UNIT I

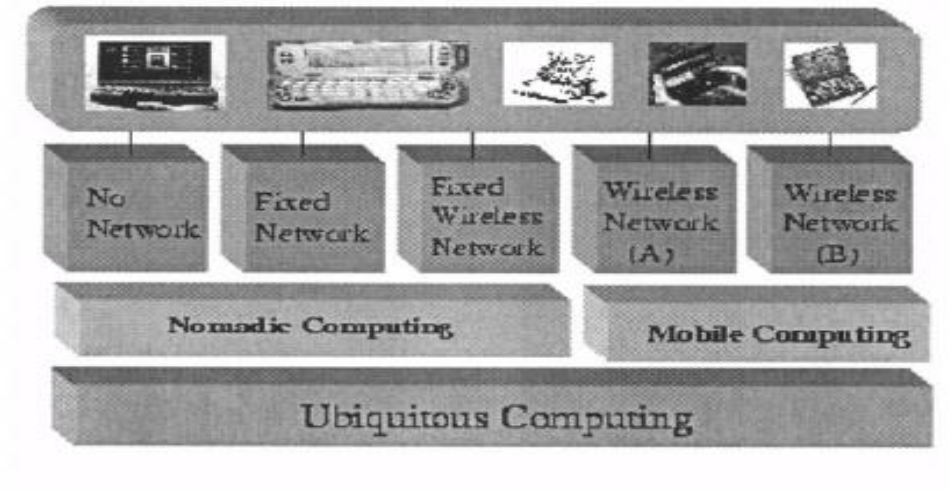
Introduction: Wireless and Mobile Computing Architecture – Limitations of wireless and mobile communication – Wireless Telecommunication Networks: Digital cellular Systems, TDMA -CDMA – Wireless Networking Techniques –Mobility Bandwidth Tradeoffs – Portable Information Appliances.

Introduction to Mobile Computing

- **Mobile computing** is human–computer interaction by which a computer is expected to be transported during normal usage. Mobile computing involves mobile communication, mobile hardware, and mobile software. Communication issues include ad-hoc and infrastructure networks as well as communication properties, protocols, data formats and concrete technologies. Hardware includes mobile devices or device components. Mobile software deals with the characteristics and requirements of mobile applications.

- Mobile computing will be the buzz of the next century. Buzzwords such as mobile, ubiquitous, nomadic, untethered, pervasive, and any time anywhere, are used by different people to refer to the new breed of computing that utilizes small portable devices and wireless communication networks.
- Defining and relating some of these buzzwords is an important prerequisite to this introduction. The difference between nomadic and mobile computing is particularly important to point out. Both nomadic and mobile computing require small portable devices. However, the kind of network used in nomadic computing does not allow mobility, or does so in the confines of a building, at pedestrian speed.
- Examples of such networks are DIAL-UP lines, which obviously do not allow any mobility, and Wireless Local Area Networks (WLAN), which allow for limited mobility within a building facility. Nomadic computing refers to the interleaved pattern of user relocation and "in-door" connection. Travelers carrying laptops with DIAL-UP modems are, therefore, nomadic users engaged in nomadic computing.
- Mobile computing, on the other hand, requires the availability of wireless networks that support "outdoor" mobility and handoff from one network to the next, at pedestrian or vehicular speeds. A bus traveler with a laptop connected to a GSM phone or a CDPD modem is a mobile user engaged in mobile computing.

- Figure 1.1 depicts this taxonomy. It also shows ubiquitous computing to be the aggregate ability to compute in both the nomadic and the mobile modes. Mark Weiser, a pioneer and a visionary from Xerox PARC, had different view and definition for ubiquitous computing.



:

Fig: 1.1 Ubiquitous=nomadic + mobile

- The reader is referred to his famous 1991 article in Scientific American [91]. We caution the reader that, in this book, the term mobile computing is used to refer to both nomadic and mobile computing, to reduce the clutter.

Impressive Technology

- An important question to ask is *which technology drove mobile computing to where it is today?* Is it the wireless network technology or the miniaturization and portable computing technology? Unfortunately, there is no easy answer.
- An individual with a Palm Pilot will probably answer in favor of the portable technology, whereas a UPS package delivery worker will be more thankful to the wireless technology. Whatever the right answer might be, more important questions need to be answered: *where are we now?* and *what are the challenges and impediments facing mobile computing?* This book attempts to answer these two questions by organizing a morass of information about technologies, standards, research, and commercial products.

Fig: 1.2 Emerging Portable information appliances

- Figure 1.2 shows how pervasive the portables technology has become. A large collection of portable devices are available in the market today. These are too many that we dedicated a chapter in this book to classify and describe each of them.
- Future portable devices that are currently in the prototype development phase or are just pure concepts.
- Similarly, the wireless communication technology is growing and expanding at a breath-taking pace. It is changing the way people live and interact. Subscribers are given the power of "ubiquitous" communication at affordable prices Antenna, power technology, and miniaturization breakthroughs have led to the small size design of radio equipment and the elimination of large tower and monopole infrastructures. It is now amazingly easy to deploy cellular networks (especially pico-cellular technology) in record times, and at an ever decreasing cost.
- There are several applications for mobile computing including wireless remote access by travelers and commuters, point of sale, stock trading, medical emergency care, law enforcement, package delivery, education, insurance industry, disaster recovery and management, trucking industry, intelligence and military.

- Most of these applications can be classified into: (1) wireless and mobile access to the Internet, (2) wireless and mobile access to private Intranets, and (3) wireless and ad-hocly mobile access between mobile computers.
- An example of a wireless and mobile access to the Internet is shown in Figure 1.5, where a traveler, through a Wireless Service Provider (WSP), is able to issue queries based on her location, direction of motion in a particular highway, and perhaps her vehicular speed.

Wireless and Mobile Computing Architecture

- The architectural model of a mobile computing environment is shown in Figure 1.6 and consists of stationary and mobile components. Fixed hosts are connected together via a fixed high-speed network (Mbps to Gbps).
- Some of the fixed hosts are special computers equipped with wireless interfaces, and are known as base (radio) stations (BS). They are also known as mobile support stations (MSS). Base stations, which are placed in the center of a cellular coverage areas, act as access points between the mobile computers and the fixed network. Mobile computers can be in one of three states.

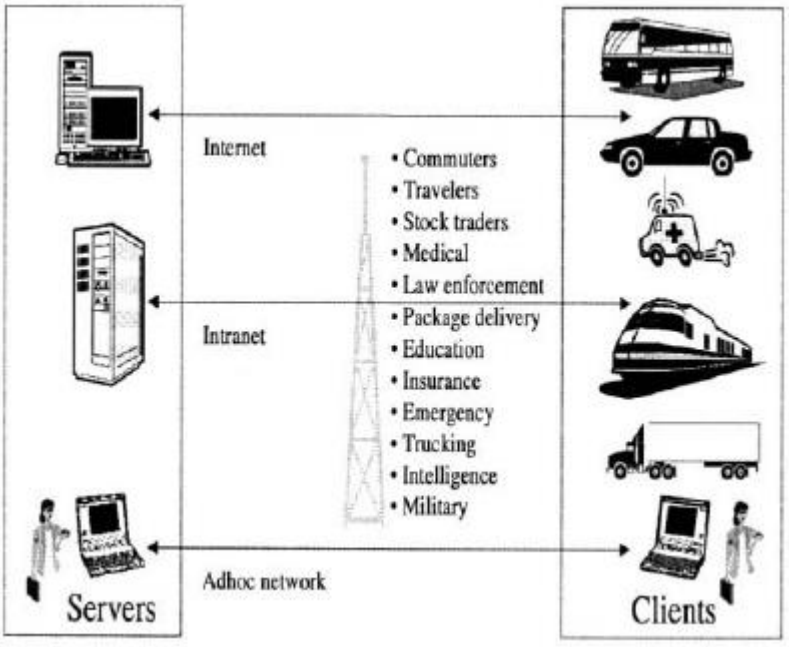
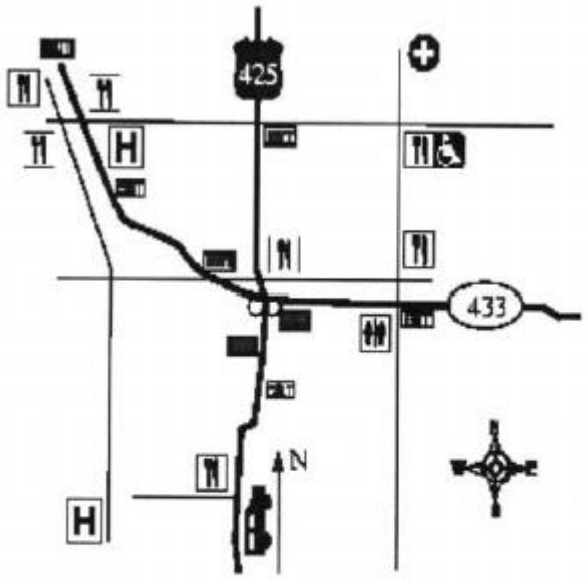


Fig 1.4: Beneficiaries of ubiquitous computing



- Nearest Japanese restaurant
- Nearest hospital w/ certain capabilities and availability
- travel info (nearest server station, hotel w/ pool, etc.)
- Pizza Hut nearest to destination

Fig 1.5: Location sensitive and continous queries

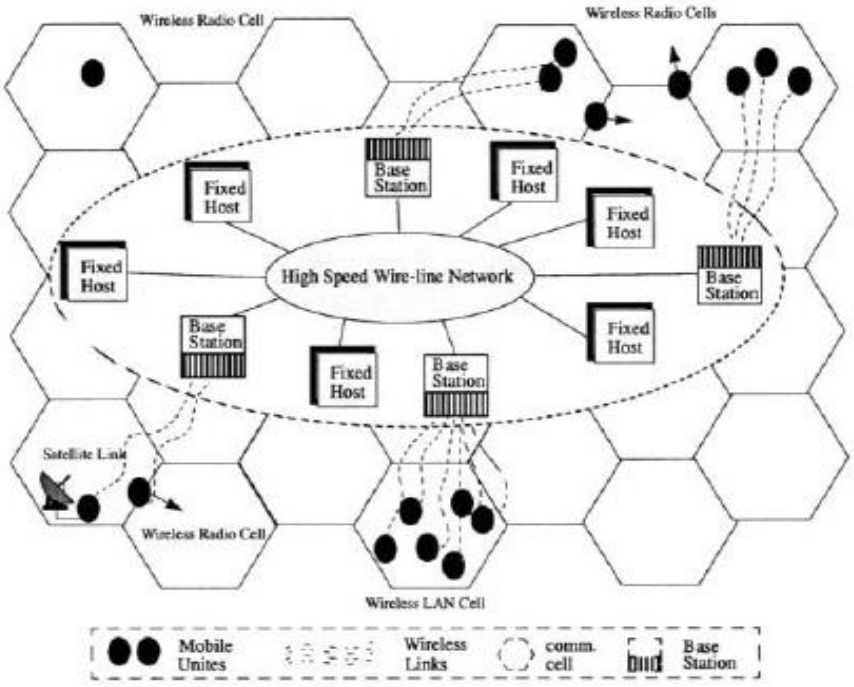


Fig 1.6: Mobile Computing Environment

- The first state places a mobile computer within a cell and capable of communicating. The second state places the mobile computer out of range of any service cell and not capable of communication. The third state places a mobile computer in a cell, communicating, but just ready to cross a cell boundary. These scenarios are depicted in Figure 1.6.
- Figure 1.6 is a generalized architectural overview of a typical wireless/nomadic system. Many such systems have been deployed both in the United States and Europe as well as in many other parts of the world.
- One such European system is the Global System for Mobile Communications (GSM). GSM, which is depicted in Figure 1.7, was originally developed by the European Institute for Research and Strategic Studies in Telecommunications (EURESCOM) as an advanced mobile communications technology.
- During early stages of deployment, GSM was hailed as a superior wireless technology because the general architecture supported such features as roaming, minimum disruption when crossing cell boundaries, and connectivity to any number of public wired infrastructures. Today, these features are common to most wireless infrastructures.

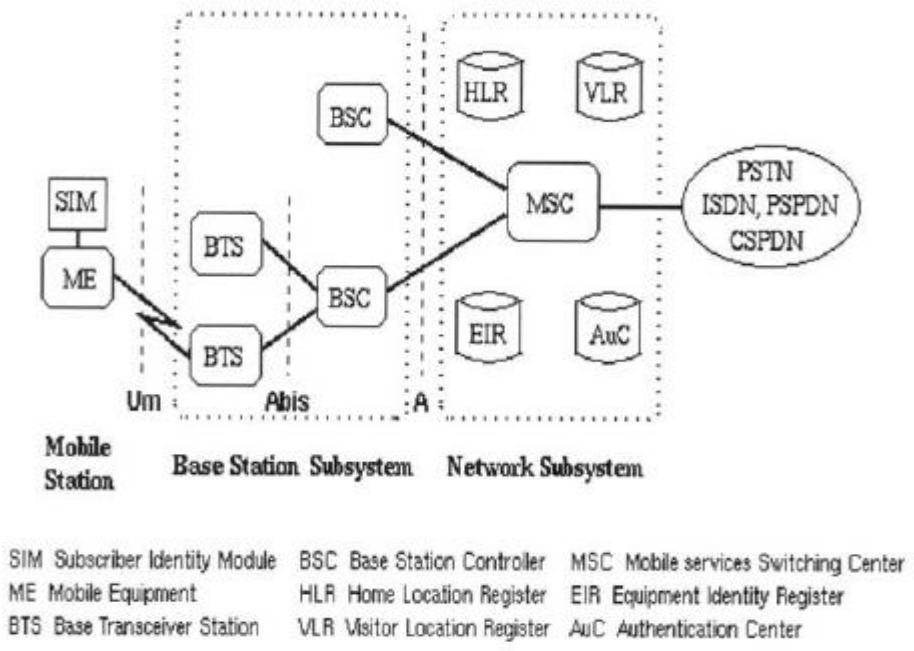


Fig: 1.7 GSM Architecture

- GSM is gaining increased popularity in North America.
- Figure 1.8 quantifies GSM penetration in terms of number of states with GSM services in the US. Although all of the wireless architecture's are unique in some respects, they all share many similar system components.
- The use of base stations for communication with the mobile computers, the centralized exchange systems which switch communications between the wireless domain and the wired infrastructure, and the use of location registers (HLR and VLR) so the system "knows" where the mobile computer is currently located and from where it came, are a few examples of the similarities of these systems.
- Another variable, which one might consider a service provider issue, is the incorporation of Advanced Intelligent Networking (AIN) . AIN was a joint effort between Bellcore (now Telcordia) and the RBOCs in the late 1980's, with standards completion around 1991. AIN deployment has been slow, but could play an important role in the realization of third generation wireless networks. Figure 1.9 depicts how the Personal Communications System (PCS) may be incorporated with AIN in overlay network architecture.

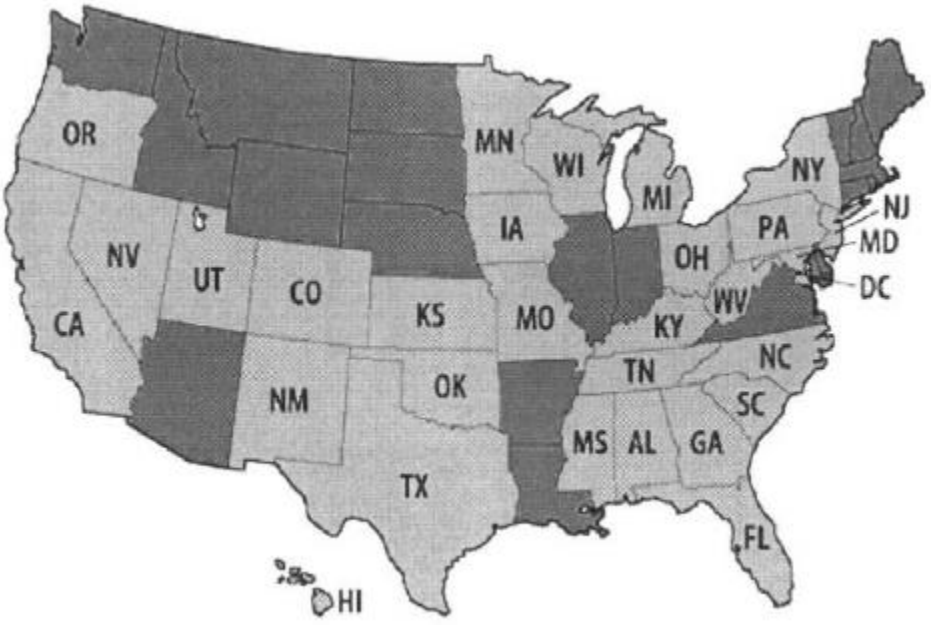


Fig 1.8: GSM Penetration in the US (shown in light gray)

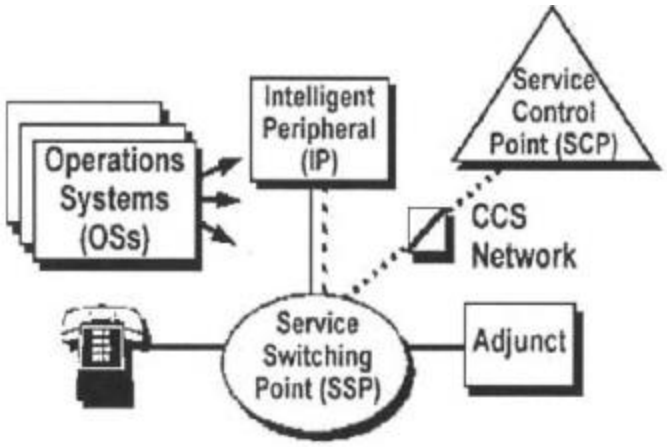


Fig 1.9: Advanced Intelligent Network Architecture (Telecordia)

- The Exact functionality and nature of the system components is left up to the service providers. Telcordia has only defined the high level attributes of the functional components and the interconnect between functional planes (layers) to aid them service providers in product selection and deployment.
- The PCS system with AIN services outlined in Fig is comparable to the overlay internetworking system.
- The PCS/AIN system shown above is comprised of many different forms of communications (eg: cellular, PCS, wired, POTS (Plain Old Telephone Service), etc.)With a centralized management scheme as defined by the Telcordia AIN standards. There exists interconnection across planes and between overlay planes to establish service attributes. One of the many issues to be addressed is how do wireless service providers and applications developers create, deploy, and control applications support services.

Limitations of Wireless and Mobile Communication

For the most part, the existing research and development on mobile computing is driven by the particularities and limitations of the mobile environment. Such limitations include:

Frequent disconnection caused by one of the following events:

- handoff blank out in cellular networks; the problem is worse in micro- cellular networks
- long down time of the mobile computer due to limited battery lifetime

- voluntary disconnection by the mobile user
- disconnection due to hostile events such as theft and destruction
- roaming-off outside the geographical coverage area of the wireless service

Limited communication bandwidth impacting the following:

- quality of service (QoS) and performance guarantees
- throughput and response time and their variances
- efficient use of battery due to long communication delays (wireless interface requires battery energy during the slow send and receive)

Heterogeneous and fragmented wireless network infrastructure leading to the following problems:

- rapid and large fluctuations in the network QoS
- Mobility transparent applications perform poorly without some sort of mobility middleware or proxy.
- Poor end-to-end performance of different transport protocols across networks of different parameters and transmission characteristics.

Other problems include:

- Security and anonymity
- Service relocation
- support for location-sensitive applications

There are other limitations related to platform and application development methodologies and languages. Operating systems for portable devices (other than laptops) are yet to reach maturity. Palm-OS Windows-CE, EPOCH, and GeOs are the most significant operating systems developed for mobile computing. A version of Linux for hand-held devices is also being developed. These operating systems are light weight with simplified, single-address space memory management.

Application portability across these operating systems is currently a major problem. The use of Java is currently limited due to the inadequate performance of JVM on most of these platforms. Development of mobile applications on these platforms is typically done through platform-specific SDKs supplied by the operating system vendors. Windows-CE development can also be done using Microsoft Visual C++. A unified and truly portable environment is most needed by application developers and inventors of future killer apps.

Wireless Telecommunication Networks

- **Wireless network** refers to any type of computer network that is not connected by cables of any kind. It is a method by which homes, telecommunications networks and enterprise (business) installations avoid the costly process of introducing cables into a building, or as a connection between various equipment locations. Wireless telecommunications networks are generally implemented and administered using radio communication. This implementation takes place at the physical level (layer) of the OSI model network structure.
- Today, person to person voice communications, enabled by the telephone, is still perhaps the most powerful technology available to the average person.
- The benefit to cost ratio of this technology for the individual is enormous. An individual can use a telephone to conduct commerce, earn a paycheck in countless ways, call for medical assistance, consult experts worldwide on any topic, and essentially obtain almost any critical information imaginable.
- The most sophisticated part of this technology is not in the telephone handset itself but in the enormous worldwide communications network to which the handset is attached. The introduction of cellular telephones has certainly improved the individuals ability to access (or be accessed by) this voice network in any location. But the global network is now providing more than person to person voice communications.
- Data, images, and live video are now routinely transferred to the individual desktop computer. It is expected that these expanding capabilities will soon be available within some type of portable information appliance.
- There are several well-established cellular infrastructures available today in different parts of the world .The European community has standardized largely on GSM. North America has broad AMPS coverage with a number of other standards competing in the PCS frequencies. Japan deployed the PHS infrastructure everywhere. A brief comparison of these predominant standards is shown in Table.

	PHS	AMPS	GSM
Usage area	Cordless (in-home) Mobile/ In-building (Japan)	Mobile/ In-building (N. America)	Mobile/ In-building (Europe) (N. Africa/Asia)
Applicable travel speed	slow driving	driving speed	driving speed
Voice signal	Digital	Analog	Digital
Frequency band	1.9 GHz	900 MHz	900 MHz
Channel multiplex number	4	1	8
Radio wave coverage	Indoor: 50–100m Outdoor: 100–400m	1.5–10km	1.5–10Km
Terminal to terminal communication	Possible	Not possible	Not possible
Data communication	32 kbps (plan)	14 kbps	9.6 kbps
Standby time	around 200 hrs	up to 20 hrs	up to 40 hrs
Talk time	5 hrs	up to 150 min	up to 240 min
Terminal output	less than 10 mW	600 mW	800 mW
Modulation method	Shifted QPSK	FM	GMSK
Voice transmission speed	32 kbps, ADPCM	Analog	22.8 kbps

Table: PHS, AMPS and GSM Wireless Technologies

Digital Cellular Systems

- Analog cellular systems such as North America's AMPS have the disadvantage that they are very expensive to expand and grow. Each mobile phone requires a dedicated channel to communicate in a cell site. The only way to expand in AMPS is to build additional cell sites which cost in the range of $500,000 to $1,000,000.
- In 1988, the Cellular Telecommunications Industry Association (CTIA) commissioned a subcommittee called Advanced Radio Technology to define alternative technologies that allows the cost effective cellular expansion in the US

- Proposed technologies focused on Multiple Access network technologies. The first digital system accepted by CTIA is the TDMA system, which stands for Time Division Multiple Access and which allows users to share the radio channel through time division. The second digital system accepted by CTIA is CDMA, which stands for Code Division Multiple Access, and which allows users to share the entire radio spectrum through different, uniquely assigned codes for transmission and reception.

Time-Division Multiple Access (TDMA)

- TDMA is a digital transmission technology that allows a number of users to access a single radio frequency channel without interference, by allocating unique time slots to each user within each channel.
- Currently, a single channel is divided into six time slots, with each signal using two slots. This provides a 3 to 1 gain in capacity of AMPS. In dispatch systems (e.g. Motorola iDEN), a dispatch signal uses one time slot, thus providing a 6 to 1 gain in capacity. D-AMPS, GSM, iDEN and several PCS systems currently use TDMA. The
- Telecommunications Industry Association (TIA) provided an early standard for TDMA over AMPS, known as IS-54, which required digitizing the voice signal, compressing it and transmitting it in regular series of bursts, interspersed with other users' conversations. Second generation standard for TDMA by TIA is the IS-136 which uses TDMA on the control channel.
- TDMA is expected to be called TIA / EIA-136 once it becomes an ANSI standard. One problem with TDMA is the wasted bandwidth of unused slots.
- Time slots are allocated to specific users whether or not they are using the slots (talking or transmitting data). Hughes Systems Network has contributed an enhancement of TDMA known as Enhanced TDMA (ETDMA) that attempts to correct this problem. Instead of waiting to determine whether a subscriber is transmitting, ETDMA assigns subscribers dynamically based on whether a user has voice/data to transmit. A phone conversation with long pauses will, therefore, not cause a loss of bandwidth, and will increase the spectral efficiency of TDMA.
- Today, TDMA is becoming a very popular air interface. Over 8 million digital subscribers worldwide utilize the IS-54 and IS-136 today. In the US alone, three of the top four carriers are deploying TDMA IS-136.

Code-Division Multiple Access (CDMA)

- In frequency and time division multiplex systems, several hundred channels are available within the spectrum allocation of a carrier service. One channel of one base station is used for each conversation. Upon handoff, the subscriber station is directed via messaging to discontinue use of the old channel and tune to the new one.
- Without reusing the frequency assigned in the spectrum, the total number of cells that can be deployed cannot exceed the available number of channels. Frequency reuse is very essential to the design of cellular systems that are based on frequency division multiplex.
- Frequency reuse utilizes the fact that the attenuation of electromagnetic fields tends to increase with distance. Therefore, to reuse the frequency without incurring significant interference, only non-adjacent cells are assigned the same frequencies.
- Ideally, cellular frequency reuse is achieved by imposing a hexagonal array of cells in a service area. A seven cells hexagonal array is shown in Figure. Seven frequency channels represented by different gray levels are used, one for each cell. The hexagonal array can be replicated and connected, providing a larger coverage area, without using any but the seven frequency channels. Systems that use frequency reuse includes AMPS in North America,NMT in Scandinavia, and TACS in the United Kingdom.
- In reality, cell coverage areas are highly irregular, and do not compare to the ideal hexagons shown in Figure. And even if ideal hexagons are possible, the frequency division approach offers limited capacity. Take AMPS as an example.
- Each AMPS operator in North America is allocated 416 channels (30KHz each).In a seven-way reuse hexagon, each cell will be allocated 416/7 = 59 channels.
- In this example, the capacity of cellular systems cannot grow beyond the bandwidth offered by 59 channels, which is 1.8MHz.

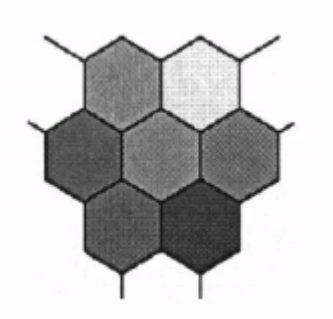


Fig: A Hexagonal array of seven cells using seven different channels

- Code-Division Multiple Access (CDMA) offers a solution to the capacity limitation problem. It allows all mobile stations to concurrently use the entire spectrum (all channels) with much less interference. Instead of partitioning either spectrum or time into disjoint “slots”, each subscriber is assigned a unique instance of a pseudo-noise digital signal.
- The transmission signal is “spread” over the entire spectrum, using the noise signal. CDMA is, therefore, known as a spread spectrum modulation scheme. The spreading technique is also known as Direct Sequence scheme.
- Frequency Hopping is another spreading technique, where the different segments of the subscriber conversation (or data) known as frames are transmitted on a sequence of randomly chosen frequencies within the spectrum. In either direct sequence or frequency hopping, the subscriber unit must communicate with the base station to agree on the direct sequence (the pseudo random digital code) or the sequence of frequencies to hop through.
- Signal interference in CDMA (between neighboring cells) is much less sensitive to most of the system parameters and is confined within a predictable average. This is one reason CDMA is attractive since it is easier to predict the achieved bandwidth based on the acceptable Noise to Signal Ratio (NSR) and the gain of signal spreading.
- Originally, CDMA was invented by Claude Shannon, who suggested that through noise-like carrier waves, bandwidth can be increased. Versions of CDMA has been in use for quite some time by the military for the different reason of security.
- Transmitted signal is difficult to decode by an intercepting party due to the spreading and the unknown spreading noise signal. It is known by the military to be a Low Probability of Intercept (LPI) and Low Probability of Detection (LPD) air interface scheme.
- Since late 1980s, CDMA has been migrating into civilian applications and is now reaching maturity and impressive market penetration. Future wireless networks known as third and fourth generation wireless networks (based on where you are in the globe) are mostly based on CDMA.

Wireless Networking Techniques

- Wireless technologies can be grouped into at least six major categories: (1) in room, point to point infrared, (2) in-room radio, (3) in-building radio frequency,(4) campus or metropolitan area packet networks, (5) wide-area packet/circuit switched data networks, and (6) regional-area Satellite Data Networks. These six classes of networks have unique technologies which constrain the nature of the applications which can be supported by each of them.
- A similar taxonomy is provided .Typically; an overlay of two or more network categories is used to provide continuous coverage in a mixed nomadic/mobile environment. Figure . Shows an overlay of several network technologies.

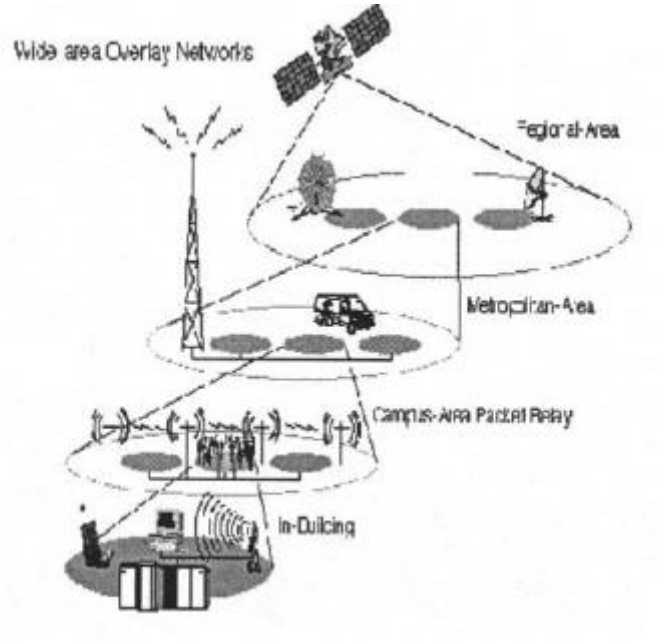


Fig: Wireless Network Overlay

In-room Infrared

- The in-room infrared class of networks generally has a network diameter of about 40–50m and supports bandwidths of about 1 Mbps. Applications supported by this type of infrastructure are limited to E-mail and collaborative work applications due to the limited range of the system. The Infrared Data Association (IrDA) provides the most common standard used today for this network technology.

In-room Radio Frequency

- The in-room radio frequency class of networks emerged in 1998 with the organized effort of the Bluetooth Special Interest Group. Bluetooth is a low-cost, short range radio that connects mobile PCs with other Bluetooth devices within a radius of about 10m. Very low energy consumption and about 1Mbps transmission speed makes this type of network attractive and suitable for inter-office device communication.
- Hospital intensive care units, bank tellers, and desktop component interconnect may be example applications that could utilize in-room RF wireless technologies. The proliferation of portable devices such as 3COM's Palm Pilot, Windows-CE hand-held computers, and highly portable and powerful laptops such as the IBM ThinkPads may incorporate Bluetooth transceivers to bridge the in-room wireless technology with fixed network infrastructures.
- The challenge laying ahead is to identify a suitable API for applications that will run a top this specific technology. Such API will allow for the design of"infrastructure literate" applications that can accommodate the user expected performance levels while maintaining consistency across the infrastructure.

In-building Radio Frequency

- This type of network, which is also known as Wireless LAN, expands the range of the infrared and the Bluetooth technologies by increasing the network diameter to about 200m.
- Unlike infrared and Bluetooth, in-building radio frequency is a cellular network, where mobile computers are allowed to roam within and across cells. Several standards are available today for this type of networks including the IEEE 802.11 and the OpenAir interface.
- Examples of Wireless LANs include Lucent/NCR WaveLAN and Proxim RangeLAN. Both ISA and PC Card interfaces are available with support for Windows and Linux. Proxim also provides additional support to a variety of Windows-CE devices.
- Wireless LANs can be used in both Infrastructure and Ad-Hoc Modes. In the former, Access Points are used and are connected to the fixed network through a dedicated router port.

