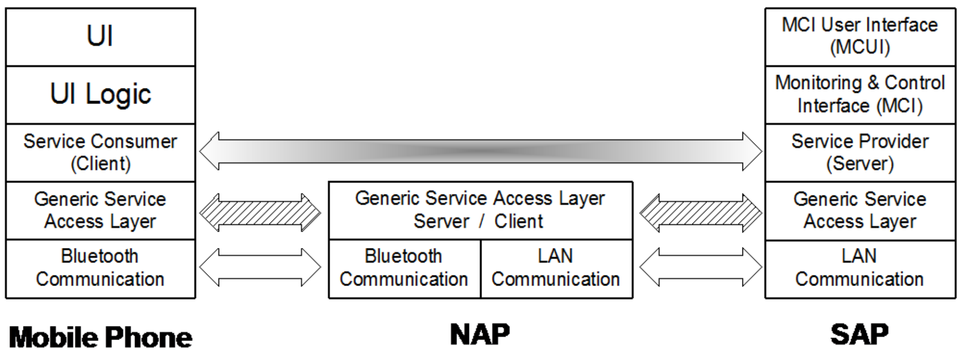


**Fig. 11** – System Architecture

implementation. Each layer relies on and uses the service provided by the lower one.

**Communication Layer:** is responsible for providing the networking services for the upper layers in a high-level abstracted mode. It does that by using the more specific and detailed low level networking services offered by the OS to provide streamed network communication that appears to the upper layer as memory buffer. This layer can be thought of as an abstraction to the transport, network, data link and physical layers. According to the underlying network type, this layer can be either a Bluetooth communication layer or LAN communication layer.

**Generic Service Access Layer (GSA):** provides service-independent or service-neutral functions. Such functions are not limited to a certain type of services or resources; instead, it is more generic and may be needed regardless of the service or resource being accessed. For example, it maintains Access Parameters Registry (APR) at NAP. It also provides multiplexing capability so that several Service Provider layer instances can run on the same SAP.

**Service Provider/Consumer layer:** is responsible for providing services and consuming them. Being service dependent, there might be several versions of this layer running simultaneously above the GSA layer at both of the mobile phone (as a consumer) and the SAP (as a provider).

The remaining upper layers are not related to the interaction protocol between the system components. Instead, they perform functions local to the component on which they reside. For example, the UI and UI Logic layers on the mobile phone are responsible for managing and rendering the graphical presentation of application data, while the Monitoring and Control Interface (MCI) and MCI UI layers on the SAP are responsible for monitoring and configuring the operation of the SAP. These layers are not part of the core system but can be considered as add-ons.

**PDU Formats**

Exchanging of information between peer layers takes the form of PDUs, for example two separate GSA layers communicate by exchanging (virtually) GSA PDUs. Typically, a layer will take the upper layer's PDU and use it as payload, add layer's specific header to it then pass it down to the lower layer and so on till it reaches the physical layer where the actual data transmission takes place.

Fig. 12 shows the different PDU formats used by the system layers. The service request can be of any format since it is service dependent. For the system implemented as an example, where it provides a file access service similar to File Transfer Protocol (FTP), the request takes the format of a command field followed by a parameters/data field. The command field is of a fixed length (3 Bytes) and text based (although it can be used as a binary field too). The parameters / data field is of a variable length and extends to the end of the packet. This format is simple but yet very flexible and can be used for a very wide range of services.

The service provider/consumer layer uses the Request_PDU format to transfer requests from the mobile phone to the SAP. The 1 Byte m field specifies the length (maximum of 255 Bytes) of the Service Provider Address field. The Payload field contains the service request to be transferred to the addressed service provider.

The GSA layer uses the GSA_PDU format to carry requests. It is identical to the Request_PDU format, but it has GSA Address field instead. The Payload field represents a Request_PDU to be relayed to the addressed GSA layer. A special case of the GSA_PDU is when the r field has the value of 0 (zero). This format is used only for certain PDUs sent to the NAP GSA layer by the GSA layer of a mobile phone or SAP. In this case, the GSA_PDU does not carry a request or response;

instead, it represents a GSA command (GSA request) that is to be served by the NAP GSA layer. The GSA Address is removed and the rest of the PDU consists of a text based Command field (3 Bytes) followed by a parameters/data field.

When a response is transferred between any two layers, it is transferred as raw data. That means no headers are needed. Each node the request is relayed to, keeps the connection on which the request arrived open. Therefore, by the time the request reaches its final destination, the complete path is kept alive (session-oriented or connection-oriented communication). Hence, only request carrying PDUs need to have address fields.

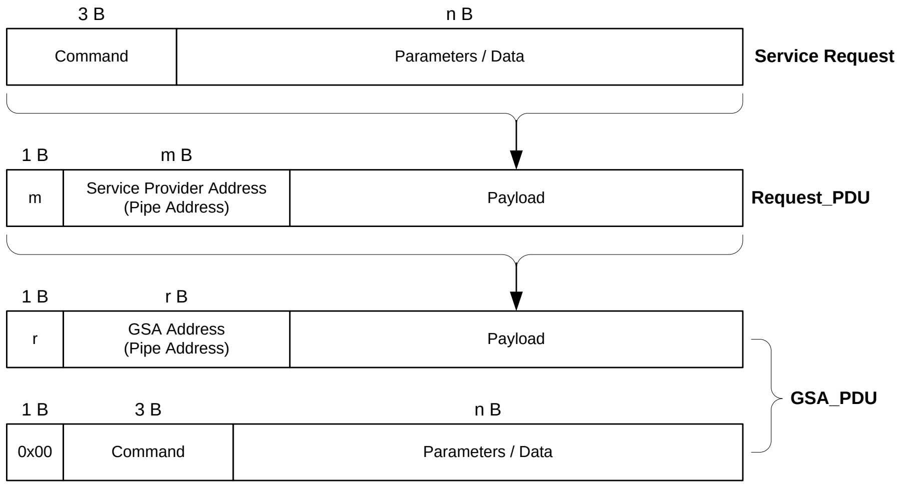


**Fig. 12 –** PDU Formats

**Fig. 13 shows the usage of different PDU formats by system layers.**

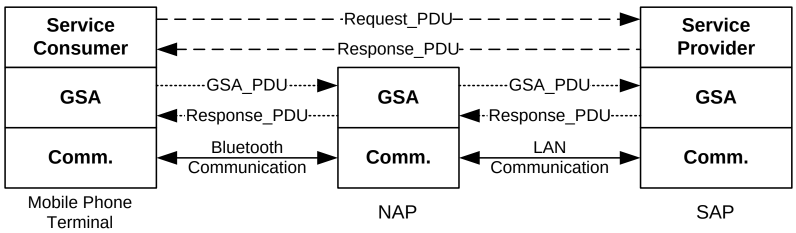


**Fig. 13 –** PDU Formats Usage

## System Dynamic Behavior Model

The system activity diagram, Fig. 14, is a general simplified layout for the system behavior. The diagram is divided into four vertical lanes following the location the activity took place at. Again, the diagram is simplified and general; therefore not all layers and their activities are present. Instead, only the key role playing layers and abstract activities are shown by the diagram.

Fig. 14 summarizes the system layers and components into four abstract items: First is the mobile phone, which includes all the layers on the mobile terminal. Second is the NAP/GSA, which includes all the layers on the NAP with emphasizing on the GSA layer. Third is the SAP/GSA, which represents the SAP's GSA layer. Fourth is the SAP/SP, which represents the SAP's service provider layer. Notice that the vertical axis, which represents general time progress, does not flow linearly and equally in each component (lane).

The activity flow in NAP/GSA, SAP/GSA and SAP/SP is started by the administrator when he/she starts up the software on both the SAP and the NAP. After initialization, the activity flow waits for incoming requests: NAP/GSA waits for requests from mobile phones on a Bluetooth connection, SAP/GSA waits for requests relayed from a remote NAP/GSA on a LAN (pipe) connection, SAP/SP waits for requests relayed from a local SAP/GSA on Inter Process Communication (IPC) (pipe) connection.
Upon the detection of such request, the activity flow forks into two parallel paths: the first path returns to the request wait state in order to accept other requests, the second path performs further activities to serve the accepted request and provide the proper response.
In NAP/GSA, the request is examined to decide if it can be served by the NAP/GSA (labeled as GSA request in the activity diagram) or it needs to be relayed to a SAP (labeled as SP request). An example of GSA request is when a mobile phone queries the NAP for available SAPs. In both cases, when the response is valid it is relayed to the mobile terminal.