- Wireless or nomadic devices with Wireless LAN interfaces access the network through the access point in the coverage area (cell). In this mode, the wireless LAN is used as a wireless extension of a fixed, high-speed network infrastructure (hence the name).
- In the ad-hoc mode, several portable devices with wireless LAN interfaces are placed in the transmission range of each others. Each device is capable of communicating with any other device directly, without the help of any networking infrastructure.
- A private network is used to configure the network software (TCP/IP) among the ad-hoc group of devices. Ad-hoc networks is becoming increasingly important technology. This technology, even though highly mature at this point in time, faces a few challenges.
- First, the IEEE 802.11 standard does not seem to be universally accepted (at least not yet). The OpenAir interface consortium, for instance, provides a competing proposal that is gaining popularity. Also, there is a lack of consensus on which air interface to use (direct sequence or the frequency hopping).
- Another challenge lies in the fact that wireless LANs are MAC-level networks that do not understand important features of IPv6 such as Multicast, RSVP, among other features. Unless, somehow, these features are implemented for wireless LANs, certain applications will be difficult to implement.

Campus/Metropolitan Area Packet Networks

- This network type encompasses the more traditional "cellular" networking paradigm. It is typified by a "pole top infrastructure" supporting network diameters of 0.2 to 5 miles with data rates of 20-128 kbps. Relay (or router) nodes are strategically placed to support the wider network diameter with a small price for increased latency.
- For example, typical latency between a mobile device and the first relay node is about 40ms (assuming an uncongested network), and about 20ms between relay nodes.

Wide-Area Packet/Circuit Switched Data Networks

- This network is comprised of a more familiar set of technologies and Regional Bell Operating Company (RBOC) services.
- One such offering is the Cellular Digital Packet Data (CDPD) service which is a packetized wireless transport that utilizes the unused channels of a cellular infrastructure. Motorola's ARDIS and iDEN systems, Ericsson's RAM (now called MobiTex), and the EuropeanGSM system are contained in this taxonomy.

- The iDEN network (Integrated Digital Enhanced Network) is a packet based voice/data network that uses the Mobile-IP networking protocol to route data packets.
- Not only is this technology capable of supporting larger diameter networks, but they also tend to have lower bandwidths and higher latency effects than do the in-building networks. This tends to present a unique set of problems in application development. Significant body of research on network and system adaptation through infrastructure awareness components has been or is being conducted.

Satellite Networks

- Satellite technology is still emerging. It is a downlink technology where mobile computers can only receive direct broadcast from a satellite. Outbound communication is initiated by the mobile computer through a modem DIAL-UP or other wireless technology.
- Hughes Network Systems pioneered the DirecPC network which uses the Galaxy satellite and which delivers 400 kbps downlink rate. DirecPC also transmits continuous streams of multimedia information ranging from CNN broadcasts, to news, sports, and financial news feeds.
- Other Low Earth Orbit (LEO) systems are in planning and deployment phases including the Internet in the Sky project.

Mobility Bandwidth Tradeoffs

- Another classification of the current wireless networking technology can be based on the "degree of mobility" offered by these networks. Multi-cellular wireless infrastructures range from in-building cells, to micro-cells (urban coverage), to macro-cells (suburban coverage), to satellite (global coverage).
- In building cellular offers the highest bandwidth (bi-directional), but very limited mobility. Micro-cellular offers lower bandwidth but allows for limited-speed mobility; macro-cellular offers much lower bandwidth but allows for the highest degrees of mobility. As can be noticed, in these networks, the larger the coverage area (the cell size), the higher the degree of mobility.
- Satellite networks are an exception and do not follow this trend.

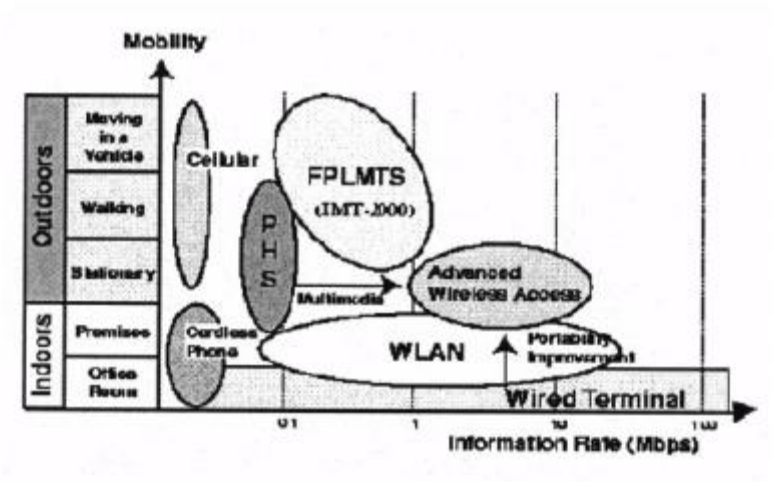


Fig: Mobility Bandwidth Tradeoffs

- They offer the highest downlink bandwidth (no uplink possible with satellite networks), but they do not offer any mobility. Instead, they require a satellite dish to be stationed aiming at the satellite.
- Figure shows a mapping of the mobility/bandwidth classification onto individual wireless networking technologies. In this mapping, mobility is further classified into indoor and outdoor, with outdoor mobility ranging from stationary, walking (pedestrian pace), and vehicular speed.
- The current mapping of wireless technology to the mobility/bandwidth classification is bound to change. At least this is ITU's and ETSI's vision and expectation of the third and fourth generation networks.
- For example, wireless LANs (an in-building technology) is expected to evolve into a network that allows for limited-speed mobility. Also, macro-cell networks are expected to improve on the bandwidth they offer.

SYSTEMS ISSUES

- The rapid expansion of wireless Wide Area Network (WAN) services, wireless Local Area Networks (LANs), satellite services such as Hughes' DirecPC and the planned Low Earth Orbit (LEO) systems have created a large and fragmented wireless infrastructure.

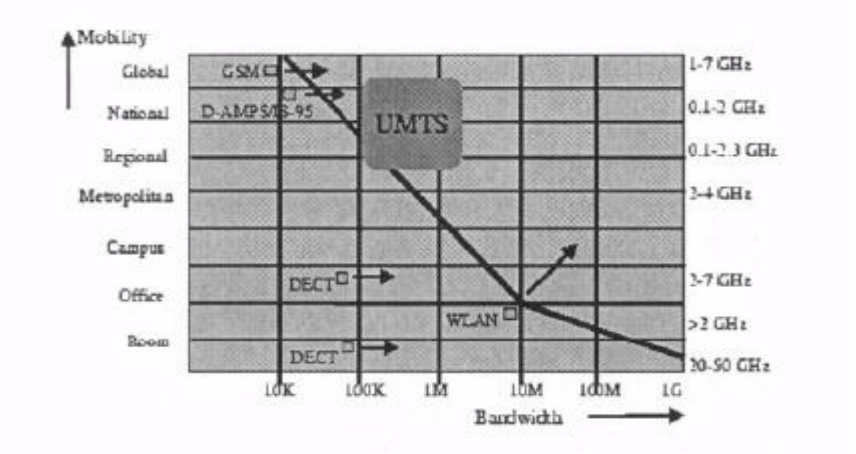


Fig: Expected mobility/Bandwidth tradeoffs in 3G and 4G Networks

- Given such a diverse set of technologies, the need to support mobile applications remains critical and even strategic to many industries.
- The ability to scale performance and latency while accommodating an increasing user density is of paramount importance when designing and/or selecting a wireless infrastructure for a particular application. The choice of a wireless infrastructure must take into consideration the attributes of the application and the applications class of service requirements including bandwidth, network latency, service coverage, and general performance issues. Table (a) summarizes application classes as stringently defined by ITU-T Recommendation I.211.
- These classifications have some loose definitions. For example, "interactive" usually means conversational, implying a person on either end of the application connection. The term "messaging" generally refers to a person talking to a machine.
- An example would include leaving voice mail or sending a FAX. The term" retrieval" is generally thought of as a machine transferring information to a person. Also, the term "distribution" is typically thought of as a machine sending to people or machines who listen passively.
- The Client/Server architecture is a primary example of this application class. Application updates may include human intervention, but could be automated. The last five application classes listed in Table (a) are considered machine-to-machine interactions, although they may have to be "user" activated, while the actual transaction is between machines.

Application Class	Example Applications
Interactive Video	Video Conferencing, Distance Learning, etc.
Interactive Audio	Telephone, Digitized Voice over the Internet
Interactive Text/Data	Transaction Management, Credit Verification
Interactive Image	Teleconferencing, Collaborative Workgroups
Video Messaging	Multimedia E-mail
Audio Messaging	Voice Mail
Text/Data Messaging	E-mail, Telex, FAX
Image Messaging	High-Resolution FAX
Video Distribution	Television, VOD, PPV
Audio Distribution	Radio, Audio Feed, etc.
Text Distribution	News Feed, Netnews
Image Distribution	Weather Satellite Pictures
Video Retrieval	VOD
Audio Retrieval	Audio Library
Text/Data Retrieval	File Transfer
Image Retrieval	Library Browsing
Aggregate LAN	LAN Interconnection or Emulation
Remote Terminal	Tele–commuting, Telnet
Remote Procedure Call	Distributed Simulation

Table (a) : Application classes with examples

Frequency band	2 GHz
Carrier bandwidth	5 MHz
Chip rate	5,115 Mcps
Frame length	10 ms
Voice	0.4-16 kbps
Video	128 kbps
Packet data	up to 128 kbps

Table (b): Wideband CDMA Standard

Multimedia Applications

- As of today, there are limitations which prevent the effective exploitation of wireless networks by portable information appliances beyond the area of voice communications, text messaging, and limited data. "While it is technically possible to transmit multimedia information such as a motion video clip from the intemet into a portable wireless device, the standards infrastructure bandwidth limitations, service costs, data compression technology, and power consumption considerations make this impractical at this time (1999).

- This limitations can be attributed to existing standards, which are limited in the level of service they Can provide to the user of a portable information appliance. While they are effective for voice and text messaging these standards do not support graphics intensive intemet browsing or real time video at a high enough speed to make them practical.
- In the case of video, the MPEG 1 standard provides for 352 X 240 pixel resolution, comparable to VCR quality video, and requires 1.14 Mbps data rate. This is well beyond any of the deployed wireless network standards. Today consumer expectation is set by MPEG 2 which supports high resolution video of 1920 X 1080 pixels, which requires up to 80 Mbps of bandwidth (typical applications of this standard, however, may only require 6 to 8 Mbps).
- To achieve wireless motion video data rates for portable devices, new wireless infrastructure standards will have to be deployed. One such standard is the Wideband CDMA approach proposed by Ericsson has the specifications shown in Table (b).
- This standard has been adopted by the European community for the next generation of cellular service and could be implemented globally by 2002. The motion video quality enabled by such a service would, however, be less than MPEG 1 in terms of resolution and/or frame rate.

Portable Information Appliances

- The first portable information appliance was probably a piece of stone or clay with markings on it, used to record numeric information. This information was probably very important to the user of this appliance and in some way directly affected his livelihood.
- Ease of use probably meant that the individual marks had to be deep enough in the appliance so as to be detectable by touching.
- Durability would have been important since the user did not have the means to protect the device from temperature variations, moisture, abrasion, and shock. To tile user, this device may have played a very important role in establishing his credibility, accountability, and responsibility with respect to the rest of his community.
- As the technology of mathematics and writing developed, human civilization progressed onward to the papyrus scroll (Figure) and ink pen. This appliance was highly portable and could convey very complex information.

Fig: (1) Papyrus; ancient Egypt's portable information appliance

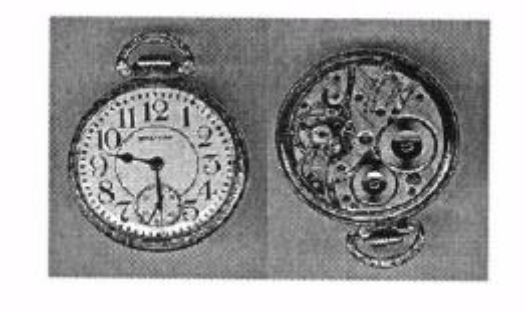


Fig: (2)Pocket Watch

- The user interface took a while to learn (reading and writing), and until relatively. recently, only a limited number of individuals were able to use the technology. Still pen and paper persisted for several thousand years and is still the preferred portable information technology for most of the world's population.
- Two other portable information appliances, the pocket watch and the printed book are relatively recent inventions which have trans- formed human society. The pocket watch enabled the level of logistical synchronization between individuals required for industrialization.
- Printed books, while not as interactive as paper and pencil, have also evolved as the preferred method for accessing standardized information in a portable format. Thus, paper and pencil, the printed book, and the pocket watch have been the dominant

portable information appliances since the dawn of the industrial revolution.

- In 1970 there were several bulky hand-held calculators on the market at price points of around $300 and above. By 1975, calculators had shrunk to pocket size and had fallen below the $20 price point. The age of portable electronic devices, enabled by the integrated circuit, was upon us.
- About this time, digital watches also began to replace mechanical watches which had been in place for hundreds of years.

Fig 3: An Early Calculator

Fig 4: Portable Video Cameras

- By the early 1980s portable video camcorders had sold over 1 million units worldwide and penetration of portable electronics to the consumer had begun in earnest. This rapid penetration was driven by the compelling application of acquiring and storing motion video images. This trend was further accelerated by the introduction of 8mm format models which were highly miniaturized.
- Cellular phones have seen remarkable penetration worldwide. By the late 1980s over 10 million units had been sold worldwide and the cell phone became a necessity for many and a status symbol for many others.

- By the early 1990s, over one million Notebook computers had been sold worldwide as

these products demonstrated their usefulness by turning spreadsheets and word processing into portable capabilities.

THE ADVENT OF THE PDA

- PDAs emerged in 1993 amid claims of single-point data organization, ubiquitous and instantaneous communications, and new operating paradigms using glitzy graphical user interfaces (GUI) and handwriting recognition. Most if not all of these claims fell short of consumer expectations.
- The reasons, while obvious in hindsight, lay hidden at the time. They were: high customer expectations, immature applications, and incompatible and unrealized infrastructures.
- Early on it was clear the success of the PDA rested heavily upon a variety of component and service infrastructures with the most critical of these enablers being wireless communications. In 1993, riding a sustained boom of 40% growth per year and giddy about recent cooperative initiatives, the cellular
- There were other problems as well with this initial surge of PDAs, but they served only to add to the mass confusion. The industry backlash, however, was both clear and severe. With hundreds of millions of dollars invested, two of the major players (AT&T EO, and IBM Simon) dropped out completely.

PALMTOP COMPUTERS

- It is likely that wireless network connectivity will trail wired connectivity in terms of performance for the foreseeable future.

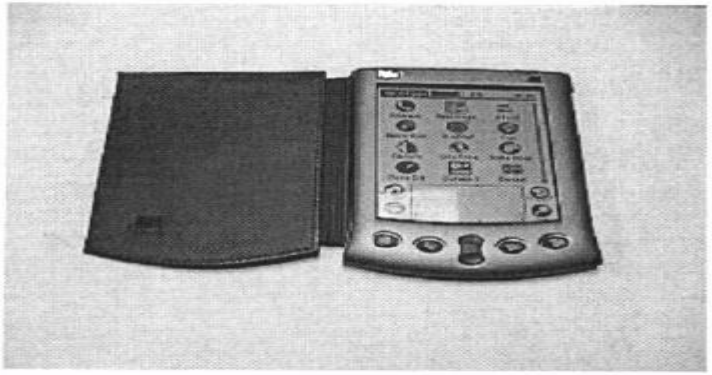


Fig:Palm Pilot V

- The best strategy for the developers of portable information appliance is to design products which either provide useful standalone functions such as an electronic still

camera, or which complement wired network platforms.

- The emerging market of Palmtop Computers is a breakthrough in terms of the ability of the Palmtop to complement the desktop computer.

The Palm Pilot

- The Pilot is a highly portable appliance which is the first truly viable substitute for traditional pencil and paper technology.
- With desktop synchronization, this device allows the desktop user to augment the networked desktop computing experience with a portable time management interface. While the Pilot is unlikely to provide services like high quality real-time video in the near future, this product concept has made important inroads into sensibly merging the interactions of portable and stationary information appliances.
- Many other contemporary product designers have failed to take this approach by attempting to combine and therefore replace other devices.
- One example would be a smart phone that combines the functions of a cellular phone and a notebook computer. Such product concepts often end-up compromising the features which make the individual products appealing. For instance, may smart phones have poor display quality, unusable keypads, poor battery life, poor performance, and are much bulkier than most cellular phones.
- The result is a product that does not effectively replace either of the products that it is competing with palm top

Item	Specifications
Size	4.7" x 3.2" x 0.4" (L x H x W)
Weight	4.0 oz. (including batteries)
Storage Capacity	2MB: 6000 addresses, 3000 appointments (approx. 5 years), 1500 to do items, 1500 memos, and 200 email messages.
Battery life	4-12 weeks (based on use) on 2 AAA batteries
Connectivity	RS-232C 9-Pin connector and 25-pin adapter; IR port; TCP/IP ready
Operating System	Palm OS
Applications	Date Book, Address Book, Mail, To Do List, Memo Pad, Expense, Calculator, Security, Games, HotSync, Others

Table: The Palm Pilot V Specification

Hand-held computers

The hand-held computer is another device that attempts to complement the desktop. It is

much more capable than a Palm Computer, larger in size and weight, but cannot be fitted in a pocket. Since their first emergence, hand-held computers have been competing with the Palm Computer market.

Sharp Power Zaurus

Item	Specifications
Processor	MIPS RISC Processor
Memory	16MB (ROM Upgradeable)
Display	6.5 High-Contrast Color LCD Touch Screen with Backlight (viewable area measured diagonally)
Colors	256
Resolution	640 x 240
Contrast control	Keyboard
Keyboard	64 Keys + 7 One Touch Application Keys
PC Card	one Type II slot
Audio	WAV file compatible with microphone, speaker, and external record button
Expansion Ports	Serial Port, PC Link, Printing
IR Port	IrDA 1.1 (115.2 kbps)compliant
Dimensions (w x d x h)	7.3 x 3.7 x 1.2 (186mm x 95mm x 29.6mm)
Weight	17.3 oz (490g)
Operating system	Windows CE

Table: Sharp Power Zaurus specification

The sharp Zaurus is a popular hand held computer that competes with a palm computer market

VADEM Clio

Clio is a Windows CE based hand-held PC with a swing-top design that provides three

modes of interaction: keyboard, pen and tablet, and presentation modes. The three modes are achieved by swinging and/or folding the display around the keyboard base. The specifications of the Clio, which is shown in Figure, are listed in Table

Fig: The VADEM Clio tablet hand-held PC

Item	Specifications
Processor	MIPS 4000
Storage	24MB ROM, 16MB RAM
Display	9.4" 640X480, 256 color
Operating System	Windows CE 2.1
Connectivity	IR port and built-in 33.6 kbps modem
I/O	keyboard, pen, and Type II PC card

Table VADEM Clio specification

Communicators

- The Communicator is a PDA concept that combines the benefits, portability and functionality of digital cellular phones and palmtop computers.
- The idea is to stick a palmtop computer to a cell phone with data capabilities to provide remote access in addition to the stand-alone form factor applications that can be found on palmtop computers.
- Internet access, telnet, email, and web browsing are all applications offered by communicators.

Item	Specifications
Memory	8MB total: 4MB OS and applications, 2MB program execution, 2MB user data storage
Processor	embedded INTEL 386 processor
Operating System	GeOS TM3.0
E-mail protocols	SMTP, IMAP4, POP3 and MIME1
Weight	397g
Dimensions	173 x 64 x 38 mm
Displays	Grayscale 640x200 (illuminated) LCD

Table: The Nokia 9000 Specifications

Nokia 9000

- The Nokia 9000 is the most popular communicator, not only because of its appearance in the hands of Agent 007 in one of his recent movies (1997), but because of the unprecedented unique features and capabilities.
- The Nokia 9000 combined a compact personal organizer with Internet access and a versatile voice and text messaging system. The organizer includes: an address book, note editor calendar with to-do list, calculator, and world clock. A built-in browser, Telnet, and a VT100 Terminal emulation are built-in applications that bring the Internet to the mobile user anywhere GSM coverage is available.
- A multi-protocol email client, Short Message System (SMS) and a Fax application are also bundled to provide a wide spectrum of communication alternative, of course, in addition to the digital voice phone interface.
- The picture to the right shows the communicator on a recharge base station and reveals the cell phone side of the device. The picture to the left shows an open communicator with a Web page on the backlit display.

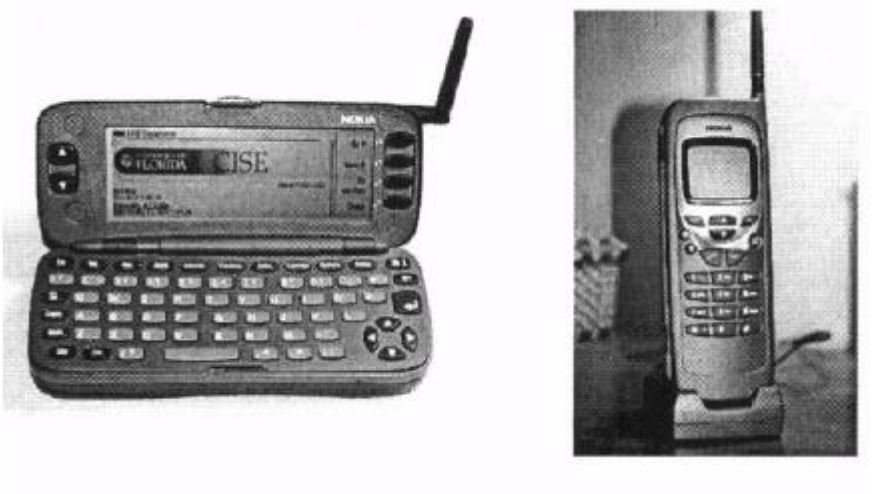


Figure Nokia 9000i Communicator

Motorola Marco

- The Marco wireless communicator was introduced to the market one year before the Nokia 9000 communicator (in 1995). It featured a built-in two-way wireless packet data modem allowing users to send and receive messages.
- The Marco Wireless Communicator, depicted in Figure ,also included a fax and data modem, allowing information to be communicated through any telephone network. To augment its functionality, the Marco was equipped with two PCMCIA Type II slots to allow users to simultaneously operate third party software applications and add memory to store more data.

Figure :Motorola Marco hand-held computer

SUB-NOTEBOOKS (MICRO-NOTEBOOKS)

As mobile users continue to demand lightweight, long battery life, and rugged portable computers, advances have been made in a number of diverse product concepts including what is now known as higher performance "micronotebooks", or sub-notebooks.

Figure A Sony sub-notebook

NOTEBOOKS

The notebook computer has enjoyed great success as the portable extension of the desktop computing environment. Notebooks are now starting to replace desktops for many users. Today the notebook market provides a most wanted portability by an increasing majority of users.

LAPTOPS

- Laptops are designed to replace the desktop. They can also be envisioned as nomadic desktops that can be easily moved from one place to another.
- The users of laptops require high performance, large high quality displays, and occasional portability. Such laptops may have maximum capabilities (as of 1999) such as up to 15.0in Color TFT (1024x768), integrated AC adapter, two battery support, up to 14GB disk storage, and 256MB memory. These capabilities come at the price of limited portability with these laptops weighing up to 8 Ibs.

Item	Specifications
Model Number	985TX
Processor	Intel Pentium 233MHz with MMX Technology
System Memory	32 Megabyte SDRAM (included), Up to 160 Meg
System Cache	512k L2 Cache
Bus Architecture	PCI/CardBus
Display	13.3in LCD TFT Active Matrix Color
Video Memory	4 Megabytes SGRAM EDO
Maximum Resolution	1024 x 768/16M
Video	MPEG I/MPEG II
Zoomed Video	3D Graphics
TV Out	1280 x 1024/256 (External Monitor)
Hard Drive	5GB EIDE (Formated Size)
Floppy Drive	1.44 Meg Removable

Figure Fujitsu Lifebook 900 laptop

OTHER INFORMATION APPLIANCES

- HP's Capshare 910 is a hand-held portable device that allows mobile users to capture, store, communicate and print documents, The 5.5L x 4.1H x 1.5W (inches) device shown in Figure weighs 12.5 oz and uses two AA NiMH rechargeable batteries that last for 100 document capture followed by a download.
- Typically, a mobile user capture documents from a newspaper or a magazine and then stores the document into his laptop or other portable device. Both PDF and TIFF formats are supported.
- The device has 4MB of memory and can capture from business cards and small receipts

up to legal-size documents or 25in. newspaper columns. Maximum capture area of 119 square inches

Future information Appliances

Wearable Computing (MIT)

- The systems include some of the following components: heads-up displays, unobtrusive input devices, personal wireless local area networks communication and context sensing tools. Interaction with the mobile computing system is based on the context of the situation.
- Applications and services offered might fall into the following categories: intelligent assistant, remembrance agent, augmented reality, or intellectual collectives.
- A video image of the camera's input is continuously projected into each eye. Two CRTs are driven by one video camera who's focal length is adjusted to avoid angular modification of the user's expected visual field.

Toshiba Desk Area Network (DAN)

- This type of network is a wireless desk area network, referred to as DAN. It allows one group or multiple groups of users to quickly form shared data (text or graphics) without needing the magic of a networking guru.
- The implementation of wireless DANs brings nomadic computing, the paperless office, and groupware one step closer to the business office. Toshiba DAN discovers neighboring devices and autonomously creates a self-organized network of terminals or laptops without a server. When machines enter or leave the area, it automatically and dynamically rebuilds an ad hoc network environment to reconstruct groupings and maintain communication routes

BlueTooth

- It is a technology specification for low-cost, short range radio links between mobile PCs, mobile phones, and other portable devices.
- The idea is to standardize a cheaper and shorter range version of existing RF technology. By doing so, it will be feasible and cost-effective to equip different portable devices with a wireless extension. The goal is to enable users to connect a wide range of computing and telecommunications devices easily and simply, without the need to buy, carry, or connect cables. BlueTooth can also be used to quickly

setup ad-hoc networks by allowing automatic, unconscious, connections between devices.

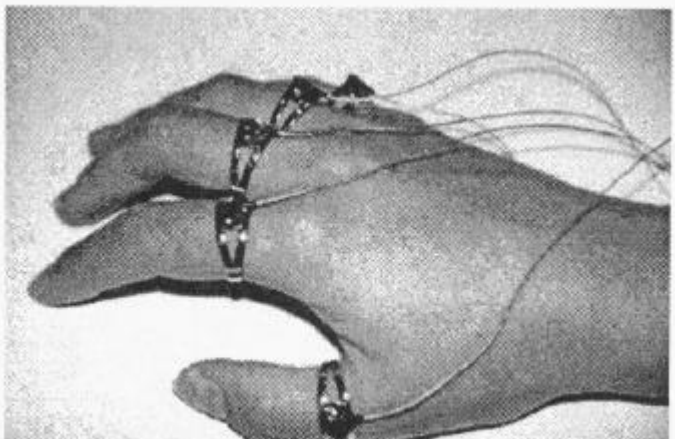
Seiko Wristwatch PC

- Seiko Instruments Inc. successfully commercialized the world's first wristwatch PC in Japan on June, 1998.
- The watch, which is called the Ruputer, is the cheapest wearable PC on the market today with models pricing in the range of $300. Hardware specifications include a 16-bit CPU, 128 KB of main memory, 512 KB of ROM, 512 KB of flash memory, 102x64 dots backlit display, 19,200 bps infrared port, 19,200 bps serial port, cursor pointer (only left and right movements), and four buttons.
- Its software includes several applications, some of which run under Microsoft's Windows-95 operating system such as Schedule, Address Book, Memo Book, Family Book, data entry editors, and viewers (text, images, and sound). The watch can also be used to play games (of course!). Both text and images can be downloaded to the watch from PCs. The watch can exchange data, via infrared signal, with other Ruputer watches, or can be connected to a computer or a laptop through a serial line.

Fig: Seiko Wristwatch PC

NTT Ring Keyboard

- Because the size of portable devices is most often limited by the keyboard and display, advances improving portability of these two components are herald events.
- One unique idea developed out of Nippon Telephone and Telegraph (NTT), Japans telecommunications giant, is the Ring-Keyboard. Shock-sensor rings allow a person to type on any surface and translates the different finger



Item	Specifications
Size	4.7" x 3.2" x 0.4" (L x H x W)
Weight	4.0 oz. (including batteries)
Storage Capacity	2MB: 6000 addresses, 3000 appointments (approx. 5 years), 1500 to do items, 1500 memos, and 200 email messages.
Batt	
Con	
Ope	
App	

volution since the

ingle new

ntroduced

flat panel

Item	Specifications
Size	4.7" x 3.2" x 0.4" (L x H x W)
Weight	4.0 oz. (including batteries)
Storage Capacity	2MB: 6000 addresses, 3000 appointments (approx. 5 years), 1500 to do items, 1500 memos, and 200 email messages.
Battery life	4-12 weeks (based on use) on 2 AAA batteries
Connectivity	RS-232C 9-Pin connector and 25-pin adapter;

ess Book, Mail, To Do List,
nse, Calculator, Security,
Others

UNIT II

Emerging Wireless Network Standards: 3 G Wireless Networks – State of Industry – Mobility support Software – End User Client Application – Mobility Middleware –Middleware for Application Development - Adaptation and Agents - Service Discovery Middleware – Finding Needed Services - Interoperability and Standardization.

Emerging Wireless Network Standards

- lTU (International Telecommunication Union) is a United Nation affiliated organization that oversees global telecommunication systems and standards. ETSI (European Telecommunications Standards Institute) is Europe's premier telecom standards organization well known for its development of the GSM standards. Both organizations are currently leading efforts to promote cooperation in the definition and development of future wireless networks.
- One goal common to both organizations is achieving seamless communication for the global consumer through cooperation on technical developments. Decisions made within these two organizations will have a dramatic effect on the future directions of wireless networks and services.

IMT-2000

- IMT 2000's vision is to "provide direction to the many related technological developments in the wireless industry to assist the convergence of these essentially competing wireless access technologies." IMT 2000 is expected to unify many different wireless systems, leading to the global offering of a wide range of portable services.
- It is expected that the IMT 2000 project will enable the merging of wireless services and Intemet services, leading to the creation of a mobile multimedia technology and new modes of communication.

IMT 2000 has the following goals:

- Incorporation of a variety of systems
- Achieve a high degree of commonality of design world wide
- Compatibility of services within IMT 2000 and with the fixed network
- High quality and integrity, comparable to the fixed network
- Accommodation of a variety of types of terminals including the pocket size terminal.

- The ability to use a small pocket terminal world wide
- Connection of mobile users to other mobile users or fixed users.
- Provision of services by more than one network in any coverage area
- Availability of a range of voice and data services to the mobile user
- Service portability-no difference between the services, transport capabilities, source coding, customer service or human-machine interface, regardless of where the call was placed.
- Provision of these services over a wide range of user densities and coverage areas
- Efficient use of the radio spectrum consistent with providing service at acceptable cost
- Provision of a framework for the continuing expansion of mobile network services and access to services and facilities of the fixed network.
- An open architecture which will permit easy introduction of advances in technology and of different applications
- A modular structure which will allow the system to start from as small and simple a configuration as possible and grow as needed, in size and complexity.

Third generation IMT 2000 network requirements include:

- Operation in a multi-cell environment (satellite, macro, micro and pico)
- Operation in a multi-operator environment
- Ear-wire line quality voice service
- Near-universal geographical coverage
- Low equipment cost, both subscriber stations and fixed plant
- Minimum number of fixed radio sites
- Seamless inter-frequency hand-off
- Mobile speed data rate of 144 kbps
- Portable speed data rate of 384 kbps
- In-building fixed wireless data rate of 2 Mbps
- BER (bit enol' rate) of ~ 10^{-3} (Voice), ~ 10^{-6} (Data)
- Creation of direct satellite access
- Transmission scheme suited for high speeds (e.g. fast running trains ~ 100 km/h)
- Flexibility for evolution from pre-third-generation and for post third generation systems

IMT 2000's vision of future wireless teleservices is specified in terms of information rate, user delay sensitivity, and bit enol' rate requirements.

UMTS

UMTS, which stands for Universal Mobile Telecommunications System, is currently a project under the SMG (Special Mobile Group), a committee in ETSI. Decisions made in early 1998 by ETSI has given Europe a clear direction towards the realization of its third generation wireless communication system. ETSI agreed in January 1998 on two different UTRA methods: W-CDMA in the paired portion of the radio spectrum, and TD-CDMA in the unpaired portion.

The main goals of the UMTS system can be summarized as follows:

Service Classification	Teleservices	Information Rate (kbps)	BER	Delay Sensitivity (ms)
Voice	Speech	8-32	10 -;$	40
	Emergency Call	8-32	10^{-3}	40
	Teleconference	32-128	10^{-3}	40
Voice Band Audio	Facsimile	32-64	10 -o	100
	Telefax	64	10^{-6}	100
	Modems	32-64	10^{-6}	200
	Data Terminals	2.4-64	10^{-6}	200
Sound	Program Sound	128	10^{-0}	200
	High Quality Audio	940	10^{-5}	200
Video	Conferencing	384-768	10	90
	Surveillance	64-768	10^{-7}	90
	Telephony	64-384	10^{-7}	40-90
Messaging	SMS & Paging	1.2-9.6	10 -o	100
	Voice Mail	8-32	10^{-4}	90
	Facsimile Mail	32-64	10^{-6}	90
	Video Mail	64	10^{-7}	90
	E-Mail	1.2-64	10^{-6}	100
Broadcast	Message	1.2-9.6	10 -o	100
	Multicast	1.2-9.6	10^{-6}	100
	SMS Cell	1.2-9.6	10^{-6}	100
	Public/Emergency Announcement (Voice)	8-32	10^{-4}	90
	Public/Emergency Announcement (Data)	1.2-9.6	10^{-6}	100
Data	Database Access	2.4-768	10 -o	200+
	Teleshopping	2.4-768	10^{-7}	90
	Newspapers	2.4-2000	10^{-6}	200
	CPS	64	10^{-6}	100
Teleaction	Remote Control	1.2-9.6	10^{-0}	100
	Remote Terminal	1.2-64	10^{-6}	100
	User Profile Editing	1.2-9.6	10^{-6}	200

Third-generation requirements for wireless teleservices

- The accommodation of high speed, multimedia interfaces to support Internet applications at speeds of up to 2 Mbps, through a quantum leap in technology
- At least a 3-fold increase in spectral efficiency
- Support from an evolved GSM core network

ACTS

Another organization that is greatly influencing the direction of wireless communications, particularly W-ATM are the projects funded out of ACTS (the Advanced Communications Technologies and Services). ACTS is a group of European research projects with budget 50% funded by the European Economic Commission (EEC). The remaining 50% of the research funding is provided by those industry organizations involved in the research. ACTS broad objective is to develop advanced communications systems and services for economic

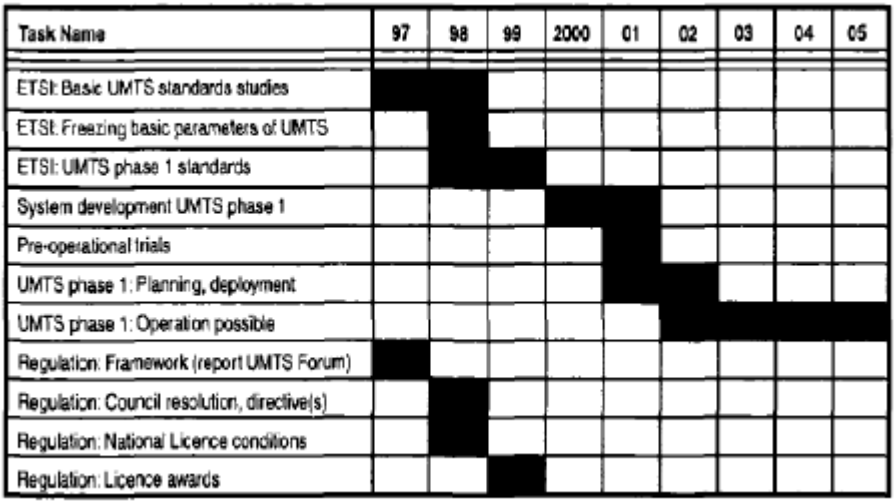


Figure UMTS schedule and milestones

3 G Wireless Networks

- IMT 2000's original vision for third generation wireless networks was to create a single global communication system common to all countries and regions. This vision was too revolutionary to second generation wireless network providers who have invested heavily in current technology.
- To protect their investments, carriers requested lTU to consider a more evolutionary approach to third generation network standards. ITU, in turn modified its vision into creating a "family of systems" that would converge and comply with a common set of

requirements for third generation networks.

- Following IMT 2000's vision, current research development, and global standardization efforts are focused on upgrading second generation systems including GSM, CDMA, and TDMA. A major goal of this conversion is to upgrade these system evolutionary over time while maintaining the operation and profitability of the existing second generation network infrastructure.
- TD-CDMA (Time Division, Code Division Multiple Access) and W-CDMA (Wideband Code Division Multiple Access) are the two major evolutionary network schemes currently under consideration by ITU. SDMA (Space Division Multiple Access) is also receiving attention as a network scheme with similar evolutionary nature.

Time Division/Code Division Multiple Access

- TD-CDMA is a proposed radio interface standard that uses CDMA signal spreading techniques to enhance the capacity offered by conventional TDMA system. Digitized voice and data would be transmitted on a 1.6 MHz wide channel using time-segmented TDMA technology. Each time slot of the TDMA channel would be individually coded using CDMA technology, thus supporting multiple users per time slot.
- One design goal of TD-CDMA is to allow the CDMA technology to be smoothly integrated into the existing, second generation GSM TDMA structure worldwide. This will allow GSM operators to compete for wide band multimedia services while protecting their current and future investments.
- An important feature of TD-CDMA is its ability to adjust the ratio of spectrum allocated for the uplink and the downlink. The air interface can therefore be tuned to enhance the performance of certain applications such as Internet access and voice applications.
- The compatibility between the TD-CDMA and GSM time bursts and frame structure permits the evolutionary step to third-generation systems. As many as 8 simultaneous CDMA codes are allowed in one time slot in TDCDMA. This permits 8 users per time slot, or a larger combination of voice and data users to communicate without interference.
- For example, TD-CDMA can accommodate 11 voice users and 5 data users and still maintains the appropriate BER (10^{-3} for voice and 10^{-6} for data). Eight users per time slot appears to be selected because it offers a happy medium between the number of voice and data calls that the system can accommodate.

- If all eight time slots were allocated to a single subscriber in a pico cell environment or where mobility is restricted, 1024 kbps can be achieved. Changing to a different data
- Assuming a bandwidth of 1.6 MHz, a time slot with an information rate of 16 kbps using QPSK data modulation, eight possible users per time slot (eight CDMA codes per time slot) gives you an information rate of 128 kbps. If all eight time slots were allocated to a single subscriber in a pico cell environment or where mobility is restricted, 1024 kbps can be achieved. Changing to a different data

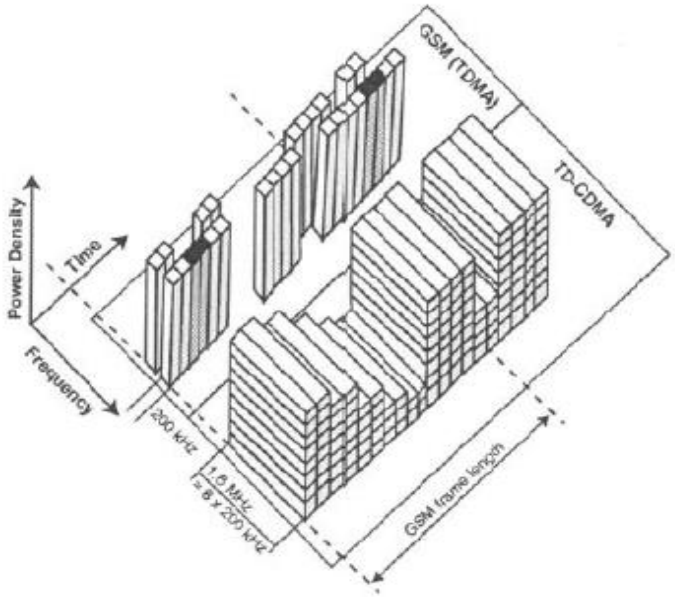


Fig: TD-CDMA VS GSM(TDMA)

- Another advantage of TD-CDMA is the fact that intra-cell interference is orthogonal by time. This enables multiple subscriber signals to be received at differing power levels thereby eliminating the near-far effect and the need for a soft hand-off. The hand-off is conducted through a separate TD-CDMA or GSM carrier simplifying dual mode, dual band handsets.

1. The networks that TD-CDMA is catered towards are significantly deployed infrastructures, including:

- GSM: deployed in 74 countries 200+ networks, and 20+ million subscribers, and
- AMPS: deployed in 110 countries 40+ million AMPS subscribers, 1.5+ million D-AMPS subscribers.

2. The TD-CDMA system is designed to be an evolutionary -not a revolutionary- step from GSM

second generation infrastructure to third generation infrastructure. The benefits to taking a revolutionary step include the absence of a legacy system and a quantum leap in abilities. The risk, however, is shortening the return on investment on second generation infrastructure But regardless of the philosophical underpinnings, keeping deployed infrastructure profitable is a concept well-embedded in the telecommunication industry.

3. TD-CDMA promises to be future proof:

- Spectral efficiency twice that of GSM
- Reuse of existing GSM network structure and principles: cell sites, Planning, hierarchical cell structures
- Efficient interworking with GSM
- Inherent TDD (time division duplex) support for cordless operation
- Data rate up to 2 Mbps indoor, 1 Mbit in all environments
- No soft hand-off and fast power control

Wideband Code Division Multiple Access

- W-CDMA is a spread spectrum technology in which the entire bandwidth is shared by multiple subscribers for transmission. A subscriber's data is modulated with PN codes· the signal is then spread and transmitted across a wideband.
- The receiver is responsible for despreading the desired signal from the wideband transmission and contending with interference. The despreading process at the receiver shrinks the spread signal back down to the original signal and at the same time decreases the power spectral density of the interference.

Bandwidth (MHz)	Pilot Channel Number	Sync Channel Number	Paging Channel Number
1.25	0	32	1-7 (sequential)
5.0	0 and 64	32 and/or 96	1-7 (sequential)
10.0	0 and 128	64 and/or 192	1-7 (sequential)
15.0	0	384	1-7 (sequential)

W-CDMA channel usage (48J)

W-CDMA's channel responsibilities can be described as follows:

- **Pilot channel (Forward link)**

The BTS (base transceiver station) transmits one or two pilot channels carrying a reference clock necessary for demodulation and the hand-off process. The pilot channel also carries information used in estimating BTS signal strength therein indicating the best communication link for the subscriber terminal. After deciding on the best pilot signal, the subscriber terminal demodulates the synchronization channel.

- **Synchronization channel (Forward link)**

  The synchronization channel contains system parameters, offset time, access parameters, channel lists and neighboring radio channel lists, necessary in synchronizing with the paging, access and voice channels. The synchronization channel always operates at 1200 bps.

- **Paging channel (Forward link)**

  System parameters and paging information to groups or a single subscriber are continually sent on the paging channel. Pages are combined into groups permitting a sleep mode to be built into the subscriber terminal extending battery life. Subscriber terminals can monitor multiple paging channels. Thus when another cell's paging channel has a better signal, a hand-off is requested. The paging channel has a data rate of 9600 bps or 4800 bps.

- **Access channel (Reverse link)**

  When a page is detected the terminal attempts to access the system through the access channel. The terminal increases signal strength sent to the BTS until the system responds, a random time limit has expired or maximum power levels have been exceeded.

- **Traffic channel (Both forward and reverse links)**

  Within the traffic channel, there are two types of in-band signaling used. Blank-and-burst in-band signaling where an entire 20 msec frame is replaced with control information. Dim-and-burst is also in-band signaling but the control information is distributed throughout a variable number of 20 msec frames.

The forward and reverse channels are modulated differently, QPSK for the forward channel, and O-QPSK for the reverse channel. There are five steps to the modulation process:

1. A PN multiplier multiplies the user data by the Walsh or Hadamard function, uniquely identifying that information to a specific subscriber terminal. The functions are time-shifted so that the set of functions are orthogonal

2. The output of the multiplier is a code rate of 4.096 Mcps using 5 MHz bandwidth (see Table

5.5) that is split into two signals: in phase (1) signal and quadrature (Q) signal

3. The pulse shapes for the I and Q signals are smoothed minimizing rapid signal transition that results in radio frequency emissions outside the allocated bandwidth
4. The balanced modulator multiplies the I and Q signals by two signals that are 90 degree phase-shifted. The number of bits per chip depends on the data rate supplied to the balanced modulator

5. The output of the balanced modulator is then fed to a RF (radio frequency) amplifier

The following are the features in the NTT DoCoMo/Ericsson W-CDMA experimental system:

1. Subscriber unit can receive multiple channels resulting in multimedia bandwidth. The NIT DoCoMo W -CDMA system can accommodate up to six 64 kbps channels simultaneously for a total bandwidth of 384 kbps per subscriber, enabling six different teleservices at the one time.

2. The system allows for future expansion with the aid of adaptive antennas. Adaptive antennas use SDMA techniques. SDMA manages interference and thus increases the network capacity, improves link quality, increases signal range, reduces transmission power, and extends the life and profitability of the deployed infrastructure.

3. New random access procedure with fast synchronization that provides flexibility in user data rates

4.Protocol structure that is similar to the GSM protocol structure

5. Inter-Frequency Hand-off (IFHO)

6. Hierarchical Cell Structure (HCS), permitting hand-offs between different wireless systems, (i.e. a hand-off between PHS infrastructure to the *W*CDMA infrastructure)

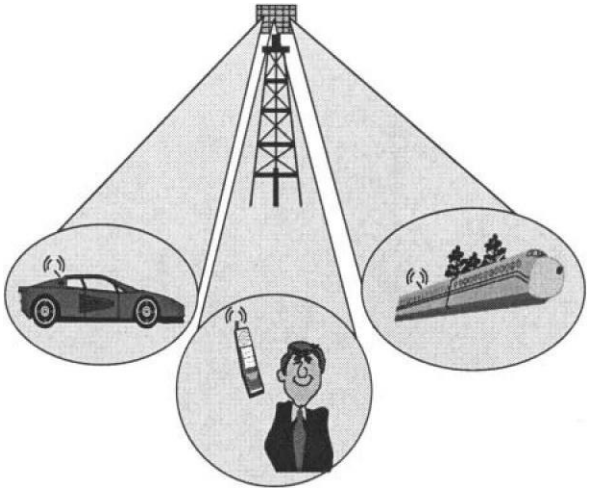
7. VOX - Voice activation silence suppression, does not send data when the audio level is below a threshold. VOX is also noted in the PHS ARIB standard as a low power consumption operation for the private system

8. Speech coding Orthogonal Variable Spreading Factor codes (OVSF). Utilization of a speech detection tool and orthogonal speech codes provides maximum bandwidth utilization in the W -CDMA environment. The speech detection tool, as explained earlier, assists in transmitting only the necessary data by transmitting less when speech

activity is low. The orthogonal speech codes prevent interference with other channels decreasing interference and increasing capacity

Space Division Multiple Access

SDMA is a technology which enhances the quality and coverage of wireless communication systems. It uses a technique wherein the subscriber's access is via a narrow focused radio beam and the location of the subscriber is tracked adaptively by an intelligent antenna array system



Space Division Multiple Access (SDMA)

- SDMA is derived from the physical spatial characteristics between the focused radio beams. Spatial processing is not a new concept; it is used in presently deployed cellular infrastructures. For example, some cell sites are sectored at 120 degree. Also, most base station sites use two antennas for diversity reception regardless of whether they are sectored or not.
- The most distinguishing aspect of SDMA is its management of interference. Reducing interference increases the effective network capacity, link quality and signal range. It also reduces the transmission power. Collectively, all the benefits brought by SDMA are expected to extend the life and profitability of second-generation network infrastructure.
- SDMA is applied to the TDMA and CDMA systems differently because of the systems' basic differences. TDMA co-cell subscribers are orthogonal by time. Increasing the capacity in a TDMA environment by employing SDMA techniques

requires multiple users on different radio beams to be assigned to the same carrier frequency and time slot.

- If the spatial component becomes insufficient between subscribers then an intra-sector hand-off is required to be initiated. The TDMA protocol needs to be expanded to permit these intrasector hand-offs.

Four types of interference concern cellular systems

1. Background noise
2. External interference
3. Other cell interference, and
4. Other user noise

All systems deal with interference types 1 and 2. Interference types 3 and 4 are dealt with differently depending on the type of access method. In a TDMA system, interference types 3 and 4 are orthogonal either by frequency or time and do improve with frequency reuse planning. An interference signal from a neighboring cell base station is orthogonal by frequency to the desired signal. Also an interference signal from a co-subscriber within the same cell is orthogonal by time to the desired signal. In a CDMA system, interference types 3 and 4 are spread across the same frequency and not necessarily orthogonal by time.

State of industry: Mobility support software

- End user applications such as Pocket Quicken, Farcast, Wyndmail, and Pact, that are narrowly focused on email, personal organizers, two-way paging and other similar services are not covered here. Instead, we focus on those end-user applications which involve remote access, client/server, and/or data synchronization.
- In those applications, we further focus on the top layers of the middleware communications software that provide support for mobile computing and data access.

Competing philosophies

The mobile client-application architectures which are emerging in commercial products can be roughly divided into three overlapping classes: Remote-Node Client Proxy, and Replication. The basis for this subdivision is the need to address the problems associated with wireless bandwidth and battery limitations and the alternatives that are commercially available today, for managing those problems.

- **Remote-Node:** This approach attempts to create a facsimile of a fixed network client node by hiding all artifacts introduced by wireless communications. Under this model all client software which run on a wired network platform would function without change on a mobile platform that includes a compatible OS and other library services. Accordingly, it places the most stringent demands on the middleware and other software (which supports the client application) to mediate the problems that arise as wireless artifacts. As a result, this approach is most susceptible to failures in the wireless infrastructure. Software packages which adopt this approach may recognize some of the wireless limitations and adapt their behavior accordingly. For example, when response time is of concern, the limited bandwidth of wireless communications encourages the system to deliver records one at a time as they are retrieved from a database server rather than sending all record hits for some query. However, the ultimate goal is to provide an opaque overlay for the underlying ensemble of networks that shields the user from any concern for their interoperability. Remote-node applications can be realized by porting full clients (as used in the wire line network) to a mobile computer with compatible communication middleware. Shiva PPP is a famous middleware that supports most TCP/IP clients.
- **Client Proxy***:* This approach, characterized by products like Oracle Mobile Agents, attempts to minimize transmission costs and the impact of disconnects by buffering a client's requests, and/or the servers responses and by resorting to batch transmissions. In this way, a user may select a variety of record types from several different tables, and then save battery power by disconnecting while the server processes the request. At some later time, the client can reconnect and receive a batch of records that satisfies all of the requests. The underlying assumption is that the end-user recognizes that periods of disconnect will occur, and that these periods will not impact the user's ability to perform useful work.
- **Replication:** Clients which will be disconnected for extended periods of time, but which require immediate access to important data can satisfy those requests from locally cached replicas of key subsets of the databases which are stored at some server site. Changes to the data that occur either at the client or the server must be reconciled through periodic client connects which may be initiated manually by the user, or automatically by the replication software. Some update conflicts may occur when multiple disconnected clients alter the same records. These collisions must be reconciled in some way.

End User Client Application

Recent literature search suggests that many of these products never materialized, were re-targeted to wired networks, or in some cases, are still struggling with weak sales. However, there are some big players with deep enough pockets to continue to pursue this marketplace.

Oracle Mobile Agents

- This product is a buffering and communications package for wireless platforms.
- A software agent that runs on the mobile client platform intercepts requests made by the client to the Oracle server and buffers them for a later transmission to the server. A companion Oracle agent runs on the Oracle server platform.
- That agent receives the buffered requests, submits them to the Oracle server, and buffers the responses for later transmission to the client. The server agent is capable of serving any number of mobile agents simultaneously. Conversely, a client agent can access any server agent that it knows about and for which it holds the appropriate DBA access privileges.
- Oracle agents can run on mobile platforms equipped with NT, Unix, or Windows and can communicate over TCP/IP using Shiva's PPP communications middleware. This product does not automatically support transactions or queries that span multiple Oracle servers.

Oracle Lite

- This product is a cut-down version of the Oracle server that can run in a small portable system (or a desktop workstation). It can be used as a companion technology for the Oracle Agent Software to store local copies of subsets of corporate databases and can accumulate updates to the data that are generated locally at the mobile client.
- Oracle may provide the Oracle Lite server with "two way" replication which could automatically propagate updates either to the client from the central server site, or vice versa.
- Recently, Oracle and Palm Computing (a 3Com company) announced an alliance to integrate the Oracle Lite client database and the 3Com Palm III and PalmPilot organizers, allowing new and existing Palm Computing platform applications and data to be replicated, synchronized, and shared with an Oracle 8 database server.

Oracle Software Manager

- This product is intended for a database administrator who needs to propagate software updates to remote copies of the Oracle server. It is capable of performing the distribution via hardwired networks or through wireless connections.
- It is not clear whether this package is versatile enough to accomplish a distributed software update to a collection of mobile devices as though the entire operation were a distributed transaction. For example, if the DBA needs to update the mobile Oracle-Lite server software for entire sales staff, the updates may have to be performed individually by the DBA.

Oracle Replication Manager

Oracle has announced a version of its Replication Manager which will eventually support bi-directional replication among a collection of distributed and centralized server databases. The Oracle approach is based on a peer-to-peer model, much like Lotus Notes, in which a collection of distributed processes manage replication collectively.

Sybase SQL Remote

- Unlike the Oracle Replication Manager, the Sybase product called SQL Remote has adopted a centralized model for managing replication. This product is a member of the Sybase SQL Any Where suite of tools (formerly called Watcom SQL).
- Also, Sybase has optimized its replication server to accommodate users that are only occasionally connected. So while this product has been developed with wired network users as a primary target, the software does include a component that recognizes the frequent disconnects that typify mobile users.

Mobility Middleware

- The majority of products targeted for the middleware market rely on TCP/IP and socket-like connections for the client server interface whether they are intended to be deployed in the wire line network arena or the wireless domain.
- Variants of TCP have been proposed to circumvent the problems that plague TCP for some wireless applications.
- By choosing to adopt this defacto standard transport protocol, vendors are positioning their products for deployment in a large existing infrastructure. As a result, it is already possible to surf the Internet using a Netscape interface on many wireless platforms and a simple cellular phone connection.

- ❖ Two key players in the wired-network middleware market that provide support for distributed users are Novell's Netware and Microsoft's Remote Access.
- ❖ Neither of these products will be discussed further since neither has yet announced plans (that we have seen) for moving into the wireless middleware domain.
- ❖ However, Microsoft Exchange has been integrated with Shiva's PPP software that allows communication of clients to servers through the cellular phone network.

MobileWare Office Server

- ❖ This suite of products was introduced in 1995 as a solution to managing mobile access to corporate data.
- ❖ The basic strategy that underlies MobileWare is to minimize mobile platform connect time by executing data transfers in a burst mode. The intent of this software is to make the mobile platform appear to the user as though it were actually a node connected into the wired network.
- ❖ The initial customer target focused on large sales staffs that were primarily mobile and who needed access on demand to sales support information that was too bulky and/or volatile to carry on extended trips.
- ❖ The current flagship product, MobileWare Office Server, includes a native Lotus Notes mail and database replication support.
- ❖ MobileWare Office Server is an agent-based middleware for wireless or wire line access to application data.
- ❖ Services supported by Mobile Ware Office Server includes Lotus Notes, Web browsing, e-mail and file transfer. A core component of the MobileWare Office Server is the Intelligent Transport Engine.

The transport engine provides several features including:

- Connection Profiles. The user chooses from a collection of profiles based on current working environment (LAN, Dial-up, or wireless connections and *TCP/IP*, NetBios, etc.). Each profile contains a set of tuned parameters that optimize the communication between the clients and the servers.
- Data check pointing. This ensures efficient recovery and fast reconnection after failures and involuntary disconnections.
- Automatic reconnection in response to involuntary lost connections.

- Data compression.
- Dynamic Packet-Scaling. Based on the current connection quality and capacity, data packets are dynamically re-sized to minimize connection time.
- Encryption and authentication. Uses DES encryption for per-connection authentication.
- Security. Forces re-authentication from the client upon receipt of any unregistered packet.
- Queuing. Application data is stored on both the client and the server in a client assigned outbox until a connection is made to transfer the data.
- Follow-Me Server. Uses a notification and delivery mechanism for events such as arrival of data to the client's outbox on the server. The user's mobile computer is notified if connected, and alternative notification procedures (such as paging) are allowed.

Shiva PPP

- ❖ Shiva's remote access client (known as PPP for Point-to-Point Protocol) enables mobile users to access servers embedded in either wire line or mobile servers almost seamlessly.
- ❖ For example, a client application that uses transaction processing services from BEN Tuxedo can now access those services from a mobile platform using PPP. This software suite provides some limited security features such as limiting the number of login tries, or disconnecting a session and calling the user back at a pre-established number. However, it does not provide the rich collection of services available from Mobile Ware's Intelligent Transport Engine

Middleware for Application Development: Adaptation and Agents

- ❖ Application development for mobile computers is a difficult task-on their own, applications are faced with a myriad of challenges: limited power and processing speed, varying levels of network connectivity, completely disconnected operation, and discovery of needed services.
- ❖ The goal of mobile middleware is to provide abstractions that reduce development effort, to offer programming paradigms that make developing powerful mobile applications easier, and to foster interoperability between applications. *Service discovery* or the art of dynamically discovering and advertising services.

- Two other important types of middleware for mobile computing—*adaptation* and *agents.*
- Recall that adaptation helps applications to deal intelligently with limited or fluctuating resource levels. The second type of middleware, mobile agents, provides a powerful and flexible paradigm for access to remote data and services.

Adaptation

- Mobile computers must execute user- and system-level applications subject to a variety of resource constraints that generally can be ignored in modern desktop environments.
- The most important of these constraints are power, volatile and nonvolatile memory, and network bandwidth, although other physical limitations such as screen resolution are also important. In order to provide users with a reasonable computing environment, which approaches the best that currently available resources will allow, applications and/or system software must adapt to limited or fluctuating resource levels.
- For example, given a sudden severe constraint on available bandwidth, a mobile audio application might stop delivering a high-bit-rate audio stream and substitute a lower-quality stream. The user is likely to object less to the lower quality delivery than to the significant dropouts and stuttering if the application attempted to continue delivering the high-quality stream.
- Similarly, a video application might adjust dynamically to fluctuations in bandwidth, switching from high-quality, high-frame-rate color video to black-and-white video to color still images to black-and-white still images as appropriate.
- A third example is a mobile videogame application adjusting to decreased battery levels by modifying resolution or disabling three-dimensional (3D) features to conserve power.

The spectrum of adaptation

- At one end of the spectrum, adaptation may be entirely the responsibility of the mobile computer's operating system (OS); that is, the software for handling adaptation essentially is tucked under the OS hood, invisible to applications.
- At the other end, adaptation may be entirely the responsibility of individual applications; that is, each application must address all the issues of detecting and dealing with varying resource levels.
- Between these extremes, a number of *application-aware strategies* are possible, where the OS and individual application each share some of the burden of adaptation. While

applications are involved in adaptation decisions, the middleware and/or OS provides support for resource monitoring and other low-level adaptation functions.

- The spectrum of adaptation is depicted in Fig.. In this part of the chapter, we are concerned primarily with middleware for adaptation, that is, software interfaces that allow applications to take part in the adaptation process.

Resource monitoring

- All adaptation strategies must measure available resources so that adaptation policies can be carried out. For some types of resources-cash, for example—monitoring is not so difficult. The user simply sets limits and appropriate accounts. For others, more elaborate approaches are required.
- The Advanced Configuration and Power Interface (ACPI) provides developers with a standardized interface to power-level information on modern devices equipped with "smart" batteries. Accurately measuring network bandwidth over multihop networks is more difficult.

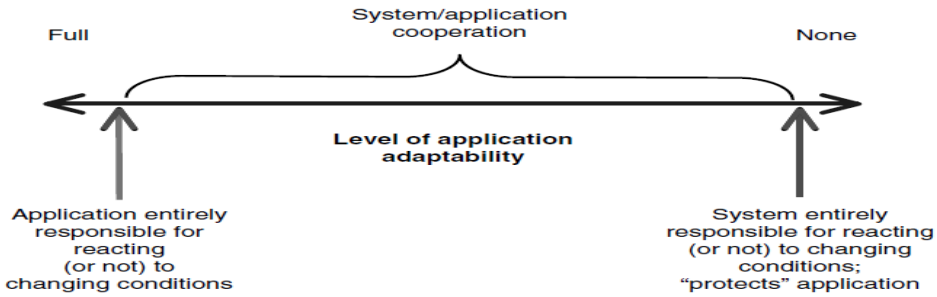


Figure: At one end of the spectrum of adaptation, applications are entirely responsible for reacting to changing resource levels. At the other end of the spectrum, the operating system reacts to changing resource levels without the interaction of individual applications.

Characterizing adaptation strategies

- *Fidelity* measures the degree to which a data item available to an application matches a reference copy. The reference copy for a data item is considered the exemplar, the ideal for that data item—essentially, the version of the data that a mobile computer would prefer given no resource constraints. Fidelity spans many dimensions, including perceived quality and consistency.
- For example, a server might store a 30- frame-per-second (fps), 24-bit color depth video at 1600 × 1200 resolution in its original form as shot by a digital video camera. This reference copy of the video is considered to have 100 percent fidelity. Owing to resource

constraints such as limited network bandwidth, a mobile host may have to settle for a version of this video that is substantially reduced in quality (assigned a lower fidelity measure, perhaps 50 percent) or even for a sequence of individual black-and-white still frames (with a fidelity measure of 1 percent).

- If the video file on the server is replaced periodically with a newer version and a mobile host experiences complete disconnection, then an older, cached version of the video may be supplied to an application by adaptation middleware. Even if this cached version is of the same visual quality as the current, up-to-date copy, its fidelity may be considered lower because it is not the most recent copy (i.e., it is *stale*).

An application-aware adaptation architecture: Odyssey

- In the spectrum of adaptation, Odyssey sits in the middle—applications are *assisted* by the Odyssey middleware in making decisions concerning fidelity levels.
- Odyssey provides a good model for understanding the issues in application-aware adaptation because the high-level architecture is clean, and the components for supporting adaptation are clearly delineated.
- The Odyssey architecture consists of several high-level components: the *interceptor*, which resides in the OS kernel, the *viceroy*, and one or more *wardens*. These are depicted in Fig.. The version of Odyssey described in Nobel and colleagues (1997) runs under NetBSD; more recent versions also support Linux and FreeBSD. To minimize changes to the OS kernel, Odyssey is implemented using the Virtual File System (VFS) interface, which is described in great detail for kernel hacker types

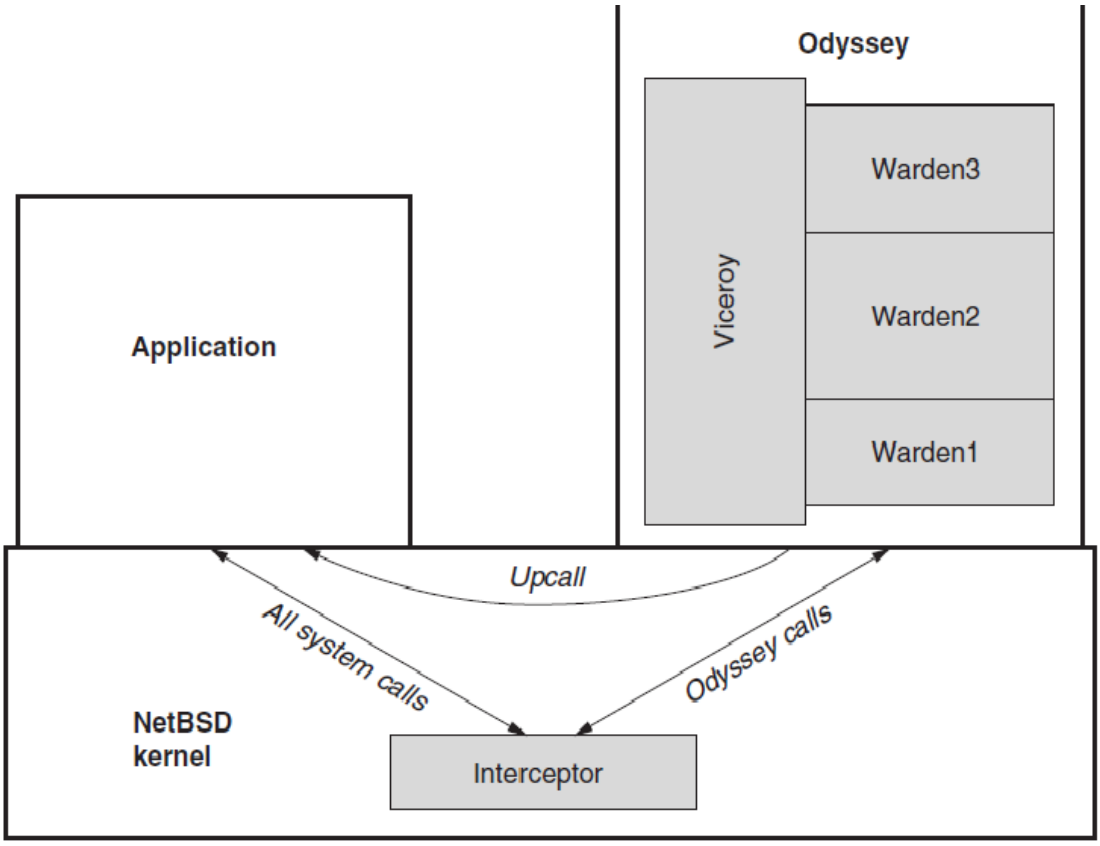


Fig:The Odyssey architecture consists of a type-independent viceroy and a number of type-specific wardens. Applications register windows of acceptable resource levels for particular types of data streams and receive notifications is when current resource levels fall outside the windows.