In SAP/GSA, the request is examined to determine which service provider layer instance it should be relayed to (de-multiplexing), when the response is valid it is relayed to the NAP/GSA. In SAP/SP, the request is served and the proper response is compiled and relayed to the SAP/GSA.
On the other end of the diagram, the activity flow in mobile phone is started by the user (mobile

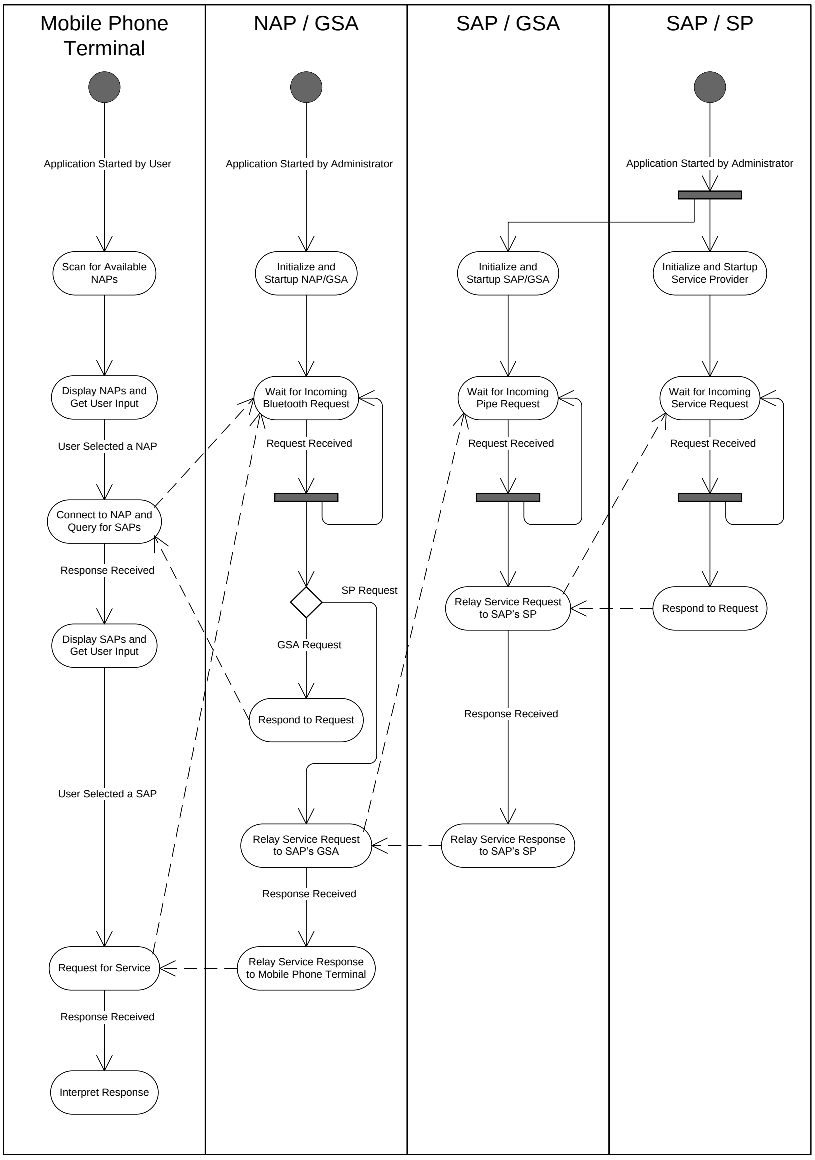


**Fig. 14 –** System Activity Diagram

owner) when he/she launches the application (component) at his/her mobile. A scan process for available NAP nodes in the neighborhood is initiated. The result is presented to the user as a list of NAPs and the user can select the NAP he/she wants to connect to. A Bluetooth connection to the selected NAP is established and a request is issued to that NAP to query for available SAPs (registered with this NAP's APR). The result is presented to the user as a list of SAPs and the user can select the SAP that provides the service he/she wants to use. A request (addressed to the user selected SAP) is issued and relayed to the NAP which the mobile phone is currently connected to. The result, which is the response relayed by the NAP, is interpreted by the mobile application to produce the expected output (in our case it is a file copied to the mobile phone file system).