- Odyssey using (mostly) file system calls, and the interceptor, which resides in the kernel, performs redirection of Odyssey-specific system calls to the other Odyssey components.

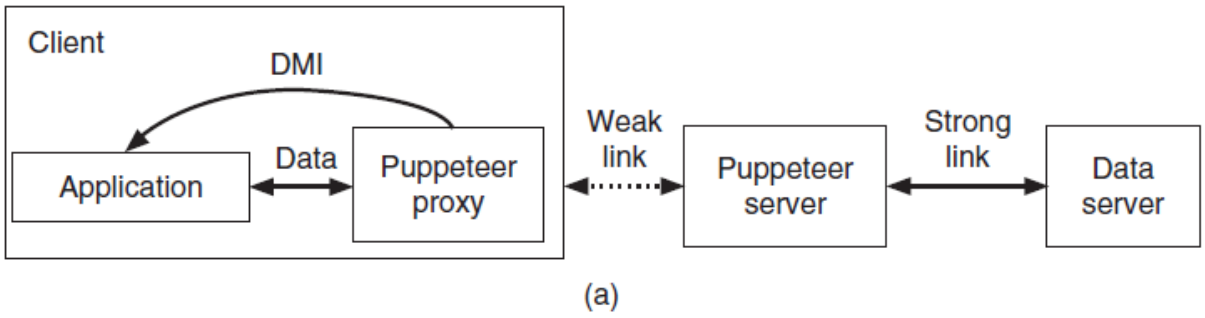
Wardens. A *warden* is a type-specific component responsible for handling null adaptation-related operations for a particular sort of data stream (e.g., a source of digital images, audio, or video).

Viceroy. In Odyssey, the viceroy is a type-independent component that is responsible for global resource control. All the wardens are statically compiled with the viceroy.

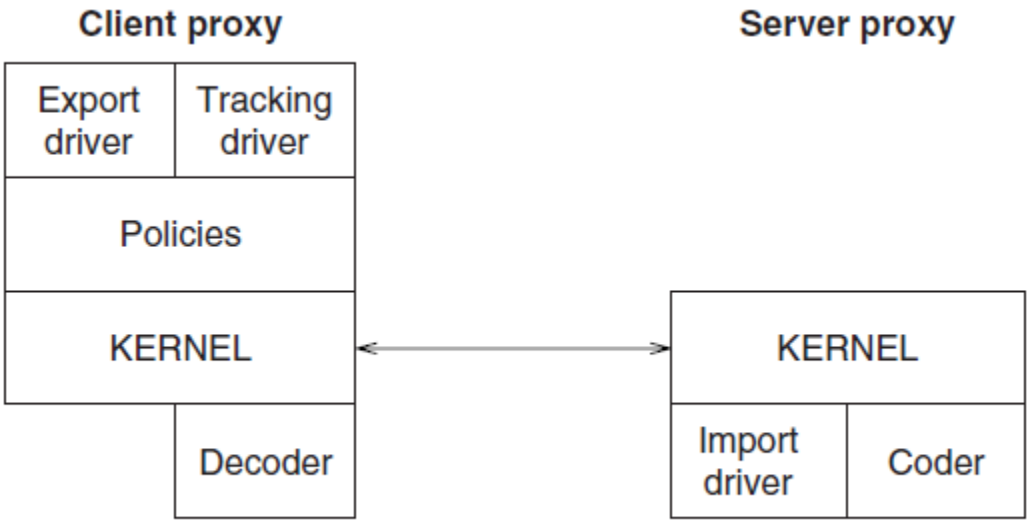
More adaptation middleware

Puppeteer. For applications with well-defined, published interfaces, it is possible to provide adaptation support without modifying the applications directly. The Puppeteer architecture allows component-based applications with published interfaces to be adapted to environments with poor network bandwidth. A typical application adaptation under Puppeteer is a retrofit of

Microsoft PowerPoint to support incremental loading of slides from a large presentation or support for progressive JPEG format to speed image loading.



(a)



(b)

The kernel also handles all communication between client and server sides. To adapt a document, the server and client side proxies communicate to establish a high-level PIF skeleton of the document. Adaptation policies control which portions of the document will be transferred and which fidelities will be chosen for the transmitted portions. For example, for a Microsoft PowerPoint document, selected slides may transferred, with images rendered at a lower fidelity than in the original presentation. The *import driver* and *export driver* parse native document format to PIF and PIF to native document format, respectively.

When the user opens a document, the Puppeteer kernel instantiates an appropriate import driver on the server side.

A typical Puppeteer adapted application operates as follows

_ The import driver parses the native document format and creates a PIF format document. The skeleton of the PIF is transmitted by the kernel to the client-side proxy.

_ On the client side, policies available to the client-side proxy result in requests to transfer selected portions of the PIF (at selected fidelities) from the server side. These items are rendered by the export driver into native format and supplied to the application through its well known interface.

_ At this point, the user regains control of the application. If specified by the policy, additional portions of the requested document can be transferred by Puppeteer in the background and supplied to the application as they arrive.

Coordinating adaptation for multiple mobile applications.

- When multiple applications are competing for shared resources, individual applications may make decisions that are suboptimal. At least three issues are introduced when multiple applications attempt to adapt to limited resources—*conflicting adaptation, suboptimal system operation*, and *suboptimal user experience*.

Mobile Agents

- Almost all computer users have used mobile code, whether they realize it or not—modern browsers support Javascript, Java applets, and other executable content, and simply viewing Web pages results in execution of the associated mobile code.
- For example, if a mobile user needs to search a set of databases, a traditional CS approach may perform remote procedure calls against the database servers.
- On the other hand, a mobile agents approach would dispatch one or more applications (agents) either directly to the database servers or to machines close to the servers. The agents then perform queries against the database servers, sifting the results to formulate a suitable solution to the mobile user's problem. Finally, the mobile agents return home and deliver the results.

Why mobile agents? And why not?

- The limitations of a single client computer are reduced. Rather than being constrained by resource limitations such as local processor power, storage space, and particularly network bandwidth, applications can send agents "into the field" to gather data and perform computations using the resources of larger, well-connected servers.

_ *The ability to customize applications easily is greatly improved.* Unlike traditional CS applications, servers in an agent system merely provide an execution *environment* for agents rather than running customized server applications. Agents can be freely customized (within the bounds of security restrictions imposed by servers) as the user's needs evolve.

_ *Flexible, disconnected operation is supported.* Once dispatched, a mobile agent is largely independent of its "home" computer. It can perform tasks in the field and return home when connectivity to the home computer is restored. Survivability is enhanced in this way, especially when the home computer is a resource-constrained device such as a PDA. With a traditional CS architecture, loss of power on a PDA might result in an abnormal termination of a user's application.

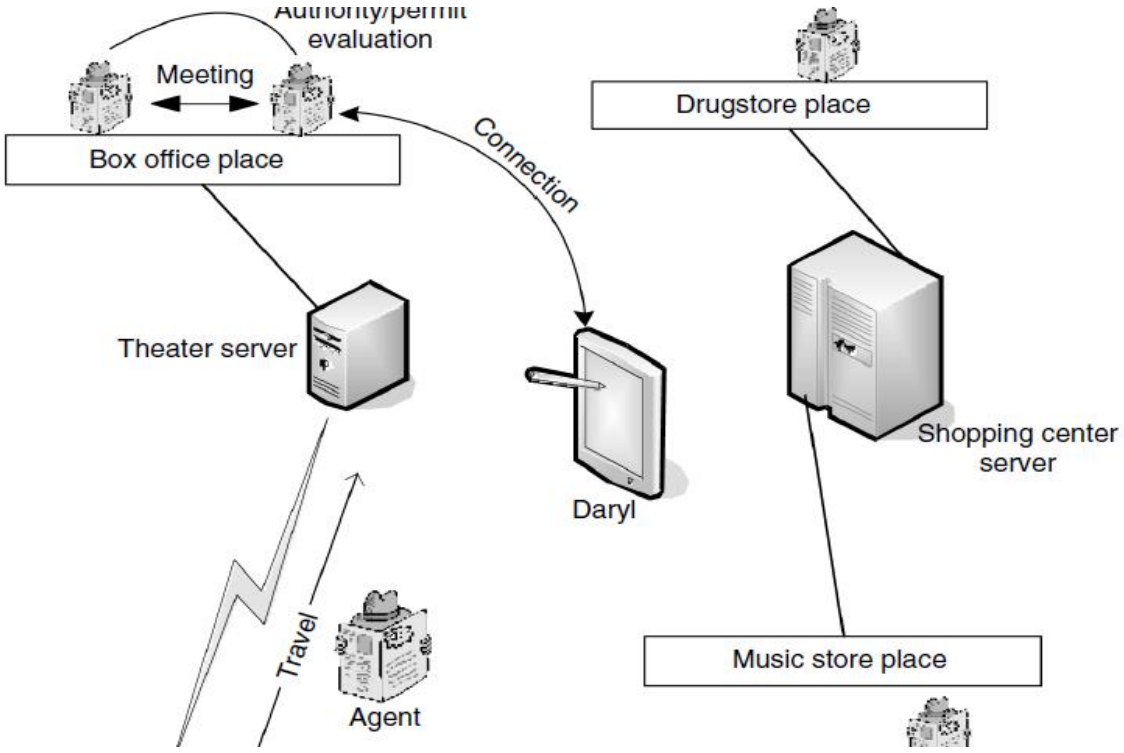
Agent architectures

There are a number of important components in the Telescript architecture: agents, places, travel, meetings, connections, authorities, and permits.

Places: In a mobile agent system, a network is composed of a set of places—each place is a location in the network where agents may visit. Each place is hosted by a server (or perhaps a user's personal computer) and provides appropriate infrastructure to support a mobile agent migrating to and from that location. Servers in a network that do not offer a "place" generally will not be visit able by agents. Places offer agents a resting spot in which they can access resources local to that place through a stationary agent that "lives" there, interacting with other agents currently visiting that place.

Travel: Travel allows agents to move closer to with needed resources. For example, an agent dispatched by a user to obtain tickets to a jazz concert and reservations at one of several restaurants (depending on availability) might travel from its home place to the place hosted by the jazz club's box office before traveling to the places hosted by the restaurants.

Meetings: Meetings are local interactions between two or more agents in the same *place.* In Telescript, this means that the agents can invoke each other's procedures. The agent in search of jazz tickets and a restaurant reservation (discussed under "Travel" above) would engage in meetings with appropriate agents at the ticket office and at the restaurant's reservation office to perform its duty.



Connections: Connections allow agents at different places to communicate and allow agents to communicate with human users or other applications over a network. An agent in search of jazz tickets, for example, might contact the human who dispatched it to indicate that an additional show has been added, although the desired show was sold out (e.g., "Is the 11 P.M. show OK?"). Connections in Telescript require an agent to identify the name and location of the remote agent, along with some other information, such as required quality of service. This remote communication method, which tightly binds two communicating agents (since both name and location are required for communication), is the most restrictive of the mechanisms

Authorities: An agent's or place's authority is the person or organization(in the real world) that it represents. In Telescript, agents may not withhold their authority; that is, anonymous operation is not allowed—the primary justification for this limitation is to deter malicious agent activity.

Permits: Permits determine what agents and places can do—they are sets of capabilities. In general, these capabilities may have virtually any form, but in Telescript they come in two flavors. The first type of capability determines whether an agent or place may execute certain types of instructions, such as instructions that create new agents. The second type of capability places resource limits on agents, such as a maximum number of bytes of network traffic that may

be generated or a maximum lifetime in seconds. If an agent attempts to exceed the limitations imposed by its permits, it is destroyed. The actions permitted an agent are those which are allowed by both its internal permits and the place(s) it visits.

Other issues: A number of details must be taken into account when designing an architecture to support mobile agents, but one of the fundamental issues is the choice of language for implementation of the agents (which might differ from the language used to implement the agent *architecture*).

Migration strategies

To support the migration of agents, it must be possible to either capture the state of an agent or to spawn an additional process that captures the state of the agent. This process state must then be transmitted to the remote machine to which the agent (or its child, in the case of spawning an additional process) will migrate.

Service Discovery Middleware – Finding Needed Services

Introduction

- Mobility introduces interesting challenges for the delivery of services to clients, since mobile devices are typically more resource-poor than their wired counterparts and know much less about their current environment.
- Service discovery frameworks make networked services significantly less tedious to deploy and configure and can be used to build rich mobile computing environments. In a service discovery enabled network, for example, a printer becomes usable (and discoverable) as soon as it is plugged in. This reduces configuration hassles and saves valuable systems administration time, since the printer adjusts to its surroundings with little additional help. Similarly, users have a better computing experience: enabled clients (e.g., a word processor) can immediately find and use the printer without forcing the user to manually search for the printer, identify its type, and then download and install device drivers.
- Services that are more interesting than the usual examples—printing, scanning, etc.—can also be enabled by service discovery technologies. An enabled keychain in a user's purse could turn on lights, transfer desktop settings, or adjust stereo systems as they move about. The same device might also suck up electronic business cards automatically when the user attends a meeting, or make a copy of a diagram scribbled on an enabled

whiteboard. Remote file storage services can be deployed to extend the limited storage capacity of small mobile computers like Personal Digital Assistants (PDAs).

- The most basic interactions between clients and services are service advertisement and service discovery. *Service advertisement* allows enabled services to announce their presence when they enter the network and to announce their demise when they leave the network.
- The advertisement typically includes necessary contact information and descriptive attributes or information that will allow these descriptive attributes to be discovered. From the client point of view, *service discovery* allows clients to dynamically discover services present either in their local network environment or on a larger scale (e.g., in the global Internet). In some cases, services are sought directly; in others, one or more service catalogs are discovered and these catalogs queried for needed services. The discovery attempt typically includes information about the type of services needed, including the standardized service type name(s) and service characteristics. These characteristics might identify the specific location of a service, device capabilities (e.g., duplex capability for a printer) in the case of hardware services, and accounting information such as the cost to use the service.
- Whether services are sought directly or a catalog is consulted, a client needs very little information about its environment it can locate services (or service catalogs) dynamically, with little or no static configuration. Service characteristics, such as the protocols necessary for communication, descriptive attributes, etc. can all be determined dynamically. Service discovery frameworks also standardize the operation of service catalogs, garbage collection facilities, security, and the development of protocols for communication between clients and services.
- The primary advantage provided to both developers and end users by service discovery protocol suites is standardization—none of the client/server interactions (such as discovery, advertisement) are particularly magical and could be developed in an ad hoc fashion by any competent programmer. But standardization brings “out of the box” compatibility to a diverse set of clients and services. Of course an implementation of a service discovery framework will necessarily provide concrete implementations of discovery, advertisement, and eventing, potentially saving a developer a significant

amount of effort over developing sophisticated client/server systems from scratch. This chapter examines the common components of service discovery frameworks in detail, exploring a range of possible design strategies, and drawing specific design choices from the range of currently deployed frameworks. The chapter is not intended as a primer for application *development* under service discovery—this would require substantially more space and there are several good books that cover development.

Most service discovery frameworks address a large subset of the following concepts:

- **Standardization of services.** In order to support dynamic discovery of services, service types must be standardized. In the standardization process, the essence of a service is defined; this includes the operations that the service supports, the protocols it speaks, and descriptive attributes that provide additional information about the service.
- **Discovery of services.** Needed services may be discovered on demand, with minimal prior knowledge of the network. This is really the point of service discovery. Typically, clients can search for services by type ("digital camera") or by descriptive attributes ("manufactured by Cameras, Inc."), or both. The *richness* of provided search facilities varies considerably among current service discovery offerings. More powerful search facilities allow clients to more carefully fine tune their discovery requests, while lightweight facilities are appropriate for a wider range of devices, include devices with severe resource constraints, such as cellular phones.
- **Service "subtyping".** Clients may occasionally be interested in a very specific type of service—for example, a high-resolution color laser printer might be required to print a digital photograph. In other cases, only basic black and white printing services are required (e.g., to print a shopping list). Service sub typing allows a client to specify a needed service type with as much (or as little) detail as necessary. Service sub typing allows the bare essence of a service type to be standardized, and more specific instances of a service type to inherit and expand upon this essence.
- **Service insertion and advertisement.** Service advertisement allows the dynamic insertion into and removal of services from a network, providing an extension of "plug and play" technologies into a networked environment. Services slip into a network with a minimum of manual configuration and advertise their availability, either directly to clients, or to servers maintaining catalogs of services. Conversely, services leaving a

network in an orderly fashion (as opposed to crashing) can advertise their demise. A primary difference between service discovery technologies and relatively static information services like the Domain Name Service (DNS) or Dynamic Host Configuration Protocol (DHCP) is that is that service discovery technologies allow highly dynamic updates—services appearing or disappearing result in immediate reconfiguration of the network. In contrast, DNS and DHCP rely on static files or databases that are configured by systems administrators with higher levels of authority than typical users.

- **Service browsing.** Browsing allows clients to explore the space of currently available services without *a priori* knowledge of the network environment and without any specific service types in mind. Service browsing is to "window shopping" as service discovery is to a focused attempt to buy a specific item. Information obtained through service browsing might be presented to the user in a graphical user interface. A user could then choose to interact with whichever services are sufficiently interesting.
- **Service catalogs.** Though some discovery frameworks operate entirely in a peer-to-peer fashion, putting clients directly in touch with services from the very start, some support catalogs that maintain listings of available services. When service catalogs are implemented, services perform advertisement against one or more catalogs rather than interacting with clients directly. Similarly, clients query catalogs for needed services rather than searching the network for services. There are some substantial advantages to deploying service catalogs, including greater flexibility in deploying services beyond the local network segment and a reduction in multicast traffic.
- **Eventing.** Evening allows asynchronous notification of interesting conditions (e.g., a needed service becoming available or an important change in the state of a service, such as a printer running out of paper or a long computation being completed). An eventing mechanism replaces polling, providing more timely notifications of important events, making software development more straightforward, and reducing the burden on network resources.
- **Garbage collection.** Garbage collection facilities remove outdated information from the network, including advertisements associated with defunct services and subscriptions to eventing services. Without garbage collection, performance could

suffer significantly, as clients try to contact non-existent services or services continue to perform operations (such as eventing) on behalf of crashed clients. Garbage collection is critical for the proper operation of service catalogs, as well, which would overflow with outdated information in the absence of such a facility.

- **Scoping.** Service discovery frameworks typically provide a mechanism for controlling the scope of both service discovery and service advertisement. Scoping is addressed in two different ways.

Services

- Services provide benefits to clients, such as file storage, printing, faxing, and access to high-performance computing facilities, in the same sense as the "server" in traditional client/server. The dynamicity added to the client/server paradigm by service discovery introduces some new concerns, such as globally unique identifiers for services, so that individual service instances can be tracked, how services are located, and methods for standardization.

XML description documents to specify services. A description document for our blender service type might look like this:

```
<?xml version="1.0"?>
<root xmlns="urn:schemas-upnp-org:device-1-0">
 <specVersion>
  <major>1</major>
  <minor>0</minor>
 </specVersion>
 <URLBase>http://10.0.0.13:5431</URLBase>
 <device>
  <deviceType>urn:schemas-upnp-org:device:blender:1</deviceType>
  <friendlyName>UPnP Blender</friendlyName>
  <manufacturer>University of New Orleans
```

```
            Dept. of Computer Science
  </manufacturer>
  <manufacturerURL>http://www.cs.uno.edu/</manufacturerURL>
  <modelDescription>UPnP-compatible blender with Accublend Whirring
  </modelDescription>
  <modelName>Plug-N-Blend Deluxe</modelName>
  <modelNumber>UBlend9873A</modelNumber>
  <modelURL>http://www.upnpblend.com/</modelURL>
  <serialNumber>999954321</serialNumber>
  <UDN>uuid:Upnp-Blender-1_0-1234567890001</UDN>
  <UPC>123456789</UPC>
  <serviceList>
    <service>
      <serviceType>
        urn:schemas-upnp-org:service:PowerSwitch:1
      </serviceType>
      <serviceId>urn:upnp-org:serviceId:PowerSwitch1</serviceId>
      <controlURL>/upnp/control/power1</controlURL>
      <eventSubURL>/upnp/event/power1</eventSubURL>
      <SCPDURL>/blenderpowerSCPD.xml</SCPDURL>
    </service>
    <service>
      <serviceType>
        urn:schemas-upnp-org:service:SpeedControl:1
```

```
      </serviceType>
      <serviceId>
        urn:upnp-org:serviceId:SpeedControl1
      </serviceId>
      <controlURL>/upnp/control/speed1</controlURL>
      <eventSubURL>/upnp/event/speed1</eventSubURL>
      <SCPDURL>/blenderspeedSCPD.xml</SCPDURL>
    </service>
    <service>
      <serviceType>
        urn:schemas-upnp-org:service:Bowl:1
      </serviceType>
      <serviceId>urn:upnp-org:serviceId:Bowl1</serviceId>
      <controlURL>/upnp/control/bowl1</controlURL>
      <eventSubURL>/upnp/event/bowl1</eventSubURL>
      <SCPDURL>/blenderbowlSCPD.xml</SCPDURL>
    </service>
  </serviceList>
  <presentationURL>/blenderdevicepres.html</presentationURL>
</device>
</root>
```

Interoperability and Standardization

- The wired infrastructure has been designed and deployed around a rich set of international standards. For example, the legacy local area network (LAN) consists of such technologies as Ethernet, Token-Ring, and Token-Bus which were defined in precise details by the IEEE 802 committees. Moreover, newer network technologies like FDDI, HIPPI, and Fibre Channel have been defined by the ANSI X3T working groups with a mature set of approved specifications.
- Both the IEEE and the ANSI bodies add further credibility to their work by helping international organizations like ISO and ITU to easily migrate the specifications into international standards bodies for worldwide acceptance.
- Similar efforts are underway in the ATM Forum to create a set of implementation agreements which should permit interoperability between different vendor implementations and products.
- The wireless industry currently embraces a small number of standards. The closest effort is within the IEEE 802 working group which recently completed the IEEE 802.11 Wireless MAC (media access control) standard. The primary objective of the IEEE 802.11 effort is to permit wireless LANs from different vendors to interoperate.
- IEEE 802.11 does not, however, address the needs of the wide area wireless networking industry which currently deploys various packetized protocols (e.g. CDPD, GPRS) across unused cellular channels.
- Each network type is based on its own set of assumptions about the kinds of service the customers are willing to purchase. Service providers for each of these types of networks have different goals and strategies and do not seem likely to provide interoperability among the other classes of service.
- Other mobile infrastructures are also lacking in internationally recognized standards. This is evident in the cellular telephone industry: a PHS telephone will not function in a cell serviced by a GSM or PCS infrastructure. The same is true for any combination of the aforementioned technologies.
- Moreover, cordless telephones, infrared transmission, satellite channels, and most mobile communication systems are either based on proprietary data interfaces, or have implemented selected parts of existing and/or emerging deployment agreements. Such key attributes as Quality of Service, Location Register contents,Database formats, update

policies, and data exchange rates are left to the equipment providers and service providers which may be based more on deployment schedules than on availability of standards and interoperability guarantees.

- The emerging UMTS system standard which is expected to be deployed by the year 2002, will provide a golden opportunity for interoperability of data links interfaces, digital voice, and wireless data and services.
- Many of the client-application products, or the communications substrate that they rely on, are recognizing that several competing wireless transmission protocols exist with each network type. They also recognize that the number of such protocols may grow or shrink. As a result, these client-level packages are adapted to use the popular underlying protocols. This limited form of interoperability appears to meet the needs for developers of client software. As an example, the Oracle Mobile Agents product discussed previously supports both CDPD and Shiva PPP. However, no client software we have seen claims to migrate seamlessly among the different wireless network classes.
- At some level, interoperability among the various network classes can be provided by adopting popular communications standards. For example, those client applications developed to exploit TCP/IP in wired networks can interoperate without change in the wireless domain if some variant of TCP/IP is offered as a service, e.g. lETF's Mobile IP. However, the quality of service that is provided by this approach may not be transparent, or even acceptable.
- In addition, it remains the client's responsibility to transfer among the various competing network services.
- The heterogeneity of the existing and emerging wireless network protocols poses not only a need for interoperability, but also a stringent quality of service requirements. This is because the inherent unreliability and bandwidth limitation largely varies from one network to the other, leading to rapid fluctuations in the quality of the provided services. Recent research efforts proposed extensions to formal open systems standards.
- The wireless application protocol (WAP) standard currently being developed by the WAP forum group offers an OSI-like protocol stack for interoperability of different wireless networks. The W AP stack allows applications to register interest in quality of

service events and thresholds (QoS). This in turn, allows the application to be mobility-aware and adaptable to changes in the environment.

- The WAP stack also provides negotiation protocols between producers and consumers of data to optimize the necessary level of data presentation based on the nature of data, the current wireless network between the source and the destination, and the capabilities of the destination device. Content negotiation should plays major role in maintaining QoS across heterogeneous networks.

UNIT III

Mobile Networking: Virtual IP Protocols - Loose Source Routing Protocols - Mobile IP – CDPD – GPRS – UMTS - Security and Authentication – Quality of Service – Mobile Access to the World Wide Web.

Mobile Networking

- Internetworking mobile computers with the fixed-network raises the additional requirements of mobility transparency and mobility and location management. The mobility behavior of a node should be transparent to a peer node.
- A peer node should be able to communicate with a mobile node using some fixed IP address irrespective of the current point of attachment.
- The mobile networking protocol should also be transparent to the hosts and routers which do not understand or support mobility. Thus, the mobility unaware routers should be able to route packets destined to a mobile host as normal IP data packets. Senility is another important concern in internetworking, In mobile networking it is more so, since the mobile nodes will be visiting foreign networks, requesting services, and accessing data. Thus, it is important that the security of the visiting network is not breached due to the presence of a foreign node in its network.
- Authentication of the mobile nodes and foreign networks is also important. Thus, at a minimum, mobile networking protocols should provide authentication and security features comparable to those found in fixed-network IP protocols such as IPv4 and IPv6.

In this section, various approaches and protocols for mobile internetworking are examined, including:

Early approaches: virtual **IP** mechanisms

- Loose source routing protocol
- The Mobile Internet Protocol (Mobile-IP)
- Cellular Digital Packet Data (CDPD)
- The General Packet Radio Service protocol (GPRS)

Emphasis is placed on protocol mechanisms, leaving out the details which can be obtained by following cited work and web resources.

Early Approaches: Virtual IP Protocols

In this approach, a mobile network is a virtual network with a virtual address space. A mapping

is maintained between the physical or actual IP addresses and the Virtual IP addresses. This mapping is performed by the mobile host which obtains a care of address from the local network being visited using either the Dynamic Host Configuration Protocol (DHCP) or the BOOTP protocols or by any of the link layer protocols.

Dynamic Host Configuration Protocol (DHCP)

- Clients should require no manual configuration. Each client should be able to discover appropriate local configuration parameters without user intervention and incorporate those parameters into its own configuration.
- Networks should require no manual configuration for individual clients. Under normal circumstances, the network manager should not have to enter any per-client configuration parameters.
- DHCP should not require a server on each subnet. To allow for scale and economy, DHCP must work across routers or through the intervention of BOOTP relay agents.
- A DHCP client must be prepared to receive multiple responses to a request for configuration parameters. Some installations may include multiple, overlapping DHCP servers to enhance reliability and increase performance.
- DHCP must coexist with statically configured, non-participating hosts and with existing network protocol implementations.
- DHCP must interoperate with the BOOTP relay agent behavior as described by RFC 951 and by RFC 1542
- DHCP must provide service to existing BOOTP clients.

Sunshine and Postel

The earliest solution for managing mobile hosts was proposed by Sunshine and Postel in 1980. They proposed that the mobile hosts be assigned a virtual **IP** address

which can be used to identify them. A mobile host in the foreign network is required to obtain a care-of-address, and to update its location in a mapping database. When a packet has to be routed to the mobile host, its current location is looked up in the database and the packet is transmitted to that location.

- All mobile hosts belong to a "virtual network".
- Current locations are maintained in a global database.
- This database queried by senders and updated by mobile hosts.

The Sony Protocol

- This protocol was proposed in 1992 by F. Teraoka et al. of Sony Laboratories. **In** this scheme, a mobile host has two **IP** addresses associated with it.
- A virtual address, which is immutable and by which it is known to the outside world, and a physical address, which is acquired from the local network. Two sub layers are introduced in the network layer and are used to map the physical address to the virtual address.
- The transport layer interfaces with the network layer through the virtual layer interface and addresses its packets to the virtual address of a mobile host. A cache called the Address Mapping Table (AMT) is used for fast address resolution. A copy of this cache is maintained at each host/router. The VIP (Virtual IP) is implemented as an IP option. A set of packet types is also defined for host communication.
- On entering a foreign network, the mobile host obtains an **IP** address and informs its home network of its current location. The home network broadcasts this information so the AMT cache gets updated.
- A stationary host, when required to communicate with a mobile host, looks up its cache. If the mapping is available, the packet is transmitted in the normal fashion by appending the VIP header. If the cache entry is not available, the packet is addressed to the VIP address. A set of connection gateways are required for the co-existence of mobility aware and mobility unaware hosts on the network.

Loose Source Routing Protocols

- This approach was proposed by David Johnson in 1993. It uses the Loose Source Route option available in the IPv4 for routing packet data. The option allows the source to specify the intermediate gateways in the IP packet. Thus, the source can control the route the IP packet takes. At each destination, the gateway picks up the next IP address from the IP packet, sets it as the destination, and advances a pointer stored in the IP packet header.
- The home network maintains a database of all mobile hosts native to its network. When a mobile host changes location, it informs its home network of its new location.
- When an IP packet destined to the mobile host arrives at the home network, the packet is forwarded to the mobile host at the current location address, and the corresponding source host is informed of the current location of the mobile host. The corresponding host can use this information to cache the location, thus avoiding communication with the home network until the mobile host changes its location again. Source route set up is done by the corresponding host.

IP Loose Source Routing

- The IP standard [Postel 81b] defines an option called *Loose Source and Record Route* (or *LSRR*) that may be used in sending an IP datagram in order to cause the datagram to be routed through a series of intermediate 3 gateways before delivery to the ultimate destination host. The route specified is "loose" in that the normal IP routing algorithm is used to deliver the datagram, over any number of intervening hops, to each succeeding address in the route.
- The sender thus need not know the complete path (between the listed gateways) needed to route the datagram through the Internet to its destination.
- The format of the LSRR option in the IP datagram header is illustrated in Figure 2. The first byte of the option gives the option type code to identify this as an LSRR option. The second byte specifies the total length of the option (in bytes), including the type code, length, and pointer fields.
- The third byte is used as a pointer to indicate the current position (in bytes) in the listed route, relative to the beginning of the LSRR option. The remainder of the option consists of a sequence of IP addresses (4 bytes each) through which the datagram should be routed.

- In routing a datagram through the Internet with an LSRR option, the datagram is first routed to the IP address specified in the destination address field in the IP header.
- Once delivered to that gateway, the IP destination in the header is replaced with the first IP address specified in the LSRR option, and the LSRR pointer is incremented to point at the next IP address in the route (incremented by 4 bytes).
- The datagram is then routed to this new destination copied from the option. Once received at that gateway, the next address is taken from the route listed in the option, and so on, until the end of the route (until the pointer has been incremented past the end of the option). The datagram is then routed normally to the final IP address taken from route listed in the option.
- The LSRR option also creates a record of the gateways through which the datagram has been routed by the option. As each gateway copies the next address from the route into the destination address field of the IP header, it replaces that entry in the route in the option with its own gateway address for the network interface through which it will be transmitting the datagram next.
- Only the gateways named in the source route add their own address to the recorded route. The total length of the option thus remains constant, as each address in the recorded route replaces exactly one address in the original option.