**System Static Structure Model**

Fig. 15 is a class diagram representation of the overall system components. This diagram is not a complete one, but it focuses on the major classes that define the core system components and behaviors. It also shows the relations between these classes and how they interact together.

## * MOBILE PHONE SOFTWARE ENVIRONMENT

Mobile phone software environment is composed of two main elements, the Symbian OS and the Series 60 platform.

**Symbian OS**

Symbian OS is a 32-bit multitasking operating system, where events often happen asynchronously and applications are designed to interact with one another. For example, a phone call may interrupt a user composing an email message; a user may switch from email to a calendar application in the middle of a telephone conversation.

From the start, Symbian OS was designed for small, resource-constrained devices with wireless communications. Key design features are:

- Performance – Symbian OS is designed to make minimal demands on batteries and to have low memory footprint.
- Multitasking – All applications are designed to work seamlessly in parallel.
- Standards – The use of technologies based on agreed-upon standards is a basic principle of Symbian OS, ensuring that applications are robust, portable and interoperable.
- Object oriented software architecture.
- Memory management optimized for embedded software environment.
- Runtime memory requirements are minimized – very small executable sizes and ROM-based code that executes in place.
- Security mechanisms for enabling secure communications and safe data storage.
- Application support for international environment with built-in Unicode character sets.
- A rich and varied Application Programming Interface (API) allowing access to reusable components in developer applications. [NOKIA2004]