IP Datagram Delivery to Mobile Hosts

To use the LSRR option for IP datagram delivery to mobile hosts, the source route is set to route the datagram through the gateway to the foreign network to which the mobile host is currently attached.

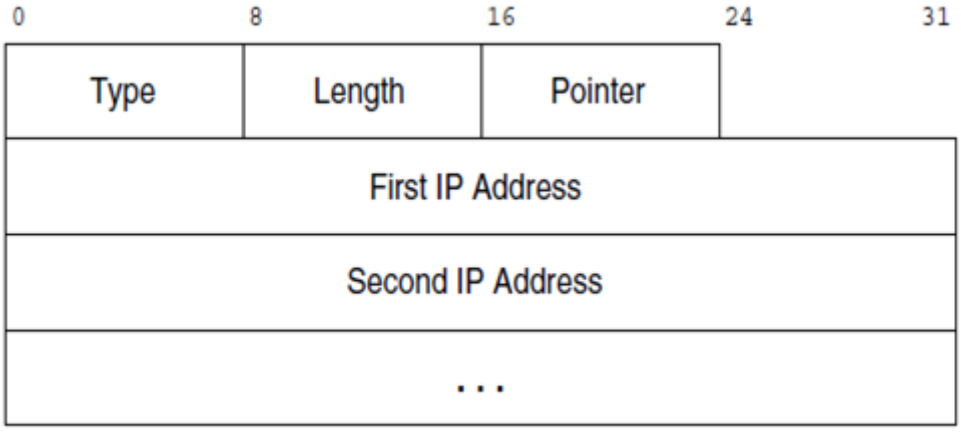


Fig:The IP Loose Source and Record Route (LSRR) option

Required Software Support

The use of the standard IP loose source routing (LSRR) option for routing IP datagrams to mobile hosts greatly reduces the amount and complexity of modifications to existing network software required to support mobile hosts. Since LSRR is an IP "option," though, it is sometimes incorrectly omitted from IP implementations. In IP, however, what is optional is the *use* of any particular option, not its implementation, and the support for interpreting and routing datagrams using LSRR is currently required of all gateways on the Internet.

The Mobile Internet Protocol (Mobile-IP)

The Mobile Internet Protocol (Mobile IP) defines enhancements to the Internet Protocol to allow routing of IP packets to mobile nodes in the internet. The IP version 4 assumes that the Internet Protocol of a node uniquely identifies the point of attachment of the node to the inter network.

Packets are routed based on the IP address. In a mobile environment, the point of attachment of the mobile node will be different from time to time, and the mobile nodes could be attached to different networks. For IPv4 to work correctly in the mobile environment, the mobile node will either have to be assigned a new IP address every time it changes its point of attachment, or the host specific routing information has to be supplied throughout the network. Both of these alternatives result in scalability and connection management problems. The mobile IP protocol describes a mechanism which allows nodes to change their point of attachment on the Internet.

The major architecture components of the mobile IP protocol are:

Mobile Node (MN): is a host or a router that changes its point of attachment to the network from one sub network to another. The MN is known throughout the network by an IP address assigned to it in the home network. The mobile node can communicate from any location as long as the link layer connectivity to the internetwork is established.

- Home Agent (HA): is a mobile-IP capable router on the mobile node's home network. The HA maintains the location information for the mobile node. It also acts as the tunneling agent for packets destined to the mobile node. The HA manages the registration and authorization information of all the mobile modes belonging to its network.

- Foreign Agent (FA): is a mobile-IP capable router that the mobile node has visited. After

attaching to the foreign network the mobile node is required to register itself with the FA. The FA detunnels and routes the packets destined to the mobile node. The FA may also act as a default router for mobile nodes registered with it.

The mobile IP protocol can be summarized as follows:

1. The mobility agents (HA and FA) in the network broadcast their availability through agent advertisement packets.
2. The mobile node, after connecting to a network, receives information about the mobility agents through the agent advertisement broadcasts. Alternatively, the mobile nodes can solicit the agent information if no broadcasts have been received.
3. The mobile node determines the network it is attached to. If it is connected to the home network, it operates without mobility services. If it is returning back to the home network, the mobile node deregisters itself with the HA and operates without mobility services.
4. If the mobile node is attached to a foreign network, a care-of-address is obtained from the FA.
5. The mobile node operating from a foreign network registers itself with its home agent. The foreign agent then acts as a relay in this registration process.
6. When the mobile node is away from its home network, datagrams destined to the mobile node are intercepted by the home agent, which then tunnels these datagrams to the mobile node's care-of-address. The tunneled packets destined to the mobile node are detunneled either by the foreign agent or by the mobile node itself. In the latter case, the mobile node obtains a temporary IP address on the foreign agent network to be used for forwarding. This can be done using the IETF Dynamic Host Configuration Protocol (DHCP).
7. The datagrams originating from the mobile node are routed in the normal fashion. The foreign agent may act as a default router in this case.

The routing path of a datagram sent from a fixed host to a mobile node is as follows: (1) the datagram is sent from the fixed host to the home agent using standard IP routing; (2) the home agent encapsulates the received datagram inside another datagram and sends it to the foreign agent (IP-in-IP tunneling); (3) the encapsulated IP packet is received by the foreign agent, decapsulated, and forwarded to the mobile node; (4) the mobile node replies by sending a datagram to the fixed host through the foreign agent.

The Mobile IP protocol stack on the fixed network and on the mobile unit is depicted in Figure

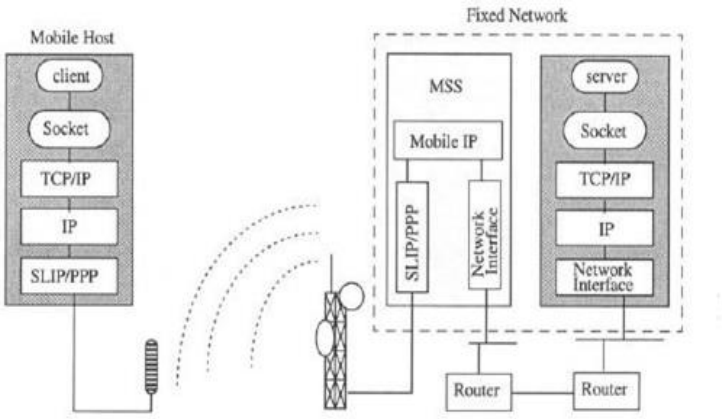


Fig: Mobile Internet architecture using Mobile-ip

The Mobile Host Protocol, known as Mobile-IP, is an evolving standard being developed by the IETF Working Group on IP Routing for Wireless/Mobile Hosts. Standards for both IPv4 and IPv6 have been proposed and are being reviewed for enhancements in scalability and performance. In particular, the triangular routing between the mobile node, the home agent, and the foreign agent (that must be performed every time the mobile node switches over to another communication cell) is a bottleneck that is being removed in IPv6. Packets addressed to the mobile node's home address are transparently routed to its care-of address. The optimized protocol enables IPv6 nodes to cache the binding of a mobile node's home address with its care-of address, and to then send any packets destined for the mobile node directly to it at this care-of address

Support for Ad-Hoc Mobility

- An ad-hoc mobile network is a collection of wireless mobile nodes forming a temporary network without the aid of any established infrastructure or centralized administration. Examples of ad-hoc networks include wireless portable devices of a group of collaborator, such as an emergency team in a disaster area.
- No routing is needed between ad-hoc nodes which are within transmission range of each others. Otherwise, additional nodes must be used to form a sequence of hops from the source to the destination. Routing algorithms in the ad-hoc environment are

therefore a necessary support for this mode of mobile connection.

- Traditional routing algorithms used in wireline networks use distance vector or link state routing algorithms, which rely on periodically broadcasting routing advertisements by each router node.
- The distance vector algorithm broadcasts its view of the distance from a router node to each host. The link state routing algorithm broadcasts its view of the adjacent network links. Neither algorithms is suitable for the ad-hoc environment because periodic broadcasts will drain battery power quickly.
- Research in ad-hoc routing is dedicated to finding algorithms that avoid the needless battery consumption and the inefficient use of the wireless bandwidth. Dynamic source routing is one such algorithms .
- It allows for route discovery, route maintenance, and the use of route caches. To discover an available route, a source node sends out a route request packet indicating the source, the target nodes, and a request identifier. When a mobile node receives a route request packet, it checks a list of recently processed requests. If a request is found for the same source and request id, the request is dropped and no further action is taken.
- Otherwise, the address of the node servicing the request is added to the route request packet before the packet is re-broadcasted. However, if the address of the node servicing the request is identical to the target node address, the requested route is discovered, and a reply is sent to the source node.
- Due to unpredictable node mobility, cached routes may become incorrect. Route maintenance is therefore necessary in this environment. This is achieved by requiring nodes routing packets to acknowledge successful forwarding and to send error messages to the source node if a route ceases to exist. Active monitoring such as MAC-level acknowledgements, as well as passive monitoring (listening to nearby broadcast, in a promiscuous mode), can be used in route maintenance.

Cellular Digital Packet Data (CDPD)

CDPD is a connectionless multi-network protocol, proposed originally by the COPO Forum (now called the WOF Forum). It is based on the early versions of Mobile-IP. The idea behind CDPD is to share unused channels in existing Advanced Mobile Phone Systems (AMPS) to

provide up to 19.2 kbps data channel.

Even though CDPD and Mobile-IP are similar, their terminologies are different. CDPD follows the OSI model terminology. For example, the mobile node is called a Mobile End-System (M-ES); the home and foreign agents are called Mobile Home and Mobile Serving Functions (MHF and SF respectively) and reside in a mobile data intermediate system (MD-IS). A Mobile Database Station (MOBS) is also defined which deals with the air link communications and acts as a data link layer relay between the M-ES and the serving MD-IS. Two protocols, the Mobile Node Registration Protocol (MNRP) and the Mobile Node Location Protocol (MNLP), are responsible for registration of the M-ES with its home MD-IS and the proper routing of packets destined for the M-ES.

The main resemblance between CDPD and Mobile-IP is in the triangular routing approach between the mobile node and the home and foreign agents. The main differences can be summarized as follows

- The user's IP address must be assigned by the CDPD service provider.
  Mobile IP makes no such assumptions.
- Mobile IP allows the mobile node to also be a foreign agent. Combining the M-ES and the Serving MD-IS was not considered and is not practical in CDPD.
- CDPD's mobility tunnelling is based on CLNP. Mobile IP's mobility tunnelling is based on the IP-in-IP protocol, which is IP-based.
- Mobile IP operates completely above the data link layer. CDPD mobility, on the other hand, is mostly above the data link layer.
- Since the infrastructure of the CDPD network is closed there are less security considerations for CDPD.

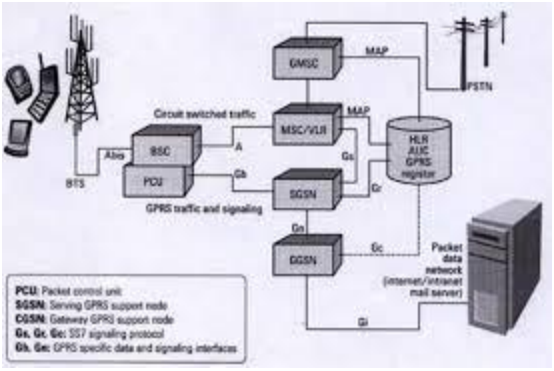
While the standardization process of Mobile IP has been progressing rather slowly, CDPD has been deployed for a few years now, and is receiving the support of major AMPS carriers. However, due to its lack of openness, the future of CDPD deployment and/or acceptance can only be guessed.

How does CDPD work?

- To effectively integrate voice and data traffic on the cellular system without degrading the level of service provided to the voice customer, the CDPD network implements a technique called channel hopping. The way this works is that when a CDPD mobile data unit desires to initiate data transmission, it will check for availability of a cellular channel. Once an available channel is located, the data link is established.
- As long as the assigned cellular channel is not needed for voice communications, the mobile data unit can continue to transmit data packet bursts on it. However, if a cellular voice customer initiates voice communication, it will take priority over the data transmission.
- At such time, the mobile data unit will be advised by the Mobile Data Base Station (which is the CDPD serving entity in the cell and constantly checks for potential voice communication on the channel) to "hop" to another available channel. In the event that there are no other available channels, then data transmission will be temporarily discontinued.
- It is important to note that these channel hops are completely transparent to the mobile data user. As far as the user can see, there is only one data stream being used to complete the entire transmission.

The GSM General Packet Radio Service (GPRS)

GPRS is a GSM packet data service developed by the European Telecommunication Standards Institute (ETSI) as part of GSM phase 2+ developments. The goal of GPRS was to support data transfer rates higher than the 9.6 kbps achieved through GSM's circuit switching technology. Unlike Mobile-IP, GPRS is not restricted to IP packet data protocols, and offers connection to standard protocols (such as TCPIIP, X.25, and CLNP) as well as specialized data packet protocols. Mobile-IP, however influenced the design of Mobility management in GPRS.



networks, mobility management with the GPRS registers, and delivery of data packets to MSs, independently of their locations.

- One GSN is designated the Gateway GSN (GGSN) and acts as a logical interface to external packet data networks.
- The GGSN is similar to the home agent in Mobile-IP. It updates the location directory of the mobile station (MS) using routing information supplied by the Serving GSN node (SGSN). The latter is similar to the foreign agent in Mobile-IP. GGSN also routes the external data network protocol packet encapsulated over the GPRS backbone to the SGSN currently serving the MS. It also decapsulates and forwards external data network packets to the appropriate data network and handles the billing of data traffic.
- The SGSN is responsible for the delivery of packets to the mobile stations within its service area.
- The main functions of the SGSN are to detect new GPRS MSs in its service area, handle the process of registering the new MSs along with the GPRS registers, send receive data packets to/from the GPRS MS, and keep a record of the location of MSs inside of its service area.

- The GPRS register acts as a database from which the SGSNs can ask whether a new MS in its area is allowed to join the GPRS network. For the coordination of circuit and packet switched services, an association between the GSM MSC and the GSN is created. This association is used to keep routing and location area information up-to-date in both entities.

UMTS

Refer unit 2

Security and Authentication Issues in Mobile Networks

- In a mobile computing environment, it is desirable to protect information about the movements and activities of mobile users from onlookers.
- In addition to the basic security concerns in wireline systems (authentication, confidentiality, and key distribution), a new issue is the privacy and anonymity of the user's movement and identity.
- In fact, a typical situation arises when a mobile user registers in one domain (home domain) and appears in a different foreign domain; the user must be authenticated and his solvency must be confirmed. Usually during this process the user has to provide a non-ambiguous identity to his home domain and has to verify it. If no care is taken, this identity can be tapped on the air interface in a cellular environment or through the signaling protocols exchanged on the registered wired network.
- In CDPD, all the mobility management, as well as security-related activity, are concentrated in the Massage-Data Intermediate System (MD-IS).

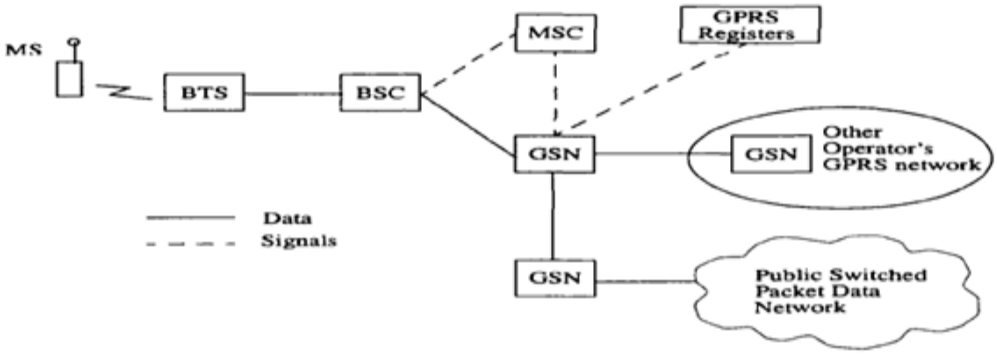


Fig: The GSM General Packet Radio Service

- Each MD- IS controls an area covered by a number of base stations. Upon arrival to a new area, the mobile unit engages in a key exchange protocol with the local MD-IS. As a result, both parties obtain a shared secret key.
- Subsequently, the mobile unit encrypts its real identity (Network Equipment Identifier) and transmits it to the local MD-IS. This approach allows the local MD-IS to discover the real identity of the mobile unit.
- Unfortunately, the key exchanging protocol itself is not secure. This means that an active attacker masquerading as the local domain authority can engage in the key exchange protocol with the mobile unit and obtain a shared key.

Quality of Service

- Mobile network protocols such as Mobile-IP and GPRS provide mobility transparency at the network layer level. This allows the higher layers of the protocol stack to be used unchanged. Unfortunately, there are ill consequences to this transparency that are mostly attributed to the constraints of the wireless and mobile environment.
- For example, transport layer protocols that rely heavily on timeout mechanisms for re-transmission, if used unchanged, will perform poorly under variable delays and limited bandwidth. This is especially true for applications that require continuous-media streams. Existing session protocols are not of much use under frequent disconnections and reconnections of the same mobile computation.
- Similarly, existing presentation layer protocols are inappropriate to use unchanged in the wireless and mobile environment. For example, a user with a limited display and limited battery PDA will not be able to browse the Web unless the presentation of the downloaded data is changed to suite her PDA's capabilities. Regardless of which particular upper layer in the protocol stack suffers the consequences of transparency, the effect on the end-user will always be felt as unacceptable fluctuations in the perceived QoS.
- we describe the following three research efforts that address QoS concerns in the wireless and mobile environment.

Optimizing TCP/IP for Mobile Networks*:* Transport or network layer solutions to get TCP/IP to work despite the fluctuations in the underlying network QoS. Solutions are not application-sensitive and do not address an overlay of heterogeneous networks.

QoS driven, high-level communication protocols*:* Session and/or application layer protocols directly addressing QoS parameters. Solutions are sensitive to applications, but do not address network heterogeneity issues.

QoS driven, full protocol stacks: All layers are aware of either QoS or the limitations introduced by mobility and by the wireless networks. Two research efforts will be discussed including, the BARW AN project and the Wireless Application Protocol (W AP) standard.

Optimizing TCP/IP for Mobile Networks

- Since mobile users will need connection-oriented communication to obtain remote services, they will have to use transport protocols developed for the fixed network. Unfortunately, such protocols like TCP perform poorly when used unmodified in the mobile network.
- For example, TCP acknowledgment timeout is in the range of tens of milliseconds. A mobile unit crossing cell boundaries blanks out during a hand-off procedure that could last up to 1,000 milliseconds. This leads to sender timeouts and repeated re-transmissions.
- Another source of re-transmission is the high error rate inherent in the wireless transmission characteristics. Another problem that can lead to performance degradation under standard TCP is bandwidth allocation under unpredictable mobility. An unpredicted number of mobile users can move into the same cell, thus competing on sharing the limited wireless link. Under this scenario, it is difficult to build applications or services that provide performance guarantees or quality of service. A few approaches have been proposed to optimize and extend the standard TCP protocol so that it can be used efficiently under a mobile network protocol such as Mobile IP.

QoS Driven, High-Level Communication Protocols

- Optimizing the behavior and performance of transport protocols is not sufficient to maintain the QoS required by applications.

- For example, most Web browsers use multiple TCP connections to access a multimedia page. While this parallelism achieves speedup in the fixed network, it is slow and inappropriate in the wireless and mobile environment.
- In addition to transport optimizations, what was found needed are application-aware (or application-specific) mechanisms to monitor, request, and maintain QoS from the application or user point of view. This section describes high-level, above-transport protocols that understands application QoS requirements and resource limitations.

The Loss Profile Approach

- Considered the problem of unpredictable mobility and its effect on the degradation of the wireless communication performance. They addressed the case where the aggregate bandwidth required by all mobile units in an overloaded cell exceeds the cell's available bandwidth.
- Their mechanism is simple and relies on policies and measures for discarding parts of the data of the mobile users. Instead of discarding data in an arbitrary manner, guidelines are proposed to avoid discarding critical portions of the data.
- A Loss Profile is proposed and is defined to be a description, provided by the application, of an "acceptable" manner in which data for its connection may be discarded. The loss profile is used in the event of bandwidth reduction at the wireless end of the connection. An elaborate example of a loss profile is given on viewer perception of a video clip under data loss. The loss profile is used by a specialized session layer which is transparent to the application.

QEX: The QoS Driven Remote Execution Protocol

- The work describes a design of a distributed system platform that supports the development of adaptable services.
- The design allows services to tolerate the heterogeneity of the environment by dynamically adapting to changes in the available communication QoS. The implementation of the distributed system is based on APM Ltd.'s ANSA ware software suite, which is based on the ANSA architecture that has had some influence on the ISO Reference Model for Open Distributed Processing (RM-ODP). The purpose of this effort is to propose extensions to emerging distributed systems standards in order to support mobile services.

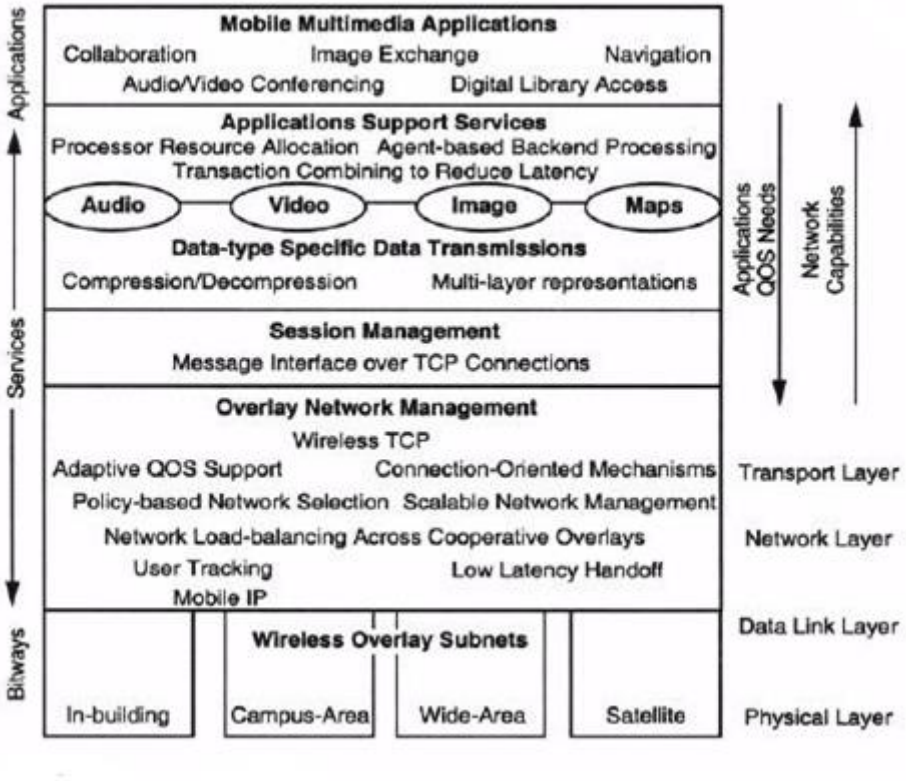
QoS Driven, Full Protocol Stacks

- The mobility of users will force an application to migrate along overlays of networks that vary in their bandwidth, latency, range, and transmission characteristics. Unless the application adapts to variations in the network overlay, the application performance is bound to suffer.
- A network overlay can include a cellular network, a personal communication system (peS), a wireless LAN, an Internet connection, and/or a satellite communication loop, among other networks. In addition to the heterogeneity of networks, the heterogeneity of the mobile platforms imposes a great impediment to mobile application portability.
- Unless applications adapt to the capabilities and limitations of the mobile computer with respect to the type and media of communicated data, applications will remain proprietary to the specific mobile computer platforms they were originally designed for. This section describes a research project that proposes a full stack solution as an overlay network stack atop a heterogeneous collection of wireless subnets. This section also describes an ongoing standardization effort called WAP that aims at proposing a specification of a full ISO/OSI-like network stack that is wireless and mobile aware.

The Wireless Overlay Network Architecture

- The architecture assumes an overlay of various wireless networks ranging from regional-area, wide-area, metropolitan-area, campus-area, in-building, and in-room wireless networks.
- A testbed of wireless overlay network management that supports media-intensive applications has been used to demonstrate the adaptability features of BARWAN. The testbed that has been developed in the San Francisco Bay Area includes the participation of over six local carriers including Nextel and Metricom.
- The testbed integrates the participants' networks and allows full coverage of the greater Bay Area. The BARWAN architecture is gate way centric, meaning it provides gateway connections from the mobile host to each participating wireless networks. Medical imaging applications have been developed to drive the testbed.
- The lowest layer is the wireless overlay subnets, which are the carrier networks including data link interface, and possibly carrier network routing. The details of this layer depends

on the specific subnets being integrated.



- Next is a layer called the Overlay Network Management Layer which includes network and transport functionalities including location tracking, QoS-based hand-off management, other QoS services, and connection-oriented transport mechanisms. The next higher up layer is the Session Management Layer which provides a "transactional" transport (called message-oriented interface).

The layer attempts to optimize transport connections related to the same application by session sharing whenever possible.

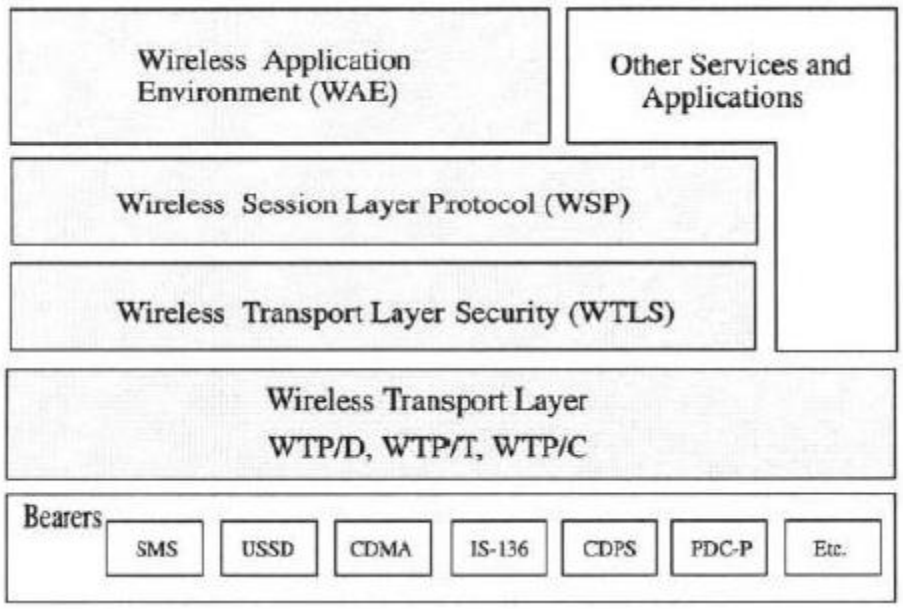
The Wireless Application Protocol (WAP)

The objective was to create the specification of a wireless application environment and a wireless ISO/OSI-like protocol stack. The goal was to provide the needed interoperability to connect different portable devices, via heterogeneous wireless networks, into the internet and corporate intranets. The focus was to bring the internet content and advanced services to digital cellular phones and other hand-held devices such as smart communicators and PDAs.

The session layer, which is the most elaborate layer, also contains critical QoS features including:

- Exception mechanisms to allow applications to register interest in QoS related network

events and parameter thresholds. This allows the application to be mobility-aware, by using QoS API to program how to adapt to changes in the environment.



Mechanisms for capability and content negotiation. This will enable the WAP stack itself to partner through its pieces (on the fixed network, on the wireless network gateways, and on the hand-held device) to perceive and adapt to the mobility and the changes in network characteristics. When certain information content is being delivered the W AP stack negotiates with the device the capability to receive and display the contents. The negotiation decides for the feasibility of the transfer and for the level of filtering that might be needed to deliver the information while maintaining QoS.

MOBILE ACCESS TO THE WORLD WIDE WEB

- More and more users are becoming increasingly dependent on information they obtain from the World Wide Web. Users are also demanding ubiquitous access, any time, anywhere, to the information they rely on.

- Several research efforts explored the problems associated with wireless access to the Web. Most solutions used a Web proxy that enabled Web browsing applications to function over wireless links without imposing changes on browsers and servers.
- Web proxies are also used to pre fetch and cache Web pages to the mobile client's machine, to compress and transform image pages for transmission over low-bandwidth links, and to support disconnected and asynchronous browsing operations.

The Wireless WWW (W4)

A prototype consisting of commercially available PDAs and a wireless LAN has been used to provide a "proof of concept" for the Wireless World Wide Web (W4). A simplified version of Mosaic was ported to the PDA for the purpose of experimenting with response time performance and to sort out design choices. A PDA cache was used to improve the performance.

Dynamic Documents

- The concept of dynamic documents was introduced in an approach to extending and customizing the WWW for mobile computing platforms.
- Dynamic documents are programs executed on a mobile platform to generate a document; they are implemented as Tel scripts as part of the browser client. A modified version of the NCSA Mosaic browser was used to run the dynamic documents it retrieves through a modified Tel interpreter. The interpreter is designed to execute only commands that do not violate safety.
- By using dynamic documents, an adaptive e-mail browser that employs application-specific caching and prefetching is built. Both the browser and the displayed e-mail messages are dynamically customized to the mobile computing environment in which they run. Dynamic documents can solve the problem of limited resources in the mobile host. For example, the Tel script could be a filter that reduces an incoming image so that it fits the screen size or resolution.
- Unfortunately, dynamic documents being placed at the client side are not wireless-media sensitive. This is because filtering occurs after all transmitted information is received by the client. Although caching and prefetching can alleviate some of the communication overhead, excess data (that would be reduced by the dynamic document) is, however, communicated, leading to inefficient utilization of the wireless bandwidth.

Dynamic URLs

- The Mosaic Web client and the URL syntax are modified so that when the user traverses a dynamic URL, the client resolves any references to dynamic information it may contain and sends the result back to the server. This is helpful in defining location-sensitive resources.
- Active documents are Web pages that notify the client browser when dynamic information changes.
- This feature also supports location-sensitive information by keeping the mobile client aware of service relocation or of services offered by a mobile server.

Mobile Browser (MOWSER)

- The design is based on a mediator server that filters retrieved information according to the limitations of the mobile unit. Color, resolution, display mode, sound capability, and maximum file size are among the factors considered.
- The browser, called MOWSER, connects to two servers in the fixed network. The first is the preference server that maintains the user profile; the second is a proxy server that implements all the filtering indicated by the preference server. MOWSER assumes that the user is aware of the mobile unit limitations, which in a way sacrifices transparency. Similar to the dynamic document approach, MOWSER does not directly consider the limitations of the wireless media (although the maximum file size indirectly preserves the limited bandwidth).

WebExpress

Two components are inserted into the data path between the Web client and the Web server: (1) the Client Side Intercept (CSI) process that RUns in the client mobile device and (2) the Server Side Intercept (SSI) process that runs within the wired and fixed network

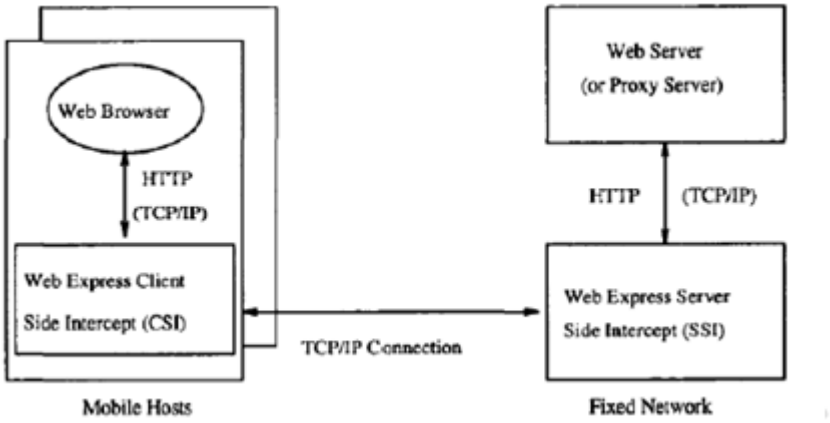


Fig: Web Express proxy intercept model

- The CSI intercepts HITP requests and, together with the SSI, performs optimizations to reduce bandwidth consumption and transmission latency over the wireless link. From the viewpoint of the browser, the CSI appears as a local Web proxy that is co-resident with the Web browser. On the mobile host, the CSI communicates with the Web browser over a local TCP connection (using the TCP/IP "loopback" feature) via the HTTP protocol. Therefore, no external communication occurs over the TCP/IP connection between the browser and the CSI.
- No changes to the browser are required other than specifying the (local) IF address of the CSI as the browser's proxy address. The CSI communicates with an SSI process over a TCP connection using a reduced version of the HTTP protocol. The SSI reconstitutes the HTML data stream and forwards it to the designated CSI Web server (or proxy server). Likewise, for responses returned by a Web server (or a proxy server), the CSI reconstitutes an HTML data stream received from the SSI and sends it to the Web browser over the local TCP connection as though it came directly from the Web server.
- The proxy approach implemented in WebExpress offers the transparency advantage to both Web browsers and Web servers (or proxy servers) and, therefore, can be employed with any Web browser. The CSI/SSI protocols facilitate highly effective data reduction and protocol optimization without limiting any of the Web browser functionality or interoperability.