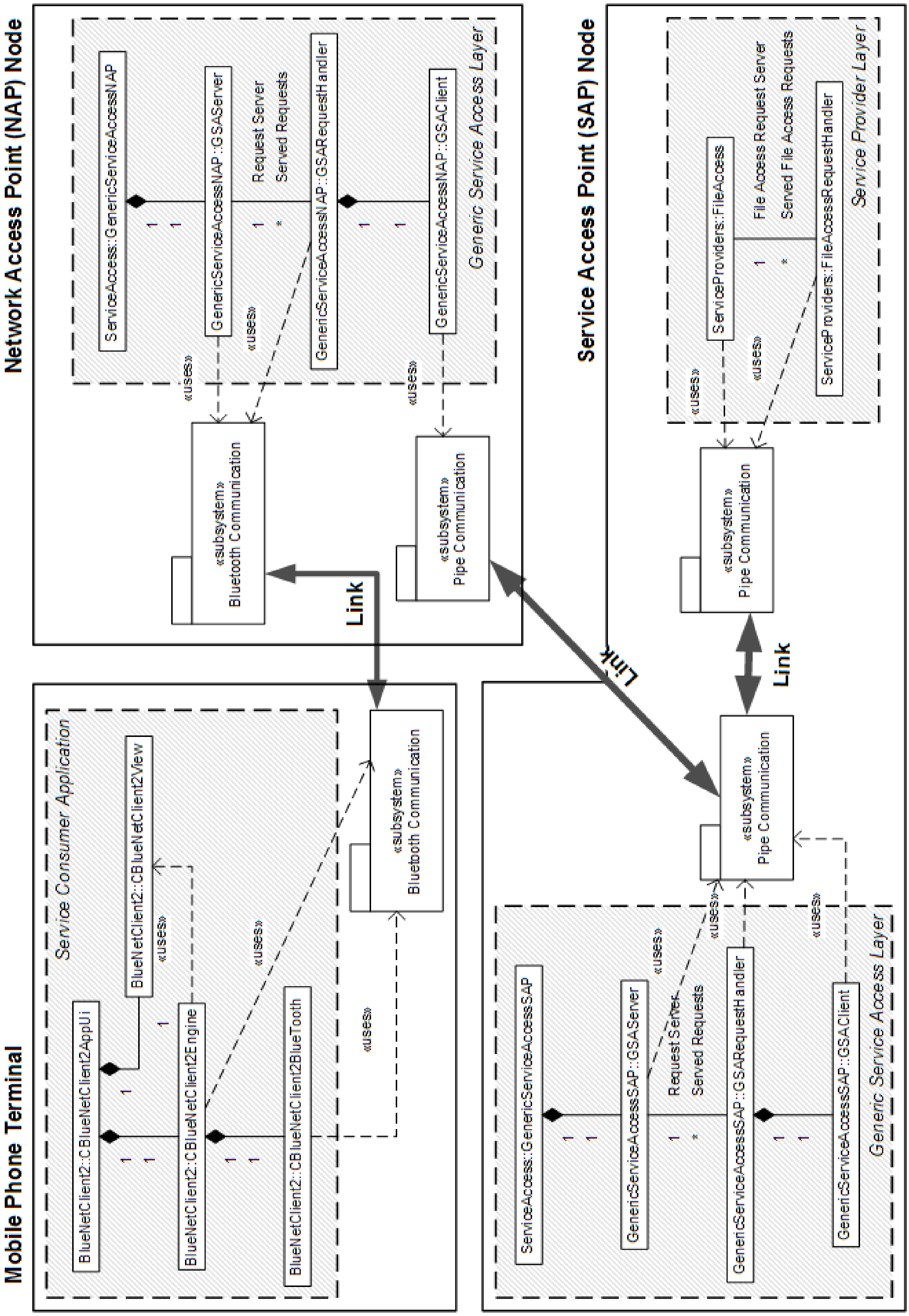


**Fig. 15 –** System Structure

Fig. 16 shows a representation of the Symbian OS components. The system kernel, file server and memory management are located in the Base operating system layer. The kernel manages system resources such as memory and is responsible for time slicing the applications and system tasks. Device drivers provide the control and interface to specific items of hardware – the keyboard, display, infrared port and so on. The upper layers of the system provide communication and extensive computing services such as data management, graphics, multimedia and security.

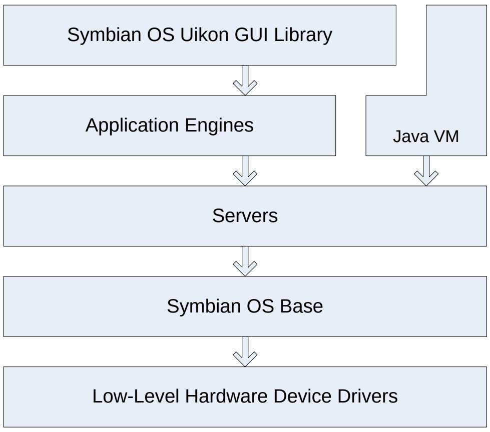


**Fig. 16 –** Symbian OS Architecture

Client/Server architecture is a key design feature of Symbian OS. User applications and system processes are clients that use the resources of a wide variety of system servers. Servers can be accessed only by their clients via well-defined interfaces. Virtually all servers run with a high priority, but without system privileges, to ensure a timely response to all of their clients while controlling access to the resources of the system. Some core application engines, written as servers, enable software developers to create their own user interfaces to the application data. Examples include contacts, calendar, multimedia services (decoding and rendering of image formats) and messaging. [LEIGH2004]

The kernel runs in privileged mode, it has access to the entire memory space. The process is a unit of protection in Symbian OS. Each process has its own virtual address space. The kernel assumes the existence of a Memory Management Unit (MMU), which is responsible for translating virtual addresses. The thread is a unit of execution in Symbian OS. A process has one or more threads. [NOKIA2003]

Memory intensive operations such as context switching are minimized. Symbian OS is primarily event-driven rather than multithreaded. Multithreading is possible but is avoided because it potentially creates several kilobytes of overhead per thread. Conversely, a primarily event-driven approach does not need any context switching and can have an overhead as low as a few tens of bytes. [NOKIA2004]

Event-driven (cooperative) multitasking approach is achieved via the use of Active Objects. Each active object is given a task to complete. Each thread has an Active Scheduler, which is responsible for running a waiting active object. A thread may contain one running active object and many active objects waiting to be run. Unlike multitasking with threads, once an active object has started to run, it cannot be preempted by another active object instead it is up to the running active object to give up use of the processor and join the waiting queue. The control is returned to the active scheduler to decide which waiting active object to run next. [NOKIA2003]

**Series 60 Platform**

The Series 60 (S60) platform, developed by Nokia, is a complete smart phone reference design that includes a host of wireless applications. The platform builds on the Symbian OS, complementing it with a configurable graphical user interface library.