WebExpress optimization methods are summarized below:

- **Caching:** Both the CSI and SSI cache graphics and HTML objects. If the URL specifies an

object that has been stored in the CSI's cache, it is returned immediately as the response. The caching functions guarantee cache integrity within a client-specified time interval. The SSI cache is populated by responses from the requested Web servers. If a requested URL received from a CSI is cached in the SSI, it is returned as the response to the request.

- **Differencing:** CSI requests might result in responses that normally vary for multiple requests to the same URL (e.g., a stock quote server), The concept of differencing is to cache a common base object on both the CSI and SSI. When a response is received, the SSI computes the difference between the base object and the response and then sends the difference to the CSI. The CSI then merges the difference with its base form to create the browser response. This same technique is used to determine the difference between HTML documents.
- **Protocol reduction:** Each CSI connects to its SSI with a single TCP/IP connection. All requests are routed over this connection to avoid the costly connection establishment overhead. Requests and responses are multiplexed over the connection.
- **Header reduction:** The HTTP protocol is stateless, requiring that each request contain the browser's capabilities. For a given browser this information is the same for all requests. When the CSI establishes a connection with its SSL it sends its capabilities only on the first request. This information is maintained by the SSI for the duration of the connection. The SSI includes the capabilities as part of the HTTP request that it forwards to the target server (in the wire line network).

UNIT IV

Mobile Data Management: Mobile Transactions - Reporting and Co Transactions –Kangaroo Transaction Model - Clustering Model –Isolation only transaction – 2 Tier Transaction Model – Semantic based nomadic transaction processing.

Mobile data management.

Mobile data access can be broadly classified into two categories: (1) data access in mobile client/server, and (2) data access in ad-hoc networks. Several research projects from each category are presented in the following subsections.

Mobile Client/Server Data Access

- In the first category, mobile data access enables the delivery of server data and the maintenance of client-server data consistency in a mobile and wireless environment. Efficient and consistent data access in mobile environments is a challenging research area because of the weak connectivity and resource constraints.
- The data access strategies in a mobile information system can be characterized by delivery modes, data organizations, and consistency requirements, among other factors. The mode for server data delivery can be server-push, client-pull, or a hybrid of both. The server-push delivery is initiated by server functions that push data from the server to the clients. The client-pull delivery is initiated by client functions which send requests to a server and "pull" data from the server in order to provide data to locally running applications.
- The hybrid delivery uses both server-push and client-pull delivery. The data organizations include mobility-specific data organizations like mobile database fragments in the server storage and data multiplexing and indexing in the server-push delivery mode. The consistency requirements range from weak consistency to strong consistency.

Broadcast Disks: A Server PUSH Approach

- When a server continuously and repeatedly broadcasts data to a client community, the broadcast channel becomes a "disk" from which clients can retrieve data as it goes by. The broadcasting data can be organized as multiple disks of different sizes and speeds on the broadcast medium. The broadcast is created by multiplexing chunks of data from different disks onto the same broadcast channel.
- The chunks of each disk are evenly interspersed with each other. The chunks of the fast disks are repeated more often than the chunks of the slow disks (see Figure). The relative speeds of these disks can be adjusted as a parameter to the configuration of the broadcast. This use of the channel effectively puts the fast disks closer to the client while at the same time pushing the slower disks further away.

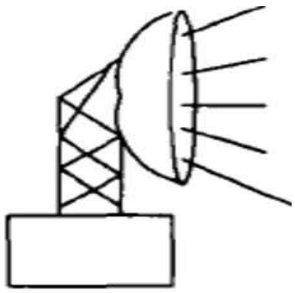


Figure: A simple broadcast disk

Odyssey: A Client PULL Approach

- Odyssey is a CMU research project led by M. Satyanarayanan. It addresses an application-aware adaptation approach to deal with application diversity and concurrency in mobile environments. The application-aware adaptation is implemented with the support of system-coordinated, type-specific operations.
- It supports concurrent execution of diverse mobile applications that execute on mobile clients but read or update remote data on servers. The data accessed by an application may be stored ill one or more general-purpose repositories such as file servers SQL servers, or Web servers. It may also be stored in more specialized repositories such as video libraries, query-by-image content databases, or back-ends of geographical information systems.
- Ideally, a data item available on a mobile client should be indistinguishable from that available to the accessing application if it were to be executed on the server storing that item. But this correspondence may be difficult to preserve as mobile resources become scarce; some form of degradation may be inevitable.
- In Odyssey fidelity is used to describe the degree to which data presented at a client matches the reference copy at the server, Fidelity has many dimensions. One well-known, universal dimension is consistency. For Video applications, data has at least two additional dimensions: frame rate and image quality of individual frames. Odyssey provides a framework within which diverse notions of fidelity can be incorporated.

Rover: A Mobile Objects Approach

- The Rover project at MIT provides mobility support to client server applications based on two ideas: relocatable dynamic object (RDO) and queued remote procedure calls (QRPC). An RDO is an object (code and data) with a well-defined interface that can be dynamically loaded into a mobile client from a server computer, or vice versa, to reduce client-server communication requirements, or to allow disconnected operation.
- Queued remote procedure call is a communication system that permits applications to continue to make non-blocking remote procedure calls even when a mobile client is disconnected; requests and responses are exchanged upon network reconnection.
- Rover gives applications control over the location where the computation is performed.

- By moving RDOs across the network, applications can automate the movement of data and/or computation from the client to the server and vice versa.

Mobile Data Access in Ad-hoc Networks

- The Bayou project at Xerox PARC developed a system to support data sharing among mobile users. The system is intended to support ad-hoc mobility, where no network infrastructure is assumed to be available. In particular, a user's mobile computer may experience extended disconnection from other computing devices.
- Bayou allows mobile users to share their appointment calendars, bibliographic databases, meeting notes, evolving design documents, news bulletin boards, and other types of data in spite of their intermittent network connectivity. The Bayou architecture supports shared databases that can be read and updated by users who may be disconnected from other users, either individually or as a group.
- Bayou supports application-specific mechanisms that detect and resolve the update conflicts, ensures that replicas move towards eventual consistency, and defines a protocol by which the resolution of update conflicts stabilizes.
- Bayou includes consistency management methods for conflict detection called dependency checks and per-write conflict resolution based on client-provided merge procedures. To guarantee eventual consistency, Bayou servers are able to rollback the effects of previously executed writes and redo them according to a global serialization order. Furthermore, Bayou permits clients to observe the results of all writes received by a server, including tentative writes whose conflicts have not been ultimately resolved.
- In the Bayou system, each data collection is replicated in full at a number of servers. Applications running as clients interact with the servers through the Bayou API, which is implemented as a client stub bound with the application. This API, as well as the underlying client-server RPC protocol, supports two basic operations: Read and Write. Read operations permit queries over a data collection, while Write operations can insert, modify, and delete a number of data items in a collection.

Mobile transactions

- A mobile transaction is a long-live transaction whose locus of control moves along with the mobile user. Mobile transactions may access remote data wirelessly, through a weak connection, or may access local replicas of data in disconnected mode. The differences between mobile and distributed transaction management are significant because their goals are different.
- In distributed transactions, the main goal is maximizing availability while achieving ACID properties. In mobile transactions, maximizing reliability while achieving some sort of consistency is the main goal.

Reporting and Co-transactions

- This model is based on the Open Nested transaction model. A computation in the mobile environment is considered to consist of a set of transactions, some of which may execute on the mobile node and some of which may execute on the fixed host.

- The model addresses sharing of partial results while in execution, and maintaining computation state in a fixed node so that the communication cost is minimized while the mobile host relocates.

Nested transaction model

- In the Nested Transaction model, transactions are composed of sub transactions or child transactions designed to localize failures within a transaction and to exploit parallelism within transactions.
- A sub transaction can be further decomposed into other sub transactions, and thus, the transaction may expand in a hierarchical manner. Sub transactions execute atomically with respect to their parent and their siblings, and can abort independently without causing the abortion of the whole transaction. However, if the parent transaction aborts, all its sub transactions have to abort. The parent transaction cannot commit until all its sub transactions have terminated.
- The model proposes to modify Reporting and Co-Transactions to suit mobile environments. The model defines a mobile transaction to be a set of relatively independent transactions which interleave with other mobile transactions.
- A component transaction can be further decomposed into other component transactions allowing arbitrary levels of nesting. Component transactions are allowed to commit or abort independently. If a transaction aborts, all components which have not yet committed may abort. Some of the transactions may have compensating duals and may be compensated.

Commit-Dependency:

If a transaction A develops a commit-dependency on another transaction B (denoted by A B), then transaction A cannot commit until transaction B either commits or aborts. This does not imply that if transaction B aborts, then transaction A should abort.

Abort-Dependency:

If a transaction A develops an abort-dependency on another transaction B (denoted A B), and if transaction B aborts, then transaction A should also abort. This neither implies that if transaction B commits, then transaction A should commit, nor that if transaction A aborts, then transaction B should abort.

The model classifies mobile transactions into the following four types:

Atomic transactions:

Normal components and may be compensatable with atomic compensating dual steps. These are associated with the significant events {Begin, Commit, Abort} having the standard abort and commit properties.

Compensatable transactions:

Atomic transactions whose effects cannot be undone at all. When ready to commit the transaction delegates all operations to its parent. The parent has the responsibility to commit or abort the transaction later on.

A compensatable component of s is a component of s which can commit its operations

even before s commits, but if s subsequently aborts, the compensating transaction of the committed component must commit.

Reporting transactions:

- It can make its results available to the parent at any point of its execution. It could be a compensating or a non-compensating transaction. A reporting component ti can share its partial results with s. That is, a reporting component reports to s by delegating some of its results at any point during its execution.
- A reporting transaction periodically reports to other transactions by delegating some of its current results. Thus, reporting transactions are associated with the Report transaction primitive in addition to the Begin, Commit and Abort primitives.
  Begin is used to initiate a reporting transaction.
  Commit and Abort are used to terminate it.
  That reporting transactions satisfy the fundamental axioms.
  Begin event can be invoked at most once by a transaction
  Only an initiated transaction can commit or abort
  A reporting transaction cannot be committed after it has been aborted
  Only a transaction in execution can report
- It species that a transaction sees the current state of the objects in the database.
- It States that conflicts have to be considered against all in-progress operations performed by different transactions.
- It specifies that all objects upon which a reporting transaction invokes an operation are atomic objects. That is, they detect conflicts and induce the appropriate dependencies.
- It States that a transaction can commit only if it is not part of a cycle of CN relations developed
- It states that if an operation is committed on an object, the invoking transaction must commit.
- It states that if a transaction commits, all the operations invoked by the transaction are committed.
- It states that if an operation is committed on an object, the invoking transaction must commit.
- It states that if a transaction commits, all the operations invoked by the transaction are committed.
- Report Set contains the operations on the objects to be delegated.
- An abort-dependency of the reporting transaction on the receiving transaction is induced at the time of the report.

Co-transactions:

It behaves in a manner similar to the co-routine construct in programming languages. Co-transactions retain their current status across executions; hence they cannot be executed concurrently. These components are reporting transactions that behave like co-routines in which control is passed from one transaction to another at the time of sharing of the partial results. That

is, co-transactions are suspended at the time of delegation and they resume execution where they were previously suspended.

Reporting transaction vs. Co-transaction:

A reporting transaction reports its results to other transactions by delegating the results. A reporting transaction can have only one recipient at any given point of time. The changes made by a reporting transaction are made permanent only when the receiving transaction commits. If the receiving transaction aborts, the reporting transaction aborts as well.

A co-transaction, on the other hand, reports its results in a way similar to reporting transactions. But upon delegation, the transaction stops execution and is resumed from the point it left off. For any pair of co-transactions either both commit or both abort.

Kangaroo Transaction Model.

- This model introduced in [39] is based on the global transactions and the split transaction models, where transaction relocation is achieved by splitting the transaction at the point of hand-off. A mobile transaction (called Kangaroo transaction) is considered a global transaction in a multi database environment.
- A Kangaroo transaction (KT) is a global transaction that consists of a set of Joey Transactions (JT). A IT is associated with the base station or the cell in which it executes. When the mobile unit moves to a new cell, the IT in the previous cell is split, and one of the ITs is moved to the current cell of the mobile unit. Each IT may consist of a set of local and global transactions. The model is built upon the existing databases. The transactions are micromanaged by the individual database transaction managers.
- A Joey Transaction should terminate in an abort, commit, or a split. For a KT to be successful, the last JT in the order of execution should end in a commit or abort, whereas all other JTs should be split. Based on the ability to compensate the split transaction component, a KT can be executed as a whole atomic transaction or in a relaxed mode where only component transactions are executed atomically.

Kangaroo transactions:

- We introduce our mobile transaction model which we call Kangaroo Transactions. Our model is built on traditional transactions which are a sequence of operations executed under the control of one DBMS. Figure 1 shows the basic structure. Here three operations (op11, op12, and op13) are performed as part of the transaction.LT is used to identify the traditional transaction as it is executed as a Local Transaction to some DBMS.
- The operations performed are the normal read, write, begin transaction, abort transaction, and commit transaction. The first operation (op11) must be a begin transaction while the last (op13) must be either a commit or abort.
- Our view of global transactions in multi database systems is somewhat broader than that which is often assumed. We actually consider two types of global transactions. The limited view of global transactions is shown in figure 2(a).Notice that the Global Transaction root (GT) is composed of sub transactions which can be viewed as Local Transaction (LT) to some existing DBMS. The local transactions are often called sub

transaction (or Global Sub Transaction, GST) of the GT. Each of these can in turn be viewed as consisting of a sequence of operations. Figure 2(b) takes the more general view of global transactions.

- In this case the sub transactions may themselves be global transactions to another multi database system. So we would have (for this figure) operations underneath LT1 and LT4. UnderneathGT2 and GT3 would be other LTs and GTs. This view of global transactions gives a recursive definition based on the limiting bottom view of local transactions.

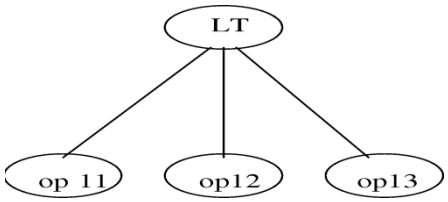


Figure 1: Basic structure

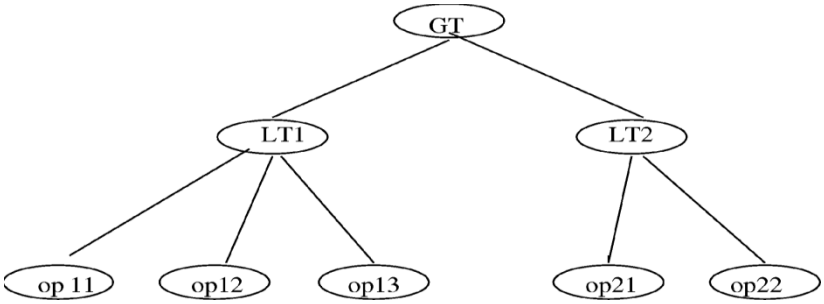


Fig 2: (a) Limited Global Transaction view

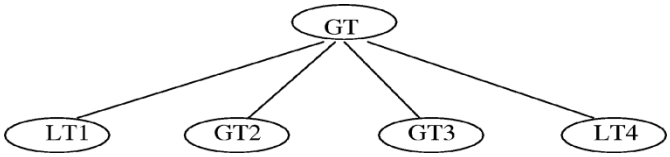


Fig2: (b) Global Transaction view

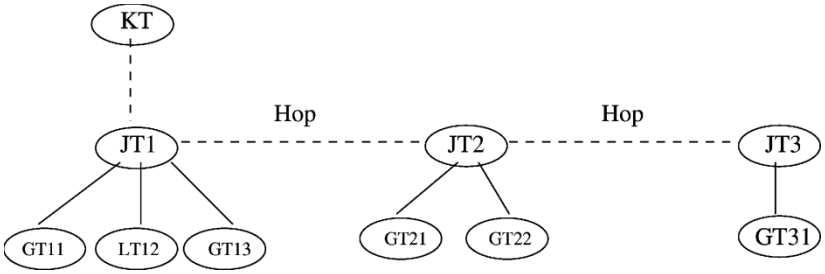


Fig 3: Basic structure of a Kangaroo Transaction

Introducing kangaroo transactions:

- Global transactions serve as the basis upon which we define our mobile transactions. Global transactions alone, however, do not capture the "hopping" nature of mobile transactions. Based on the hopping property, we call our model of mobile transactions Kangaroo Transactions (KT) figure 3 shows the basic structure of a Kangaroo Transaction.
- When a transaction request is made by a mobile unit the DAA at the associated base station creates a mobile transaction to realize this request. A Kangaroo Transaction ID (KTID) is created to identify the transaction. We define a KTID as follows:

KTID = Base Station ID + Sequence Number

where the base station ID is unique, the sequence number is unique at a base station, and + is a string catenation operation .

- Each sub transaction represents the unit of execution at one base station and is called a Joey Transaction (JT). We define a Pouch to be the sequence of global and local transactions which are executed under a given Kangaroo Transaction.
- The origination base station initially creates a JT for its execution. The only difference between a JT and a GT is that the JT is part of a KT and that it must be coordinated by a DAA at some base station site.
- When the mobile unit hops from one cell to another, the control of the KT changes to a new DAA at another base station. The DAA at the new base station site creates a new JT (as part of the handoff process). It is assumed that JTs are simply assigned identification numbers in sequence. Thus a Joey Transaction ID (JTID) consists of the KTID + Sequence Number. This creation of a new JT is accomplished by a split operation. The old JT is thus committed independently of the new JT.
- In figure 3, JT1 is committed independently from JT2 and JT3. At any time, however, the failure of a JT may cause the entire KT to be undone. This is only accomplished by compensating any previously completed JTs as the autonomy of the local DBMSs must be assured. To manage the KT execution and recovery, a doubly linked list is maintained between the base station sites involved in executing a Kangaroo Transaction. Control information about a JT is identified by its JTID.
- To complete a partially completed KT, this linked list is traversed in a forward manner starting at the originating base station site. Thus to restart an interrupted transaction, the user must be able to provide the starting site (base station) for the transaction. To undo a KT the linked list is traversed in a backward manner starting at the current JT base station site.

- There are two different processing modes for Kangaroo Transactions:
- Compensating Model and Split Mode. When a KT executes under the Compensating mode, the failure of any JT causes the current JT and any preceding or following JTs to be undone. Previously committed JTs will have to be compensated for.
- Operating in this mode re-quires that the user (or source system) provide information needed to create compensating transactions. This includes information that the JT is compensatable in the first place. Deciding whether a JT is compensatable or not is a difficult problem. Not only does the JT itself need to be compensatable, but the source system should also be able to guarantee the successful commitment of the compensating transaction. The split mode is the default mode.
- In this mode, when a JT fails no new global or local transactions are requested as part of the KT.
- However, the decision as to commit or abort currently executing ones is, of course, left up to the component DBMSs.
- Previously committed JTs will not be compensated for. Neither the Compensating nor Split modes guarantee serializability of the kangaroo transactions. Although Compensating mode ensures atomicity, isolation may be violated (thus violating the ACID principle) because locks are obtained and released at the local transaction level. With the Compensating mode, however, Joey sub transactions are serializable.
- Figure 4 shows the relationship between movement of a mobile unit between cells and the corresponding Kangaroo Transaction. Here we assume that when the transaction is started, the mobile unit is in Cell 1 which is associated with Base Station 1.

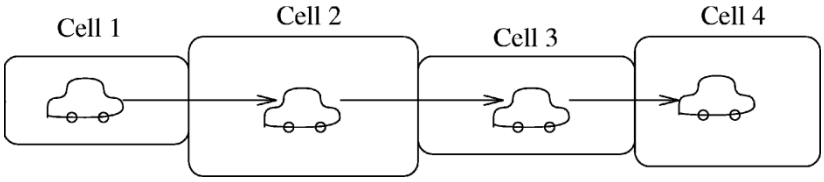


Fig 4 : (a) Movement of Mobile Unit through Cells

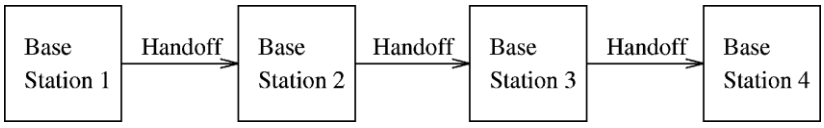


Fig 4 : (b) Hopping from Base Station to Base Station

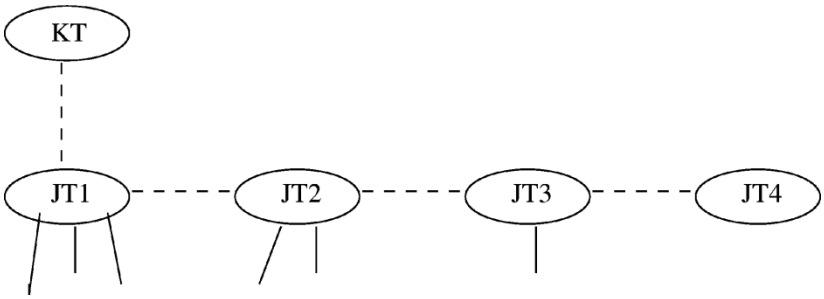


Fig 4 : (c) Kangaroo Transaction

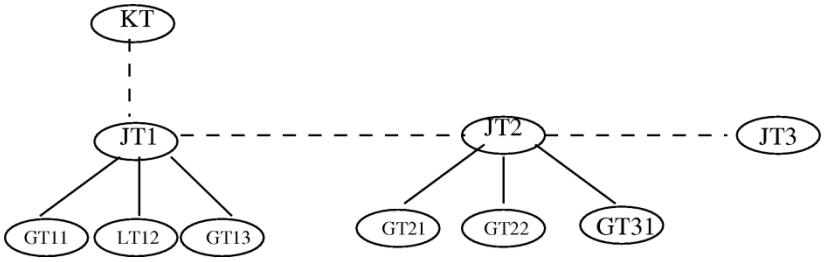


Fig 5 : (a)One possible Kangaroo Transaction for Adjustment Transaction

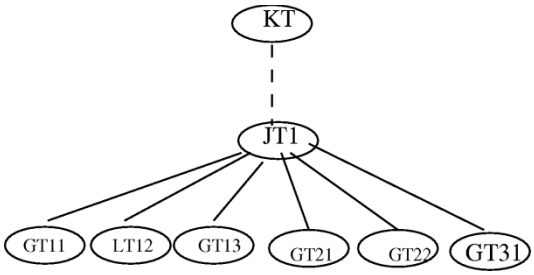


Fig 5 : (b) An equivalent Kangaroo Transaction

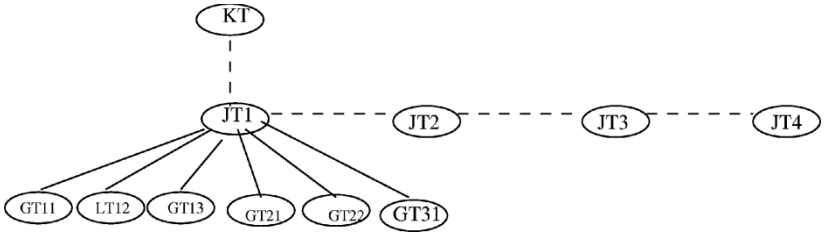


Fig 5 : (c) Another equivalent Kangaroo Transaction

- At this time, the DAA at this Base Station created a new Kangaroo Transaction and immediately created a Joey Transaction. When the mobile unit moves to Cell 2, a handoff is performed. As part of this handoff, the KT is split into two transactions. The first part of this transaction is the sub transactions under JT1, the remainder (at this time) will be part of JT2. Similarly, when the mobile unit moves into Cell 3 and Cell 4, handoffs occur and new Joey Transactions are created via a split operation. Note that this process is dynamic.
- A new Joey is created only when a hop between cells occurs: no hop – no Joey. The same transaction requested at two different times could have different structures. We illustrate this fact in figure 5. Figure 5(b) represents a short lived or slow mobility type of transaction, where as figure 5(c) shows a long lived or fast mobility transaction. This figure shows three different transactions which are equivalent to that in figure 3.
- Even though the structure of each is different in terms of the number of Joeys and the format of the Joey, the underlying set of global and local transactions of each is the same. We thus say that two Kangaroo Transactions are Equivalent if they have the same pouch.
- We conclude this section by describing in more detail the manner in which Joeys are created. The Joeys are created by a split operation. As part of the handoff procedure, a split operation is always performed.
- When looking at figure 4(b) a split is first performed during the hop from Base Station 1 to Base Station 2. The KT is then split into two sub transactions: JT1 and JT2. We assume that the sub transactions of the mobile transaction (that is the global and local transactions within it) are executed in sequence. The user does not request the next sub transaction until the previous one is completed. Thus the local and global trans-actions under JT1 will all occur before those in JT2. This guarantees that JT1 precedes JT2 in serializable order.
- Infact, we assume that no sub transactions under JT2 will be created until the currently executing one in JT1 is commit-ted. Notice that when JT2 is created there will probably be a local or global transaction under JT1 which is currently being executed. The JT2, however, must be created when the handoff occurs. The situation shown in figure 5(c) shows that JT4 has been created but no sub transactions yet exist.

A formal definition for kangaroo transactions:

In the following, we more formally define Kangaroo Transactions. They are defined recursively with the basic building block being a local transaction defined for a DBMS. Our definitions follow that found in [2] as a starting point and building block.

Definition 1: A Local Transaction LT is a sequence of read (ri) and write (wi) operations ending in either a commit (ci) or abort (ai) operation.

Definition 2: A Global Transaction GT is any sequence of global transactions Gj and local transactions Lj.

Definition 3: A Joey Transaction JT is a sequence of zero or more global transactions Gk and local transactions Lk followed by either a commit ck, abort ak, or split sk operation.

Note that this definition for JT ensures that the GTs and LTs within it represent a sequence. As stated earlier, we assume that all operations of one Joey are executed prior to those of the next.

Definition 4: A Kangaroo Transaction KT is a sequence of one or more Joey Transactions Jl. The last Joey Transaction must end in a commit cl or abort al. All Joey Transactions other than the last one must end in a split sl. The Kangaroo Transaction captures the movement behavior of the mobile transaction by forcing all joeys but the last to end in a split.

Definition 5: The sequence of local and global transactions which belong to a Kangaroo Transaction is called its Pouch.

Definition 6: Two Kangaroo Transactions are said to be Equivalent Kangaroo Transactions if they have exactly the same pouch.

Mobile transaction manager data structures:

- The functions of the MTM are those related to managing a mobile transaction. The primary data structure at each site which is used to do this is the transaction status table (see table 1).
- Each base station maintains a local log (see table 2) into which the MTM writes records needed for recovery purposes. Unlike DBMS and GDBS systems, this log contains no records dealing with recovering (undo or redo) database updates. The log however is stored in stable storage in the base station. Most of these records are related to the transaction status entries.
- During handoff processing the log buffer must be flushed to ensure recoverability if a failure occurs during the handoff process. When (after) a Kangaroo Transaction is started a BTKT (Begin Transaction KT) record is written to the log.
- During handoff processing (before), an HOKT (HandOff KT) record is written into the originator's (Base Station requesting handoff) log while a CTKT (Continuing KT) record is written into the destination's (Base Station being handed off to) log (after). Joeys are documented in the log with a Begin (BTJT) record and a Commit (ETJT) record. BTJT records are linked together in reverse order of creation while ETJT records are linked together in forward order of creation.
- Sub transactions within a joey have begin (BTST) and end (ETST) records. BTST records contain the actual local/global transaction requested and the compensating transaction if the KT is compensatable.

Kangaroo transaction processing:

The flow of control of processing Kangaroo Transactions by the MTM can be described as follows:

When a mobile unit issues a Kangaroo Transaction, the corresponding DAA passes the transaction to its MTM to generate a unique identifier (KTID) and creates an entry in the transaction status table. The MTM also creates the first Joey Transaction to execute locally in its communication cell. At the end of this setup, a BTKT record is written into the MTM transaction log.

The creation of a Joey Transaction (be it the first or otherwise) is also done by the MTM and involves generating a unique JTID and creating an entry in the transaction status table.

A BTJT record is then written into the log. Finally a JT entry is written into the transaction status table.

KT TRANSACTION STATUS TABLE ENTRIES

Record type	Attribute	Description
KT	KTID	ID for KT
	Mode	Split or Compensating
	Joey Count	Count of number of active Joeys in this KT
	Status	Active, Committing Aborting
	First JTID	Pointer to first JT status record for this KT
JT	JTID	ID for JT
	Next JTID	Pointer to next JT status record for this KT
	Prior JTID	Pointer to prev JT status record for this KT
	Status	Active, Commit, or Abort
	STList	List of local and global transactions ST
	Compensatable	Yes/No
ST	STID	ID for ST
	Status	Active, Commit, or Abort
	Request	GT or LT requested
	Compensatable	Yes/No
	CompTR	Compensating transaction

Clustering Model management.

- This model described a fully distributed system. The database is divided into clusters. A cluster defines a set of mutually consistent data. Bounded inconsistencies are allowed to exist between clusters. These inconsistencies are finally reconciled by merging the clusters. The model is based on the open nested transaction model, extended for mobile computing.
- A transaction submitted from a mobile host is composed of a set of weak and strict transactions. Transaction proxies are used to mirror the transactions on individual machines as they are relocated from one machine to another.
- A cluster is defined as a unit of consistency in that all data items inside a cluster are required to be fully consistent, while data items residing in different clusters may exhibit bounded inconsistency.

- Clusters can be defined either statically or dynamically. A wide set of parameters can be used for defining clusters. This could include the physical location of data, data semantics, and user definitions. Consistency between clusters can be defined by an m-degree relation, and the clusters are said to be m-degree consistent. The m-degree relation can be used to define the amount of deviation allowed between clusters. In this model, a mobile transaction is decomposed into a set of weak and strict transactions.
- The decomposition is done based on the consistency requirement. The read and write operations are also classified as weak and strict. The weak operations are allowed to access only data elements belonging to the same cluster, whereas strict operations are allowed database-wide access.
- For every data item, two copies can be maintained-one of them strict and the other weak. As mentioned above, a weak operation can access only the local copies of a data item. Weak operations are initially committed in their local clusters. When the clusters are finally merged, they are once again committed across the clusters.

Introduction:

- Cluster computing is an important element in mainstream computing. In recent years, cluster computers have emerged as the leaders in high performance computing. Cluster computing harnesses the combined computing power of multiple microprocessors in a parallel configuration.
- Cluster computers are a set of commodity PC's dedicated to a network designed to capture their cumulative processing power for running parallel-processing applications. Clustered computers are specifically designed to take large programs and sets of data and subdivide them into component parts, thereby allowing the individual nodes of the cluster to process their own individual chunks of the program.
- A Cluster is a group of loosely coupled computers that work together closely so that in many respects they can be viewed as though they are a single computer. Clusters are commonly, but not always, connected through fast local area networks.
- Clusters are usually deployed to improve speed and/or reliability over that provided by a single computer, while typically being much more cost-effective than single computers of comparable speed or reliability.
- Cluster can be categorized into three forms. High-Availability (HA) clusters are implemented primarily for the purpose of improving the availability of services which the cluster provides.
- They operate by having redundant nodes, which are then used to provide service when system components fail. The most common size for an HA cluster is two nodes, which is the minimum required to provide redundancy. HA cluster implementations attempt to manage the redundancy inherent in a cluster to eliminate single points of failure.