The success of smart phone category devices is highly dependent on the availability of innovative applications and content from third parties – that is, the growth of the mobile services and applications businesses. Diversification between handset designs and capabilities has greatly increased. This has resulted in minimal similarity amongst competitive devices in terms of screen size, keypad, browser and other elements of the user interface. Applications, services and other content have to be adapted to these different devices.

Nokia has made the S60 platform available for licensing by other handset manufacturers enabling them to bring phones to market with equivalent and compatible functionality. Standardizing the application environment helps service creation and application interoperability. Common input methods, APIs and supported technologies allow services and applications to interoperate seamlessly, but still give licensees the freedom to innovate and design excellent smart phones. [NOKIA2004]

The core of S60 platform is Symbian OS. S60 platform adds the extensive Avkon UI layer, a full suit of applications based on the Avkon and Uikon libraries plus a number of key application engines. S60 platform contains the majority of the UI and framework APIs used by third party Graphical UI (GUI) applications. Fig. 17 shows the components of S60 platform.

The Avkon library defines many UI components and application framework components. Avkon builds on and extends the framework and controls provided in the generic Symbian Uikon library. Each GUI application is then based on framework classes provided as part of the Avkon library and on the UI layers below that – for example in Uikon. The S60 GUI library determines the rendering of all GUI elements, so the individual applications share a common look and feel. [LEIGH2004]

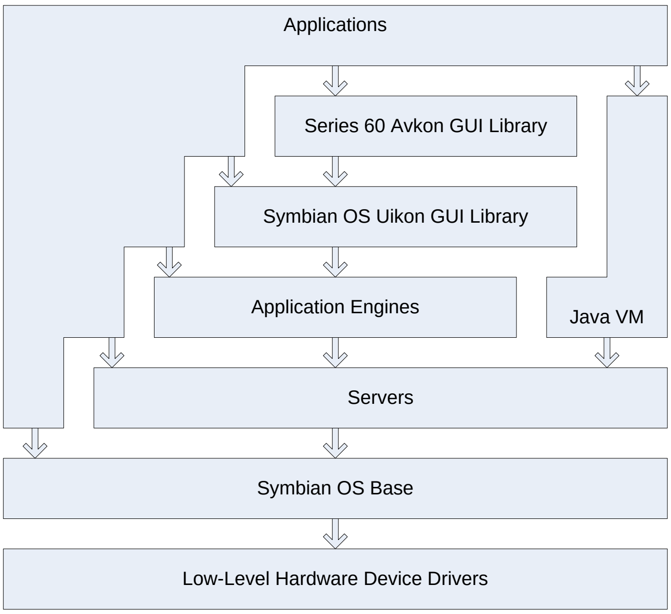


**Fig. 17 –** S60 Platform Architecture

*** RESULTS**

Fig. 18 shows screen shots for mobile phone component GUI during different user interactivities with the component. The initial screen (1) is shown when the user starts up the component application, the status pane displays the application title (**BlueNetClient2**) and icon, the main pane is empty (indicated by the **no data** string) and the control pane displays the two commands **Options** and **Back** that corresponds to the mobile terminal soft keys. **Back** is used to step back to the previous screen (and to exit the application when is used at the initial screen). **Options** command is used to display the options menu (2) at any point during the application execution.