Cluster characteristics:

A basic cluster has the following characteristics:

- Multiple computing nodes,
    - low cost
    - a fully functioning computer with its own memory, CPU, possibly storage
    - own instance of operating system
- computing nodes are connected by interconnects
    - typically low cost, high bandwidth and low latency
- permanent, high performance data storage
- a resource manager to distribute and schedule jobs
- the middleware that allows the computers act as a distributed or parallel system
- parallel applications designed to run on it

Components of a Cluster:

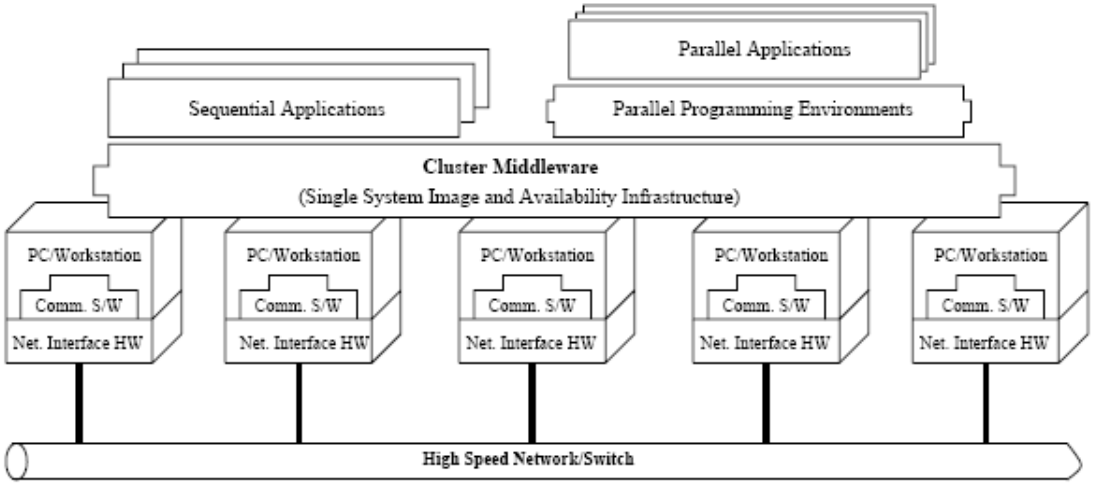


Figure 3 : An typical cluster architecture.

The following are the components of cluster computers:

- Multiple computers (computing nodes)
- Operating system of the nodes
- High performance interconnect network and fast communication protocols
    - With high bandwidth
    - Low latency

    Examples:

        - Myricom—1.28 Gbps in each direction
        - IEEE SCI latency under 2.5 microseconds, 3.2 Gbps each direction (ring or torus topology)
        - Ethernet-star topology

    In most cases limitation is the server's internal PCI bus system.

- Cluster Middleware
    - To support Single System Image (SSI) and System Availability Infrastructure
    - Resource management and scheduling software
        - Initial installation
        - Administration
        - Scheduling
        - Allocation of hardware
        - Allocation software components
- Parallel programming environments and tools
    - Compilers
    - Parallel Virtual Machine (PVM)
    - Message Passing Interface (MPI)
- Applications
    - Sequential
    - Parallel or Distributed

In a high performance computing cluster architecture, generally there are one or master nodes and one or more compute nodes interconnected by a network. The master node acts as a network file system server and a gateway to the outside world.

Types of Clusters:

We can classify clusters according to usage requirements

1. High Performance and High Throughput Clusters: They are used for applications which require high computing capability.
2. High Availability Clusters: The aim is to keep the overall services of the cluster available as much as possible, considering the fail possibility of each hardware of software. They provide redundant services across multiple systems, to overcome loss of service. If a node fails, others pick up the service to keep the system environment consistent, from the point of view of the user. The switching over should take very short time. A subset of this type is the load balancing clusters. They are usually used for business needs. The aim is to share processing load as evenly as possible. No single parallel program that runs across those nodes. Each node is independent, running separate software. There should be a central node balancing server.

Clusters can be classified according to the node type as homogeneous clusters and heterogeneous clusters.

1. Homogeneous clusters: In homogeneous clusters all nodes have similar properties. Each node is much like any other. Amount of memory and interconnects are similar.
2. Heterogeneous clusters: Nodes have different characteristics, in the sense of memory and interconnect performance.

Clusters may be classified according to the hierarchy they inherit.

1. Single level (single-tier) clusters: There is no hierarchy of nodes is defined. Any node may be used for any purpose. The main advantage of the single tier cluster is its simplicity. The main disadvantage is its limit to be expanded. [9]
2. Multi level (multi-tier) clusters: There is a hierarchy between nodes. There are node sets, where each set has a specialized function.

Benefits of clusters:

Benefits and reasons for popularity of clusters can be listed as follows:

- No expensive and long development projects. Building clusters is easy, compared to building a dedicated supercomputer.
- Price performance benefit: Highly available COTS products are used.
- Flexibility of configuration: Number of nodes, nodes' performance, inter-connection topology can be upgraded. System can be modified without loss of prior work. Two types of scaling can be defined.
    - Scale up: Increasing the throughput of each computing node.
    - Scale out: Increase the number of computing nodes. Requires efficient i/o between nodes and cost effective management of large number of nodes.

Efficiency of a cluster:

Cluster throughout is a function of the following:

1. CPUs: Total number and speed of CPU's
2. Inter-Process Communication: Efficiency of the inter-process communication between the computing nodes
3. Storage I/O: Frequency and size of input data reads and output data writes
4. Job Scheduling: Efficiency of the scheduling

Cluster computing:

Cluster computing is the technique of linking two or more computers into a local area network in order to take the advantage of parallel processing. In this technique mainly one or two computers will act as front end servers and distribute the task to different nodes available in the network.

Advantages of computer cluster of cluster computing:

- Availability: If one big Main Frame failed, entire work will be stopped, but if you adopt cluster computing if one node has been failed the task can be transferred to some other node in the network.
- Scalability: you can easily extend the processing capability of the network by just adding a new node.

Basic architecture of cluster computing:

There are many cluster configurations, but a simple architecture is shown in Fig1.

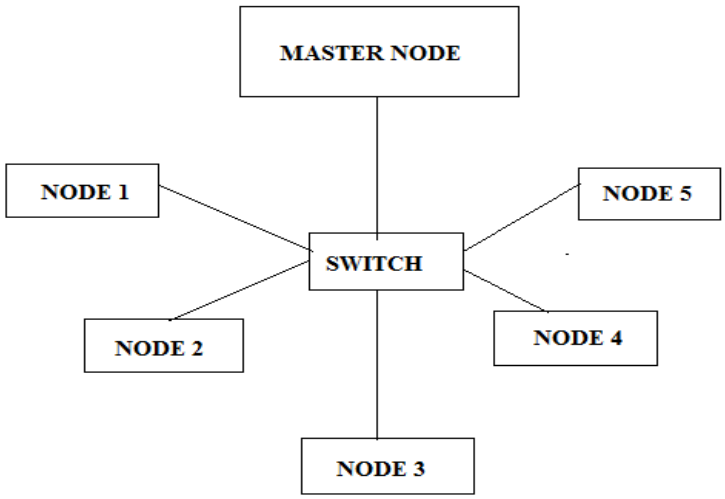


Fig. 1: Basic architecture of cluster computing

- In a typical cluster, the application runs on a Master node. However, the computational work is split up and parsed out to be done by the multiple nodes in the cluster.
- In this way, cluster is better equipped to handle larger amounts of data and complex problems than otherwise possible on a stand-alone machine. Some of the main modules related to cluster configuration are: Building a Cluster, System Administration, Hardware Management and Software Platform Maintenance. After the cluster is constructed, it requires an effective system administration to remain useful. Maintenance and administration of a cluster are similar to those of a LAN.
- Two major domains of work explored in this area are: hardware management and software platform maintenance. The main component of Hardware management is network management which can be divided into two major areas: cabling and topology. Every machine in a cluster must be able to work with the other machines. Maintaining the software on a cluster consists of administrative work multiplied by 'n' nodes - each of which is potentially dependent on other nodes.

Design of cluster computing:

- The main aim is to provide a flexible framework for Cluster Computing. The framework so used consists of three parts: Personal Computers (PC), high speed communication network, distributed applications as shown in Fig. 2.

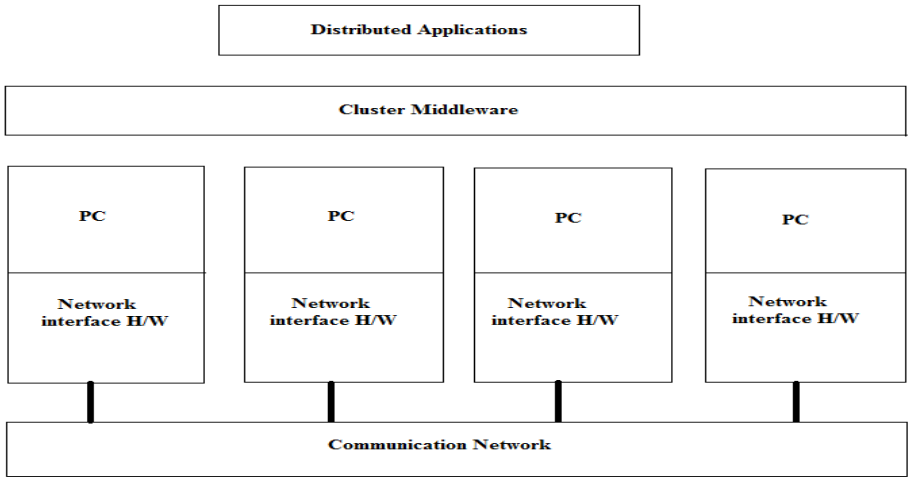


Fig. 2: Framework for Cluster Computing

- PC are connected to the network using standard Ethernet Network Interface Card (NIC).Cluster middleware is implemented in Java so that middleware can provide the Single System Image of the cluster to any computer with different OS platforms once the Java virtual machine(JVM) is installed.
- JVM makes it easier to implement, migrate and execute the mobile code at remote computer in the cluster. The user is guided through the creation and management of cluster via a graphical user interface. It frees the user from identifying the network topology of the framework of cluster. The framework has been designed in such a way that incremental changes to it can easily enhance the generality and usability of cluster.

Parallel processing using cluster computing:

- Parallel processing mainly involves concurrent use of multiple processors to process data. Significant development in Network technology is paving a way for parallel processing. Cluster computing implements MIMD (Multiple Instruction Multiple Data Stream) model of Flynn's classification of computer architecture using general purpose processors or multi computers.
- Clusters are also suitable than special parallel computers for the execution of parallel applications as they can easily integrate into existing networks. By sharing the computers of owner-users, which are normally not accessible in a non-dedicated cluster, parallel applications can gain extra processing power to perform CPU-hungry computations.
- On the other hand, owner users of their computers could suffer from a slight degradation of the execution performance. The degradation of the CPU services trends to be insignificant when the workload of the computers move towards I/O-bound applications and the number of owner-users is large in the cluster.

Issues in mobile clusters and parallel computing on mobile clusters:

- There has been an increasing interest in the use of clusters of workstations connected together by high-speed networks for solving large computation intensive problems.
- The trend is mainly driven by the cost-effectiveness of such systems as compared to large multiprocessor systems with tightly coupled processors and memories. However, recent proliferation of mobile devices and advancement in wireless connectivity has made parallel computing on mobile clusters a feasible proposal.
- Mobile devices can be part of the cluster playing several unique roles. They can be used as a frontend to the cluster functionality, such as submitting a job, managing processes, or viewing statistics. In case of a MH that has very poor computing power, the device must be able to utilize the cluster seamlessly to access the computational power. However, the MH can also be a contributor of computing power to the cluster, as in case of devices such as laptops that have a substantial amount of computing power equal to their static counter parts.
- Distributing computing power in a cluster consisting of a network of heterogeneous computing devices represents a very complex task. However, it becomes even complicated when mobile devices are also a part of it. There are several key issues that distinguish parallel computing on mobile clusters from that of the traditional workstation clusters, namely

    - Asymmetry in connectivity,
    - Mobility of nodes ,
    - Disconnectivity of mobile nodes ,
    - Timeliness issue,
    - Changing loads on the participating nodes,
    - Changing node availability on the network,
    - Difference in computing capability and memory availability,
    - Heterogeneity in architecture and operating systems.

Asymmetry in connectivity:

- The traditional cluster computing models do not face the problem of heterogeneity in the network connection as the entire set of workstations that are participating in the clusters are connected only by the wired network.
- Wireless networks deliver much lower bandwidth than wired networks and have higher error rates.
- Mobile devices are characterized by high variation in the network bandwidth that can shift from one to four orders of magnitude, depending on whether it is a static host or a mobile host and on the type of connection used at its current cell. Thus, the programming model must be able to distinguish among the types of connectivity and provide flexibility for easy variation of the grain size of the task to account for the variations in bandwidth.

However, these systems are suitable only for coarse grain level parallelism due to the communication overhead.

Mobility of nodes:

- Due to mobility of nodes, the notion of locality becomes important as users move from one cell to the other cell.
- The locality becomes important as the change in the mobile node's location means a change in the route to that node and consequent communication overhead. The ability to change locations while connected to the network increases the volatility of some of the information.
- Static data could become mobile in the context of mobile computing. As anode moves, nearby information servers get farther away and should be replaced by closer ones offering the same or more relevant contextual information.
- Traditional computers do not move, as result information that is reliant on location can be configured statically, such as the local DNS server or gateway, the available printers, and the time zone. A challenge for mobile computing is to define this information intelligently and supply means to locate configuration data appropriate to the present location.
- Mobile computing devices need to access more location related information than stationary computers if they are to serve as ubiquitous guides to a user's environment. As the mobile device moves and as the speed of motion changes, the quality of the network link and other available resources might change significantly. Thus, the system should be able to adjust according to the changing conditions.
- For example, when a MH that has taken the task moves from one cell to another, then the system still requires tracking these MHs.

Disconnectivity of mobile nodes:

The period of disconnectivity of nodes in static networks are usually treated as faults. However, in the context of mobile nodes, the disconnectivity may be due to roaming (and MH in an out of coverage area) or voluntary disconnection (doze mode) to save battery power.

Timeliness issue:

- Timeliness refers to the delay that is taken for the mobile device to regain its full state when it moves from one cell to the other or after reentering a coverage area after disconnection.
- Timeliness issue is an important issue especially in real-time systems. Whenever a mobile host moves from one cell to the other, it is associated with a handoff, to ensure that data structures related to the mobile host are also moved to the new connecting point (MSS). This involves exchange of several registration messages. This may cause some delay and it should be fast enough to avoid loss of message delivery. In addition to this, there is a possibility that the mobile host could move out of coverage after accepting the task for execution. These issues need to be addressed with respect to the mobile cluster model.

Changing loads on the participating nodes:

- When using workstations for executing parallel applications the concept of ownership is frequently present. Workstation owners do not want their machine to be overloaded by the execution of parallel applications, or they may want exclusive access to their machine when they are working.
- Reconfiguration mechanisms are thus required to balance the load among the nodes, and to allow parallel computations to coexist with other applications. To overcome these problems, some dynamic load balancing mechanisms are needed. There are differences in loads among the nodes due to multi-user environment, and when an application is run on heterogeneous cluster. In these cases, it is important to balance loads among the nodes to achieve sufficient performance.
- As static load balancing techniques would be insufficient, dynamic load balancing techniques based on runtime load information would be essential. This would be difficult for a programmer to perform load balancing explicitly for each environment/application, and automatic adaptation by the underlying runtime is indispensable. This gets aggravated when mobile devices are part of the cluster.

Changing node availability on the network

- In traditional distributed systems, nodes keep leaving and joining the system dynamically. The join and leave of nodes may be due to either node failure or link failure. However, the system must be smart enough to continue with the computation.
- The availability of node becomes fuzzier in a distributed mobile computing scenario as the movement of the nodes also affects the availability. It is possible that the node may enter an area which is not under the coverage area of any MSS.
- It is also possible that node availability is transient with respect to the execution of the program. While a mobile node is computing a subtask, it can go out of coverage and enter back into the coverage area before the completion of the execution of the program.

Difference in computing capability and memory availability

As each host may have different capabilities (such as memory) and different processing power, it is essential to allocate tasks to the nodes based on their capabilities and processing power. MHs may especially have lower computing power and memory in contrast to its static counterpart.

Heterogeneity in architecture and operating systems

- Although it is reasonable to assume that a new and standalone cluster system may be configured with a set of homogeneous nodes, there is a strong likelihood for upgraded clusters or networked clusters to have nodes with heterogeneous operating systems and architectures.
- In the operating system heterogeneity could be handled through distributed operating systems. However, it will be non-trivial to handle architectural heterogeneity, since the executable files are not compatible among architectures.

- The issues discussed in this section make parallel programming on mobile clusters difficult. With the issue of mobility and other constraints associated with mobile devices, the management of distribution at the programming level further hardens the task. The existing cluster computing models solve only a subset of these issues.

Isolation-Only Transactions

- The Coda file system at CMU provides an application-transparent file system for mobile clients by using file hoarding and optimistic concurrency control.
- A proxy logs all updates to the file system during disconnection and replays the log on reconnection. Automatic mechanisms for conflict resolution are provided for directories and files through the proxy and the file server. Hoarding is based on user-provided, prioritized list of files.
- Periodically, the proxy walks the cache to ensure that the highest priority files are present and consistent with the server. Coda provides Isolation-only Transactions (lOT) [75] to automatically detect read/write conflicts that could occur during disconnection. Unlike traditional transactions, it does not guarantee failure atomicity and only conditionally guarantees permanence.
- The SEER hoarding system [73] developed at UCLA is based on the Coda file system. It operates without user intervention by observing user activities and predicting future needs. It defines and uses a measure called "semantic distance" between files to determine how best to cluster files together in preparation for hoarding.
- The semantic difference between two files is based on the time elapsed between the events of opening the files, and on how many reference to other files occurs in between. SEER does not actually hoard files, but instead interfaces with Coda (and other replicated systems) to do the hoarding. SEER also detects hoard misses during disconnection.

What is an IOT?

- An IOT is a flat sequence of file access operations bracketed by begin _iot and end_iot. The execution of an IOT guarantees a set of properties that are specially tailored for optimistic replication and mobile Unix workstation environment.
- An IOT provides strong consistency guarantees depending on the system connectivity conditions. Unlike traditional transactions, it does not guarantee failure atomicity and only conditionally guarantees permanence.
- The IOT execution model is inspired by Kung and Robinson's optimistic concurrency control model, with a client's local cache effectively serving as the private workspace for transaction processing.
- When a transaction T is invoked by a user, its entire execution is performed on the user's client machine. Remote files are accessed through the client's local disk cache; and no partial result of the execution is visible on the servers.
- When T's execution is completed, it enters either the committed or the pending state depending on the connectivity condition. If T's execution does not contain any

partitioned file access (i.e., the client machine maintains a server connection for every file T has accessed), T is committed and its result is made visible on the servers.

- Otherwise, T enters the pending state waiting to be validated later. T's result is temporarily held within the client's local cache and is visible only to subsequent processes on the same client. When the relevant partitions are healed, T is validated according to the consistency criteria to be discussed in further sections. If the validation succeeds, T's result will be immediately reintegrated and committed to the servers.
- Otherwise, T enters the resolution state. When T is automatically or manually resolved, it will commit the new result to the servers. Figure 1 shows the complete IOT execution model from the user's viewpoint.

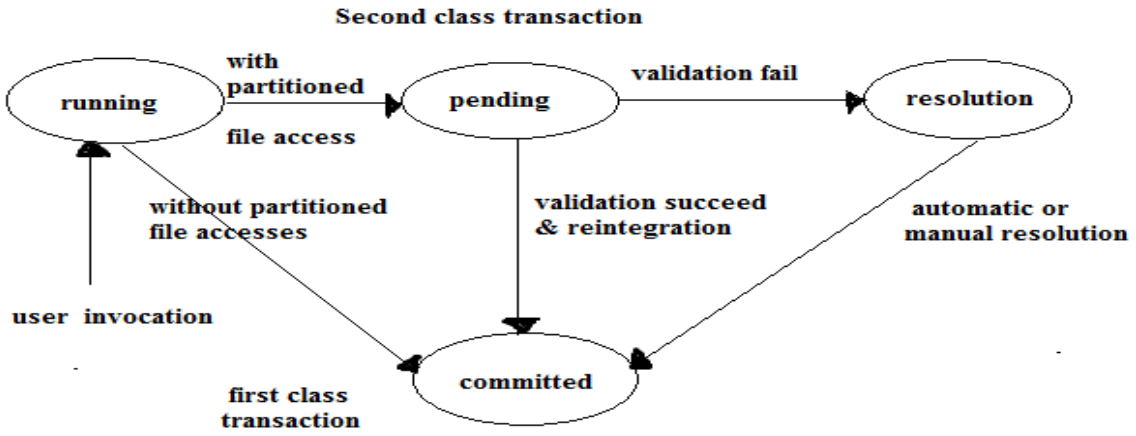


Figure 1: A State Transition Diagram for IOT Execution

Why Isolation Only?

- Lightweight operation and high efficiency are our key design goals. As a result, the IOT model does not provide the failure atomicity and permanence guarantees present in the traditional transaction model.
- Failure atomicity is not supported mainly because of high resource cost. A large amount of space is needed for undoing the effect of a transaction because it can access large objects and its execution can last long. Such cost is further magnified because space is a much more precious resource on mobile clients.
- Devoting too much space to the possible task of backing out transactions may cause denial of other valuable disconnected file services. Moreover, recent research has shown that the all-or-nothing property is not always desirable.
- The permanence guarantee of the traditional transaction model promises that once a transaction commits, its result will stay unchanged and can survive various system failures.

- One of the key and often unnoticed consequences of this property is that once a transaction makes its result visible to subsequent transactions, the result must not change until it is modified by some other transactions. In the IOT model, the result of a pending transaction is visible to subsequent transactions running on the same client. But this result is subject to change upon future validation.
- Therefore we can only offer a conditional form of the permanence guarantee. That is, the result of a transaction is permanent only when it does not contain partitioned file access, or it is successfully reintegrated or resolved.

IOT Consistency Guarantees:

- In order to maintain data consistency, the traditional transaction model provides the isolation property to make sure that transactions are executed as if they were isolated from each other. In serializability theory terms, the isolation property guarantees that the results of the interleaved execution of a set of transactions are equivalent to some serial execution of the same set of transactions[2].
- The IOT model offers substantially stronger consistency guarantees than the traditional transaction model. Transactions are classified into two categories: a first class transaction is one whose execution does not contain any partitioned file accesses. Otherwise, it is a second class transaction(see Figure 1).

Serializability(SR) for First Class Transactions

The execution of any first class transaction is guaranteed to be serializable with all committed transactions.

Local Serializability(LSR) for Second Class Transactions

The execution of any second class transaction is guaranteed to be serializable with other second class transactions executed on the same client.

Global Serializability (GSR) for Second Class Transactions

- One of the consistency criteria for validating a pending transaction T is that T must be globally serializable(GSR) with all committed transactions. It means that if T's result in the client's local cache were reintegrated to the servers as is, T would be SR with all committed transactions.GSR is significantly different from SR or LSR in that GSR cannot be enforced at transaction execution time.
- It can only be tested when the relevant partitions are healed. Therefore, as an integral part of the GSR guarantee we must specify what the system will do if the test fails. The IOT model provides the following automatic resolution options.

Re-executing the transaction

Successful re-execution of the transaction using the up-to-date server files is guaranteed to resolve the related inconsistencies. For example, this option can be used to automatically re-run make when the compilation results are inconsistent. It is our default option.

– Invoking the transaction's application specific resolver (ASR)

Sometimes it is more effective to resolve a transaction by using application-specific knowledge. The IOT model allows the transaction writer to attach an ASR to a transaction to be automatically invoked by the system. For example, non-serializable updates to an appointment calendar file can often be merged by an ASR as long as there are no time slot conflicts.

– Aborting the transaction

Simply aborting a non-GSR transaction will suffice to restore consistency. Suppose a transaction is executed on a disconnected client to compress a large file while the same task has already been done by someone else on the servers, aborting the transaction is an appropriate action in such a situation.

– Notifying the users

As a last resort, users can choose to manually resolve a non-GSR transaction. The IOT system will only mark its write-set as inaccessible and notify the users. If a transaction is used for editing the files of a co-authored paper on a disconnected laptop, this option is useful for coordinating possible concurrent updates.

Global Certification Order(GCO) for Second Class Transactions

- In certain situations, GSR alone is not adequate for voluntarily disconnected mobile clients. In the earlier example, suppose Joe ran make as a second class transaction TJ to build the new version of repair; and the librarylibresolve.ais updated by a first class transaction TL; and there are no other related file accesses.
- When Joe reconnects his Coda laptop to the servers, TJ will be admitted because it can be serialized before TL.
- To remedy this problem, we adopt a stronger consistency criterion called global certification order(GCO). GCO requires a pending transaction to be serializable not only with but also after all the committed transactions. GCO has the same set of resolution options as GSR.
- If Joe wants to make sure that his work done on an isolated laptop is compatible with the most recent system state, he can select GCO as the consistency criterion for transaction validation. Now TJ will be rejected because it can \not be serialized after TL.
- Joe can also use the default resolution option to let the system automatically re-run make to build an up-to-date version of repair.

Implementation Strategy:

- Because of the need for partitioned transaction execution, logging is the foundation for IOT implementation in Coda. Based on the properties of the Coda mobile computing environment, we choose and extend the transaction implementation technologies that offer the best engineering trade-off. For brevity, we only highlight the following issues that are critical to the overall transaction processing performance.

Concurrency Control for First Class Transactions

We chose the optimistic concurrency control (OCC) method to enforce SR for first class transactions. The main idea of OCC is trading transaction re-execution for global synchronization. This fits well with the scalable Coda architecture where client cycles are considered cheaper than server communication bandwidth. OCC is also capable of providing high throughput in the Coda environment because of low data contention. We extended the OCC scheme so that transaction history information can be utilized to process long-running transactions more efficiently.

- Transaction Validation for GSR

We apply Davidson's optimistic transaction model for GSR testing. This method builds a data structure called the precedence graph to represent the inter-dependency among transactions across partitions. GSR testing then becomes a matter of cycle detection in the corresponding precedence graph. We also extended the model so that GSR testing can be performed even when the transaction histories are truncated.

- Transaction Logging

Continuous transaction logging is needed on both servers and clients. Because log space is finite, transaction service will be unavailable to a client if its log space is exhausted. However, a server will truncate its recorded transaction history to reuse log space. Based on our preliminary observations and estimation, a modest amount of server log space(e.g. 40MB/server) may suffice for a typical working day. Fortunately, the GCO testing can be performed without using transaction histories.

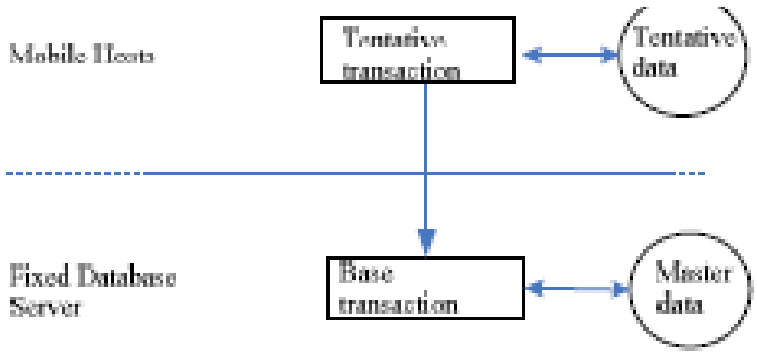
Two-tier Transaction Model.

- A two-tier replication scheme has been proposed whereby mobile disconnected applications are allowed to propose tentative update transactions. On connection, tentative transactions are applied to (re-processed at) the master data copy in the fixed network.
- At the re-processing stage, application semantics are used (such as finding commutative operations) to increase concurrency. To reduce re-processing costs that can be high in certain occasions, the work in uses a history-based approach. On reconnection, tentative

transactions, which are represented as histories, are merged with base transactions' histories. The merging process quickly identifies the set of tentative transactions that need to be backed out to resolve conflicts.

Description:

- Two – tier transaction model is a lazy replication mechanism which considers both transaction and replication approaches for mobile environment where MH are occasionally connected. This model proposed by Gray and also called Base Tentative model.
- Each object having master data copy and various replicated copy. Base transactions operate on master copy while tentative transactions access replicated copy version.
- The master copy has the most recent value received from the fixed host, which has not been yet processed by local transactions. The replicated copies have the most recent value due to local updates made by local transaction.
- Tentative transactions are local committed at mobile host in disconnected mode. After a disconnection execution, tentative transactions are re-executed taking into account their acceptance criterion at BS to reach the global consistency. This re-execution is the way to make local updates persistent.
- To execute the local transaction execution and concurrency control, this model requires a transaction manager on the mobile host. When tentative transactions (which are the re-execution as base transaction) fail, even by taking into account the acceptance criteria, then the tentative transactions are aborted and a message is returned to the user of the mobile node. This abort concerns only tentative transactions because local results are exclusively available for tentative transactions. Base transactions commit atomic commit protocol in connected mode at mobile host.



Consistency:

- To avoid application blocking at MH in disconnected mode, local availability of replicated consistent object is necessary. The consistency in Two-Tier transaction model is maintained by two versions: master and tentative.
- Both versions are located at MH, tentative version is used to support data evaluation in disconnected mode. The consistency of master copy must be sustained but sometimes it will contain old versions in disconnected mode.

- Consistency in master –copy is presented using one copy serializability method e.g. master copy. Tentative data copies are discarded at reconnection since they are completely refreshed from master copy.

Disconnection:

While the mobile hosts are disconnected from the database servers, tentative transactions are locally carried out based on replicated version of data objects. As the connection established those transactions are reprocessed and validated on the fixed hosts.

Semantic-based Nomadic Transaction Processing.

- The semantics-based mobile transaction processing scheme views mobile transactions as a concurrency and cache coherence problem. It introduces the concepts of fragmentable and reorderable objects to maximize concurrency and cache efficiency exploiting semantics of object operations. The model assumes the mobile transaction to be long-lived with unpredictable disconnections.
- Traditional definitions of concurrency and serializability are too restrictive for most operations. Commutatively of operations is an important property which allows concurrent operations on an object. If certain operations on an object are commutative, then the database server can schedule these operations in an arbitrary manner. Recovery also becomes quite simplified.
- Operations may be commutative either for all states or part of the states of the objects. The 110 values of the operations can be used to redefine serial dependencies of the operations. Though this may improve concurrency, it may require complex recovery mechanisms than normal schemes.
- Organization of the object can be used for selective caching of the object fragments, necessary for continuing the operation during the disconnected state. This approach reduces the demand on the limited wireless bandwidth and provides better utilization of the cache space available on the mobile host.
- Application semantics can also be utilized to define the "degree of inconsistency," "degree of isolation," and the "degree of transaction autonomy". Techniques like epsilon serializability and quasi copies can be used to specify allowable inconsistencies in the system.

This approach utilizes the object organization to split large and complex objects into smaller easily manageable pieces.

- The semantic information is utilized to obtain better granularity in caching and concurrency. These fragments are cached and/or operated upon by the mobile hosts and later merged back to form a whole object. A stationary server sends out the fragments of an object when requested by mobile units.
- The objects are fragmented by a split operation. The split is done using a selection criteria and a set of consistency conditions. The consistency conditions include the set of allowable operations on the object and the conditions of the possible object states.

- On completion of the transaction, the mobile hosts return the fragments to the server. These fragments are put together again by the merge operation at the server. If the fragments can be recombined in any order, then the objects are termed "reorderable" objects. Aggregate items, sets, and data structures like stacks and queues are examples of fragmentable objects.

UNIT V

Mobile Computing Models: Client Server model – Client/Proxy/Server Model – Disconnected Operation Model – Mobile Agent Model – Thin Client Model – Tools: Java, Brew, Windows CE,WAP, Sybian, and EPOC.

The Client/Server Model

- In this model neither the client nor the server are aware of the client (or server) mobility. The conventional client/server model is used without any modifications made in the application or the transport layer. The wireless media is transparent since it is handled in the data link layer.
- The mobility on the other hand, is made transparent by handling the variable client/server location through location-based routing in the network layer. Mobile IF is an example of a network protocol that hides the *CIS* mobility.
- The advantage of this model is its portability. The client or the server need not be changed in any way. Simply, the client (server) is ported to the mobile host, with a fixed address specified for the server (on the fixed network or on a mobile host).

The disadvantages of this model are listed below:

- The mobile client may suffer from a slow and unpredictable response time, especially when large server replies such as query results are transmitted without any considerations to the limited wireless bandwidth.
- The server caching strategy may not work properly in this model because the majority of the fixed network caching algorithms use call backs to invalidate the client cache. Most invalidation algorithms rely on the continuous availability of the client. Since the client could be disconnected or temporarily inaccessible (during hand-offs for example), the cache invalidation process could fail.