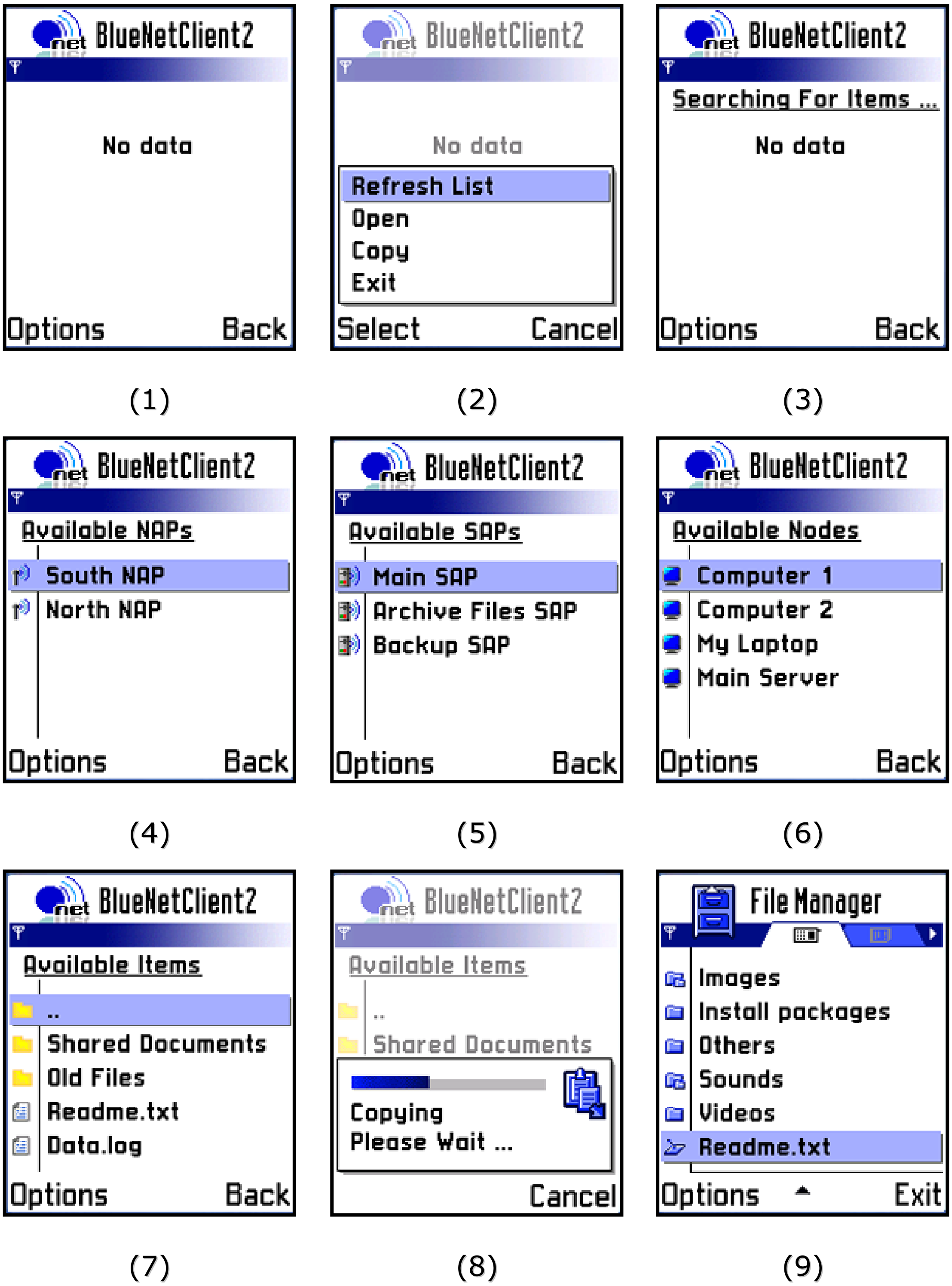


**Fig. 18** – Mobile Terminal Component GUI

In order to show the available NAPs in the neighborhood, the user selects **Refresh List** menu command**.** This triggers a scan operation (3) to search and find those NAPs, which are then displayed to the user (4). After selecting the desired NAP, the user selects the **Open** menu command from the options menu in order to list the available SAPs (5) that can be reached through the selected NAP (i.e. the SAPs that are registered with the selected NAP APR).

In order to access the service of a certain SAP, the user highlights the desired SAP and selects **Open** menu command again. The mobile phone component acts as a client to the service provided by the SAP. A different client is needed for each service type. In this implementation, the service that is developed as a demonstration is computer network shared files access; therefore, the client (**BlueNetClient2**) is used to browse and fetch the public contents shared by network computers. Once the user selects **Open** (as mentioned earlier), the available computers connected to the network (as seen by the selected SAP) are displayed (6). The same method is used for browsing the shared folders and files of a selected computer (7), by highlighting a folder and selecting the **Open** menu command. The user can go back to the parent folder by either using the **Back** soft key command or opening the "**..**" folder.

Finally, when the user finds the required file she/he can copy it to the mobile phone by highlighting the file and selecting **Copy** menu command. A progress bar will show up indicating the file is being transferred (8). The file will be stored in the mobile phone file system and can be found by launching the **File Manager** application (9) that is built into the mobile terminal as a part of the software package provided by the mobile phone manufacturer. In this example, the file transferred is the **Readme.txt** file shown at the end of the list (9).

## * CONCLUSIONS

- Mobile phones are resources limited compared to computers, but they are more available and accessible considering their cost, size and usage. An important feature in mobiles is that they combine mobility and connectivity. Summing all of this, a mobile phone can act as a lightweight personal terminal used to access remote services provided by more capable and resource rich devices like database servers, file servers … etc.
- Mobile phones support connectivity to WAN/MAN (through mobile data services like GPRS) and PAN (through infrared or Bluetooth) data networks but not to LAN because wireless LAN hardware and software standards are impractical for implementation in mobile phones. Bluetooth has built-in support to bridge a PAN and a LAN using BNEP protocol and by defining the role of NAP, although this approach's hardware (Bluetooth transmitter) can be implemented in mobile phones but the software (BNEP) is not suitable since it's designed originally for more capable devices (like PCs) communicating with each others using complex peer-to-peer model.
- By modifying the Bluetooth LAN-through-PAN scheme, its implementation becomes practical for many mobile phones. The modification, proposed by the paper, is to replace the BNEP with the RFCOMM (already supported by Bluetooth enabled mobile phones), promoting the NAP to act at the application layer and switching to client/server model instead of peer-to-peer. This simplifies and reduces resources needed for implementation by moving most of the burden from the mobile phone (the client) to the NAP (the gateway which can be considered as the "/" symbol) and the SAP (the server).

## REFERENCES

- [BNEP2003] Bluetooth Network Encapsulation Protocol V1.0 Specification, Bluetooth SIG, 2003.
- [GANGULI2002] Madhushree Ganguli, Getting Started with Bluetooth, Premier Press, 2002.
- [ITU2006] World Telecommunication/ICT Indicators, International Telecommunication Union, May 2006.
- [LEIGH2004] Leigh Edwards et al, Developing Series 60 Applications – a Guide for Symbian OS C++ Developers, Addison Wesley, 2004.
- [NOKIA2003] Symbian OS: Getting Started with C++ Application Development, Nokia Corporation, 2003.
- [NOKIA2004] Series 60 Developer Platform 1.0/2.0: Basics, Nokia Corporation, 2004.
- [PAN2003] Personal Area Networking Profile V1.0, Bluetooth SIG, 2003.
- [WINCH1998] Robert G. Winch, Telecommunication Transmission Systems, McGraw-Hill, 2nd Edition 1998.

**ABBREVIATIONS**

API: Application Programming Interface
BNEP: Bluetooth Network Encapsulation Protocol
GSA: Generic Service Access
GUI: Graphical User Interface
IEEE: Institute of Electrical and Electronics Engineers
IP: Internet Protocol
ISM: Industrial Scientific and Medical
L2CAP: Logical Link Control and Adaptation Protocol
LAN: Local Area Network
LMP: Link Manager Protocol
MAC: Medium Access Control
MAN: Metropolitan Area Network
MHz: Mega Hertz
NAP: Network Access Point
OBEX: Object Exchange Protocol
PAN: Personal Area Network
PDU: Protocol Data Unit
RFCOMM: Radio Frequency Communication
SAP: Service Access Point
SDP: Service Discovery Protocol
SIG: Special Interest Group
TCP: Transmission Control Protocol
UDP: User Datagram Protocol
VPN: Virtual Private Network
WAN: Wide Area Network
WLAN: Wireless LAN