The client server model defines

- Which process may begin the interaction?
- Which process may answer?
- How error conditions may be managed
- Although an Internet system provides a basic communication service, the protocol software cannot initiate control with, or accept contact from, a remote computer.
- Of course, two application involved in a communication cannot both wait for a message to arrive. One application must actively initiate interaction while the other application passively waits.
- Most network applications use a form of communication known as the client –server paradigm.

- A server application waits passively for contact, while a client application initiates communication actively.

Process classification

- Client process, the process that requires a service
- Server process, the process that provides the required service
- The client requires a service; the server provides the service and makes available the results to the client

Client functions

In generally, client software:

• Is an arbitrary application program that becomes a client temporarily when remote access is needed, but also performs other computation locally.

• Is invoked locally by a user, and executes only for one session

• Runs locally on a user personal computer

• Actively initiates contact with a server

• Can access multiple services as needed, but actively contacts one remote server at a time.

• Does not require special hardware or a sophisticated operating system

Server functions

• Is a special purpose, privileged program dedicated to providing one service, but can handle multiple remote clients at the same time.

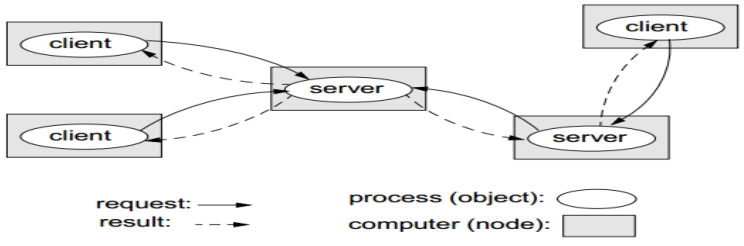
• Run on a shared computer (i.e. not a user's personal computer).

• Wait passively for contact from arbitrary remote clients

• Accepts contact from arbitrary clients, but offers a single service

• Requires powerful hardware and a sophisticated operating system.

- ❖ The client-server model is usually based on a simple request/reply protocol, implemented with Send/receive primitives or using remote procedure calls (RPC) or remote method invocation (RMI):
- ❖ The client sends a request (invocation) message to the server asking for some service;
- ❖ The server does the work and returns a result (e.g. the data requested) or an error code if the work could not be performed.
- ❖ A server can itself request services from other servers; thus, in this new relation, the server itself acts like a client.



The client/server model

The Client/Proxy/Server Model

- To overcome the shortcomings of the conventional client/server model, the client/proxy/server (C/P/S) model introduces a mobility-aware middle layer to mediate the interactions between the client and the server.
- The basic idea behind this model was introduced in the Mowgli architecture, which is to split the communication path between the client and the server into two parts by using a store-and-forward interceptor.
- The main advantage of this model is that the proxy allows the client and the server to be designed without any built-in mobility assumptions. The proxy assumes that the client is mobile and the server is in the fixed network. The result from the server is sent back to the proxy.
- The proxy filters the results according to the limitations of the wireless media and/or the client's mobile unit. The proxy may also store the filtered results until the client is connected. Examples of data filtering include: color and resolution reduction audio file removal, and file size reduction.
- The programming of the proxy involves knowledge of the mobile host hardware specifications. For example, a mobile host that does not have audio capability will benefit from audio file removal filtering. In addition to the mobile host, the mobile user profile can be useful in providing the proxy with user preferences such as no-images and no-colors.
- In support of the C/P/S computing model, C/P/S network architecture for reliable communication is introduced in the Mowgli architecture. In this architecture, the mobile host is provided with a specialized transport service, the Mowgli Data Channel Service (MDCS).
- It provides prioritized data channels with flow control between the mobile host and the base station. Existing TCP/IP protocols are used between the base station and a fixed host so that the protocol software in the fixed network remains unmodified. Two mediators, the Mowgli Agent and the Mowgli Proxy, which reside at each end of the wireless link, are used to provide functionality similar to that of TCP and UDP.

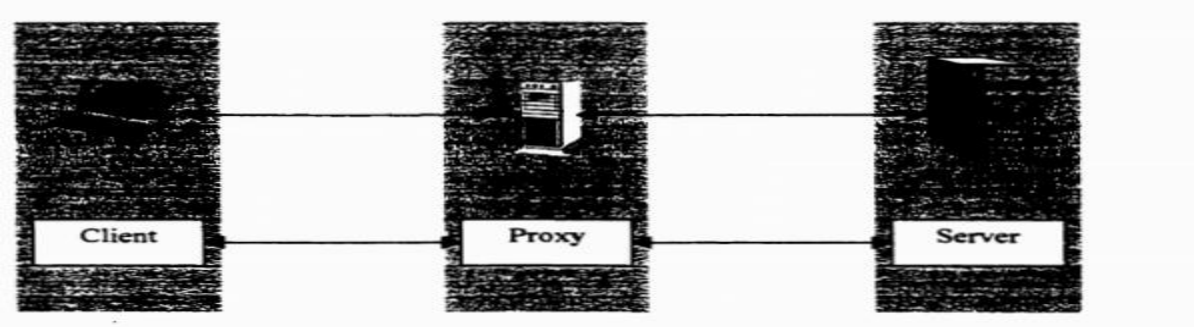


Figure 2.1 Client-Proxy-Server Model

- "A client-server application on a mobile device or desktop workstation provides some functionality to the end-user in conjunction with server(s) in the Internet.
- Examples are the WWW browser that retrieve documents from servers over the Internet, or clients that connect to FTP servers to upload or download file W. The Client-Server model is widely used and it seems to fit many applications. However, in wireless mobile environments, the typical client-server model is ill-suited.
- For instance, a mobile device, with limited capabilities, is not a preferable place for a large client program to do some heavy computational jobs. Also over a low bandwidth wireless link, it could be unacceptably slow for a client to download a big file from the server. Our design is based on an extension of this traditional client-server mode1 to a client- proxy-server model. Figure 2.1 shows the model and relevant components. The typical client program is separated into two parts running on mobile devices and proxies, respectively.
- The client provides the user interface and some part of the application logic, which usually is relatively mall. The proxy holds most part of the application logic, because it is, usually, a powerful machine. Proxies act like gateways for clients to communicate with servers, and hide the fact of the bandwidth of link and limitations of mobile devices to the servers. For servers, proxies are clients, while, for clients, they are servers.
- Existing proposals typically install a proxy that filters and/or compresses data for a specific application in order to help eliminating limitations introduced by mobile devices and the wireless link. This filter is either enabled or disabled depending on changes in the current environment.
- For example, in order to reduce the bandwidth usage, an MPEG player can download the file from the web server, decode streams at the proxy, compress the pix-map, and send the compressed results to the client. Which decompresses the result and displays the pictures.
- In our approach, proxies are connected via a wired network. Workload can be shifted not only been clients and proxies, but also between different proxies, depending on current time conditions. A proxy infrastructure is created to support client/user mobility. In the following chapters, we will talk about this in detail.

The Disconnected Operation Model

- Mobile clients may face wide variations in network conditions and local resource availability when accessing remote data. This is true in the *CIS* and the *C/P/S* models. Disconnected operations is a variation of the *CIS* model where, instead of working under the extreme case of weak-connectivity, the mobile client effectively switches to use a network of zero bandwidth and infinite latency.
- The operations that enable a client to continue accessing critical data during the dis-connection (switch off) period are called disconnected operations.
- The ability to operate when disconnected can be useful even when connectivity is available. For example, disconnected operation can extend battery life by avoiding wire-less transmission and reception. It allows radio silence to be maintained, a vital capability

in military applications. And, it is a viable fallback position when network characteristics degrade beyond usability. Voluntary disconnection can be treated as planned failures which can be anticipated and prepared.

The Mobile Agent Model

- Agents can be classified as static, mobile scripts, or mobile objects [42]. Static agents are those which execute just on a single site, either as a client or as a server. A static agent could be carrying out some activity like mail filtering. Mobile scripts are those that are downloaded from a server and executed on a client.
- Java applets, perl, or python scripts can be classified as mobile scripts. Mobile agents are mobile scripts with associated execution state information. A mobile agent could either be relocated along with the user, or it could be relocated during the execution of the agent.
- The relocation of the agent involves saving the state before initiating relocation and later restarting the mobile agent at the new location. The mobility of agents raises a large number of issues like security, authorization mechanisms, access mechanisms, and relocation mechanisms.

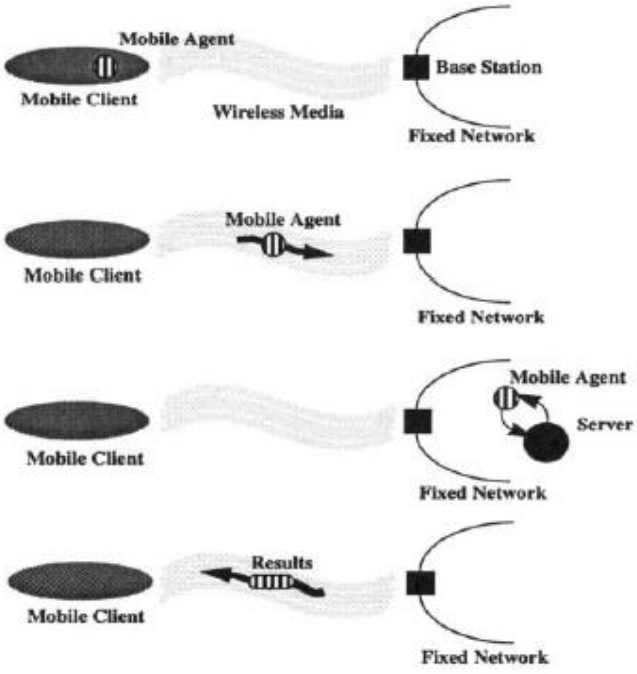


Figure 7.8 Mobile agents for mobile computing

- The mobile agent is an emerging new model that provides an alternative to the C/P/S model. A mobile agent is an active entity that is knowledgeable of both the limitations of the mobile environment and the mobile user.
- To access remote data, the mobile user sends a mobile agent on his behalf to the data source in the fixed network. The mobile agent is an execution context initially loaded

with the queries or data access requests. Once the agent moves to the data source (server), it acts as a local client to the server.

- Once the *CIS* interactions between the mobile agent and the server are completed, the agent "targets" the resulting data in preparation for transmitting the result to the mobile user. Such targeting includes filtering and transcoding actions such as color depth and resolution reduction and compression. The mobile agent paradigm is depicted in Figure 7.8.

Mobile agent technology

- Mobile Agent, namely, is a type of software agent, with the feature of autonomy, social ability, learning, and most important, mobility. Program that can migrate from system to system within a network environment.
- Agent decides when and where to move next .It performs some processing at each host. In mobile applications data may be organized as collections of objects, in which exchange between mobile and static hosts. Besides introducing stationary agents in the path between the mobile client and the server, mobile agents have also been used to accomplish tasks required by mobile.
- It reduces bandwidth usage Reduce total completion time and Reduce latency. Mobile agents should be able to execute on every machine in a network and the agent code should not have to be installed on every machine the agent could visit. Therefore Mobile Agents use mobile code systems like Java and the Java virtual machine where classes can be loaded at runtime over the network.
- The appeal of mobile agents is quite alluring - mobile agents roaming the Internet could search for information, find us great deals on goods and services, and interact with other agents that also roam networks (and meet in a gathering place) or remain bound to a particular machine.
- Significant research and development into mobile agency has been conducted in recent years , and there are many mobile agent architectures available today . However, mobile agency has failed to become a sweeping force of change, and now faces competition in the form of message passing and remote procedure call (RPC) technologies.

IMPLICATION OF MOBILE AGENTS

A. Bandwidth conservation

- One of the goals of mobile agency is to conserve bandwidth, by placing an agent directly at the point of information, rather than sending dozens or even hundreds of queries across the network For bandwidth to be conserved, the bandwidth consumed by sending across a mobile agent, and waiting for its results, must be less than that of a series of queries sent via a messaging or RPC system.
- Some electronic commerce models suggest that mobile agents would be sent out to multiple sites, perhaps to negotiate low prices with vendors This sort of activity has the potential to result in an incredible explosion of bandwidth consumption.

- Indeed, searching is a task that many people would like to see mobile agents performing. Currently, a small number of indexing agents collect information for search engines, while millions of queries are made by users.
- Imagine if the same number of queries were made instead by mobile agents that traveled across the network to sites. Two scenarios are possible. Either a much larger amount of bandwidth will be consumed, or a much lesser amount of bandwidth will be consumed as users receive more accurate search results because their agents have more control over the search process.
- Instinct suggests, however, that a simple keyword query entered via a web browser will consume less resources and bandwidth than sending an agent with specialized searching algorithms across the network.

B. Delegate tasks to agents when not connected

- The Internet, as it stands today , is made up of many millions of computers, some of which are permanently connected but the majority of which connect via dial-up modem connections for short periods of times. If you could delegate tasks to mobile agents, that would roam the network for you while not connected. This goal would be extremely desirable in the short term, until permanent connections became more prevalent. Here mobile agents may have found a sound market. This market could potentially be profitable, for the mobile agent technology vendors and the agent hosts that allow offline usage of their network. Delegation of tasks to mobile agents could also be used as a form of load sharing in distributed systems. Agents could perform tasks on remote systems, moving from system to system as required to balance the load. Mobile agency also gives greater flexibility, because new tasks and new code can be added to the system without the need for a fixed code base.

C. Mobile agents enable new types of interaction

- The ability of mobile agents to fragment themselves into many pieces that travel to different points across the network sounds promising. It might enable new forms of interaction, such as negotiating agents that travel to vendors seeking the best deal, or meeting places where agents can "get together" and communicate. The attraction of mobile agents for electronic commerce is great, and it might make sense to deploy mobile agents for electronic commerce. However, such uses could also be accomplished by message passing, or direct communication using application protocols like HTTP. Mobile agency is promising, but it is not the only mechanism for new uses of software agents.

D .Agent privacy

- If mobile agents were to become commonplace, serious privacy concerns would be raised. Agent hosts could also monitor the actions of agents, and create consumer profiles. In which an agent searched, could reveal information about its owner. When individuals query a

The Thin Client Model

- The thin client computing model attempts to offload most application logic and functionality from mobile clients to stationary servers. In this model, applications in stationary servers are usually mobile-aware and optimized for mobile client devices. This model is especially suitable for dumb terminal or small PDA applications.
- The thin client architecture from CITRIX Corporation allows a variety of remote computers, regardless of their platform, to connect to a Windows NT terminal server to remotely access a powerful desktop and its applications.
- A server called MetaFrame runs under Windows NT in the desktop machine and communicates with the thin clients executing at the remote computers using the Independent Computing Architecture protocol (ICA).
- The ICA client and the MetaFrame server collaborate to display the virtual desktop on the remote computer screen. They also collaborate to process mouse and keyboard events and to execute programs and view data stored at the server. All executions are remote and none take place at the client portable computer. The research work described in examines extensions to CITRIX thin client architecture so that it is optimized in the wireless environment. The work pointed out that bandwidth limitation is not as detrimental to the thin client performance as network latency. This is because the thin clients' use of bandwidth is limited.

THIN CLIENTS

Terms

***Fat Client*:** A full PC of any size which includes RAM, hard drive, CD drive, running a full operating system and having applications local.

***Thin Client*:** A miniature computer composed mainly of RAM which may run a slimmed down operating system but relies on connection to a server for its operating system experience and applications.

***Terminal Server*:** Centralized server computer that holds the operating system and applications for use by the thin clients.

***Terminal Server Cluster*:** As one Terminal Server can support few full-working clients, it is recommended to have many in a load balanced cluster. This allows client connections to be shared amongst many servers, relieving the stress on any one server.

***Soft Grid Application Virtualization*:** A method of allowing software to be streamed from a server to a local computer. The software is completely self-contained and can be set to be upgraded, expire, etc without affecting the underlying operating system.

Thin Client Architecture

- Thin client architecture takes the client/server relationship of UNIX terminals of the 80s and combines it with the graphical interface of Windows.
- The user workstations can be fat clients (normal PCs) and take advantage of centralized software packages, however using thin clients (tiny computers working entirely on RAM instead of hard

disks) to connect to the servers reduces maintenance of the various components of a fat client (CD/DVD drives, hard disks, etc). The server cluster provides the operating system interface and the applications to the client, consolidating all software updates to the servers rather than the clients.

- The idea is to reduce maintenance by having the clients be unchangeable with minimal moving parts, and the operating system and software upgrade and maintenance be completely centralized.
- LAWR is requesting a breakdown of costs for purchasing and maintenance of a fat client vs thin client solution for staff and possibly the labs. The areas most likely to be affected:

– Initial client buy-in
– Client maintenance
– Initial infrastructure buy-in
– Infrastructure maintenance

Initial Client Buy-in

***Fat Client*:** Currently on a 4 year rotation of staff computers, each station is estimated at $1000 (including new monitor). With approximately 20 staff users, that's an estimated $5000 per year, every year. Once staff all have decent monitors (17”-19” flat panel displays), the price will drop to $750 per

***Thin Client*:** Thin clients conceivably have a 6-10 year lifespan. Cost of thin clients are estimated at $700 with monitor, $500 without. The intial cost to replace staff computers with thin clients would likely be rolling, with replacements coming over a four year span. Each year for the first four years, the cost would be $3000. Once flat panels are in place, upgrades will come again in two years, switching or upgrading one quarter of the clients at a cost of $2000 per year for four years. In the initial 10 year period (assuming one half of staff currently have flat panels):

– Fat Client Cost: $40,000
– Thin Client Cost: $14,000

Cost savings of Thin Clients over 10 year period: $36,000 or 65% savings

Initial 20 year period:

– Fat Client Cost: $77500
– Thin Client Cost: $22000

Cost saving of Thin Clients over 20 year period: $55,300 or 71% savings

Client Maintenance

***Fat Client*:**

- Staff computers are fairly low maintenance as their software is standardized and users don't have administrative rights. With drive imaging, building new computers is much faster. Hardware maintenance is standard, meaning the computers rarely need parts and the warranties make replacements quick and cost-free.
- The two most time-consuming maintenance aspects are moving email on new workstations as that is the one thing stored locally, and rolling out software updates for non-Microsoft packages (Adobe, for example).

- Staff generally have standard software and much of it updates itself so time is minimal. Estimated time per new workstation: 4 hours. Estimated time per workstation per year: 4 hours.

***Thin Client*:** Once the entire Thin Client architecture is in place, setup of thin clients is minimal. A slight configuration change on the desktop, drop the client in place on the user's desktop, and they are ready to go. Email will be stored on a server, profile (desktop, my documents, etc) on a server, etc. Estimated time per workstation per year: 1 hour.

Initial 10 year period:

– Fat Client Maintenance Cost: $15,000

– Thin Client Maintenance Cost: $4000

Cost savings of Thin clients: $11,000 or 74%

Initial Infrastructure Buy-in

***Fat Client*:** In place, no cost.

Thin Client: Substantial. To support 20 simultaneous thin clients on a scalable, manageable terminal server setup, you need several items in place: Terminal Servers, Licensing Server, Load Balancing, E- Mail Server, SpamServer, SoftGrid Server, FileServer. The servers cannot be combined due to their nature making hardware costs alone substantial. The creation of this infrastructure would take months of planning, learning, and testing.

Infrastructure Maintenance

***Fat Client*:** The areas where fat clients depend upon infrastructure overlap with the thin client solution: Z drive (file server), networking, authentication servers.

Thin Client: Aside from the overlap with fat client infrastructure, the thin client solution relies completely on multiple servers to be fully operational. The failure of any one of the five types of servers results in either work stoppage for all, or no email for all. Upgrade of software applications and operating systems falls to the server side which centralizes general upkeep. In the short term, where hardware warranties prevail, server issues should be minimal...after a the 3-5 year mark, failures become costly and, as everything in the solution is mission-critical, costly in both IT time and normal staff time. Overall, maintenance shouldn't be much different than fat client, it just has more impact when an issue arises.

Tools: Java, Brew, Windows CE, WAP, Sybian, and EPOC.

Mobile Tools for Java

Introduction

Mobile Tools for Java (MTJ) is a proposed open source project under the Device Software Development Platform (DSDP).

Eclipse + Mobile= Mobile Tools For Java

Goal

- The goal of the Mobile Tools for Java™ (MTJ) project is to extend existing Eclipse frameworks to support mobile device Java application development.

- Mobile Tools for the Java Platform (MTJ) completed its first release (0.7) in November. MTJ's power lies in its extensibility in the highly customizable world of Java ME.
- MTJ is both an IDE for developers in the Java ME space, as well as an extensible architecture that allows for customization to aid development against any device, configuration, or profile. The MTJ architecture consists of several small plug-ins which can easily be extended or replaced to change the functionality of the IDE. In addition there is a compilation environment for Unified Emulator Interface (UEI) devices, and the ability to extend the architecture for other environments.
- The scope of the Mobile Tools for the Java Platform (MTJ) project is to extend the Eclipse platform to enable developers to develop, debug, and deploy mobile Java applications to emulators and real devices. The project will develop frameworks that can be extended by tool vendors and tools that can be used by third-party developers.

Background

- The Java programming language is becoming more and more popular in mobile devices. Also, the richness of the device Java environment is getting better all the time and more applications can be written using Java, instead of native languages. As Java continues to grow in popularity, along with the proliferation of higher functioning mobile devices, it is apparent that Java applications can be developed to run on multiple targets, with a common set of application code.
- Developing applications to the mobile Java environment presents unique challenges to developers. Specifically, unlike the straightforward J2SE and J2EE environments, there are a number of configurations and profiles (such as MIDP on top of the CLDC and Foundation and Personal Profiles on top of the CDC), along with a number of JSRs (and umbrella JSRs such as JTWI or JSR-248) that require development tools to assist in managing the runtimes/class libraries for development work and runtime binding. This ability to develop for multiple targets and use common source code with different build configurations is critical in mobile Java development projects.
- In addition to this management of runtimes and the challenges it presents, mobile Java applications have unique launching and debug requirements and unlike J2SE or J2EE, applications are not always just placed on a server for download as needed. Rather, developers require device emulators for on-development-system test and debug, the ability to launch, test, debug and analyze performance of the applications on the devices themselves, where varying classes of these devices have different methods and levels of connectivity.
- There is also the need for a robust deployment solution to deliver the code under development to the device, and also be able to map that to a production deployment solution. Of course, this is all in addition to the normal set of application development tools (such as code creation, UI design and so on) for a specific market segment.
- A common set of tooling and frameworks across these targets, and the mobile Java space makes the development effort and cost manageable for developers. This is why it is

important to have robust Eclipse frameworks in place to support mobile Java application development alongside the rich client and server counterpart

Versions

MTJ 1.1.2 [latest version]
MTJ 1.1.1
MTJ 1.1.0
MTJ 1.0.1
MTJ 1.0

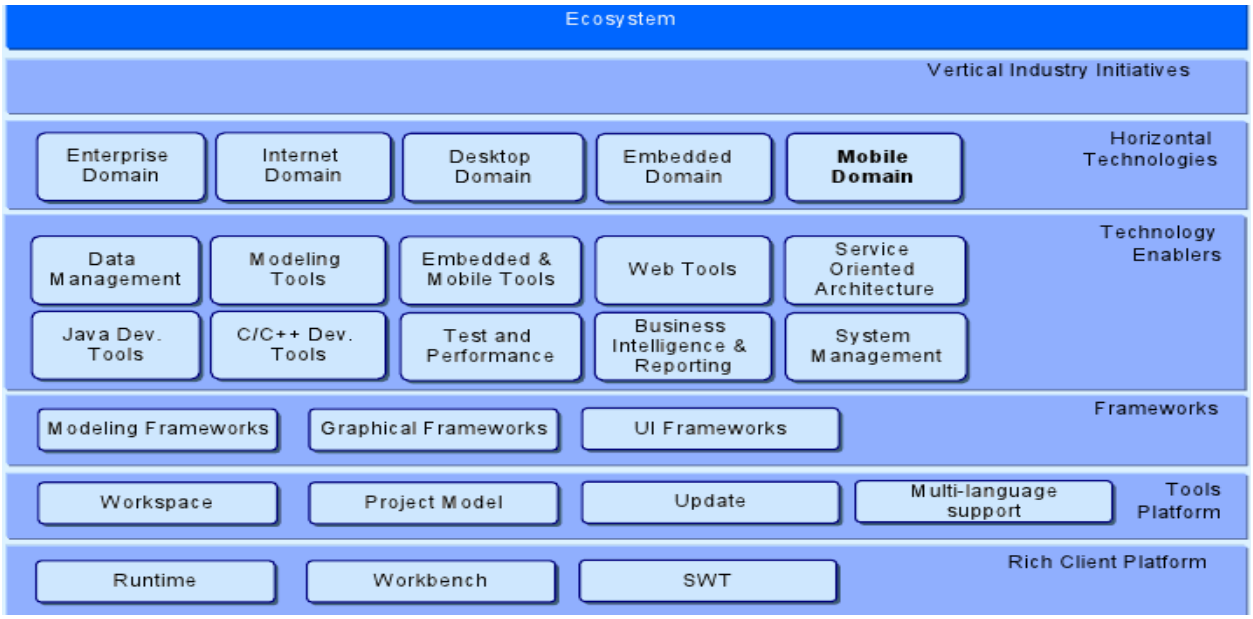
Description

The goal of the Mobile Tools for Java project is to extend existing Eclipse frameworks to support mobile device Java application development.MTJ will enable developers to develop, debug and deploy mobile Java applications to emulators and real devices.

Core features:

Device & Emulator framework

Normally runtimes are managed in Eclipse as a JRE. In the mobile segment, the runtime environment is a device emulator or real device. The MTJ project will provide features to



manage mobile runtimes and provide frameworks for device vendors to add those runtimes to the development environment.

Eclipse High Level Architecture

Deployment framework

One vital part in mobile development is testing on real devices. To make that as easy as possible the developer must have methods to transfer mobile applications to handheld devices

using local methods (e.g. Bluetooth, USB, IrDA). The idea of MTJ is to develop a framework that provides an API for vendor specific plug-ins, which then do the actual deployment.

Generic and customizable build process for mobile application development

The goal is to enhance normal build process with mobile extensions like JAR and JAD file generation. With the varying configurations and profiles of mobile Java, this is a critical feature to enable developers to manage code production. Another requirement is to provide ways to add additional tasks to the build process e.g. signing. This work should extend the builder frameworks of Eclipse.

Debugging

This is related to Device & Emulator framework. The goal of this item is to enhance the current Debugging environment so that it is possible to use mobile runtimes, either emulator or a real device. This task extends to launching the JVM and applications(s) on the local emulator or on the device itself, and allowing the developer to attach to that application under test. In the mobile space, this is tightly integrated with the Device and Emulator frameworks, and will need to provide a robust framework for device and platform makers to extend to their devices specific connectivity. This work should extend the debug frameworks of Eclipse.

Application creation wizard

Application wizards in general exist in the Eclipse platform. It is possible to generate Java projects with or without application skeleton code but the existing wizards are not usable for Mobile development. This task is tied to the differing configurations and profiles of mobile Java and relieves the developer of needing to worry about the boilerplate code for each of the application configuration/profile types. One of the goals of MTJ is to enhance the existing wizards by providing Mobile specific project wizard.

UI design tools

- The goal of these tools is to improve efficiency (easy drag and drop without coding) and also decrease the entry barrier for newcomers. There are already visual designers in the Eclipse platform but they dont contain support for mobile devices.
- The target is to bring mobile UI styles, components and layout rules to Eclipse. The idea is to create a framework that enables the use of different kinds of UI implementations e.g. different resolutions, different vendor specific look and feel.. Also in the scope of this project, in addition to this framework is at a minimum. a generic UI designer implementation.
- A Screen Flow designer tool would provide ways to develop application logic easily. It would provide easy drag and drop functionality to add different kind of screens and transitions between them. These transitions are caused by mobile specific ways like commands, list item selections.
- The idea is to utilize Eclipse Visual Editor Project and extend it so that device screen engines from different vendors can be plugged in.

Localization

The mobile application market is world wide so applications typically need to be localized. Therefore, an important requirement is for tooling that makes this task easier.

Obfuscation and Code Size/Performance Optimization

Mobile devices are restricted by memory so it is important that code is compresses as small as possible. There are a number of ways this can be accomplished. Obfuscation is one possibility, along with tooling and frameworks to enable performance and size analysis on the emulators or physical devices, which can be driven back into the build process or just for the developer to streamline their own code.

Signing tool

- Security becomes more important in mobile devices because they are open and accessible to 3rd party developers. One solution for this is to require that the applications be signed with an authorized signing certificate.
- The signing process is very similar to signing a normal JAR signing but there are some specific mobile needs. The goal of MTJ is to support tooling to sign mobile applications. This signing tool will provide extensibility to add solution specific signing tools. This is exemplary of other external tools for mobile Java that need to be able to be integrated to a customizable build process.
- The goal for the MTJ project is to have a first release simultaneous with the next main release of the Platform. Due to the aggressive schedule, the premise for this first release is to have the core set of functionality to provide a project that is usable by a mobile Java developer to generate and test code.
- It is a challenging task to define the content and future directions of the Mobile Tools for Java project at this point. The high level intent is to provide mobile tool frameworks for any interesting enhancements in the mobile world. There will be a placeholder for any JSR or de facto standard that gets industry acceptance. Here are a few examples: JSR-232 (Mobile Operational Management), JSR-249 (Mobile Service Architecture for CDC), JSR-271 (MIDP 3.0) and so on.

2.5 BREW

Qualcomm's BREW (Binary Run-time Environment for Wireless) gives application developers a new and different approach in producing mobile applications. BREW is built directly into the hardware. It is offered as an API to access the CDMA, GSM/GPRS, or UMTS chip sets that provide the support for it. But, it is primarily intended for the variations of CDMA, a technology owned and licensed by Qualcomm. BREW applications can be written on a PC using the BREW Software Development Kit (SDK). Once the application is developed, it must be tested, and then deployed. Deployment of BREW applications is a process done jointly by Qualcomm and telecommunications carriers and not just the developer.

Though the creators of BREW say that they first came up with the acronym BREW and then found the words to fit the acronym, the platform is somewhat biased toward wireless applications that run on phones. And this may be the only weakness of BREW as a mobile development platform. Although today developing mobile applications means targeting cell phones or PDAs, this is changing rapidly with new devices being introduced to the market.

BREW applications, also referred to as BREW applets, are written in C though some support for C++ is provided (although some fundamental things such as extending the base API through inheritance are not possible) and, using code generation or virtual machine technologies, other languages such as Java can be supported. One of the most impressive things about BREW is its near-full treatment of dimensions of mobility in its architecture, feature implementation, and SDK. Let us look at the various components that allow the developer to build a BREW application.

2.5.1 BREW SDK Overview

To get started programming in BREW, the first thing you need to do is to go to http://www.qualcomm.com/brew and register as a developer. This will allow you to download the BREW SDK. To date, the BREW SDK is offered mainly as an integrated set of components with Microsoft Visual C++ 6.0. Once you have downloaded the BREW SDK and installed it, you can begin developing. At the time of authoring this text, the BREW SDK is at its 2.0 version and its effective use requires installation of Microsoft Visual C++ 6.0. Once you have installed the BREW SDK,

you will have the following set of applications available for development:

1. *BREW MIF Editor*: Every BREW module, defined as the classes that make up one or more BREW applications, has an associated Module Information File (MIF). MIFs are *required*. Every BREW module must have a MIF. The MIF Editor provides a GUI tool for editing the MIF file associated with the classes that make up a module. The MIF Editor that comes with BREW SDK version 2.0 can be started as a wizard inside Visual C++ 6.0 or independently as a stand-alone application. We will look at the use of the MIF Editor and building a simple application.
2. *BREW Device Configurator*: This is a stand-alone application that allows developers to make up their own handset by configuring a vanilla mobile phone and specifying the behavior of the keys, the look and feel of the screen, and other specifics of the device. This development tool addresses the large variety of existing devices by allowing developers to create their own device emulator and testing the application. Remember, also, that because BREW is a platform for writing application for the handset, it is possible to use the application to build some adaptive behavior to adapt to each type of device. Still, the Device Configurator is invaluable in that it allows developers to test the application on their own emulated device environment.
3. *BREW Emulator*: For those who have designed and implemented any mobile application, it is obvious that one of the most difficult steps in the development process is the incremental unit testing. Although most platforms provide some sort of a generic emulator, most do not allow for custom configuration of a device (done by the Device Configurator) or using the custom configuration to simulate running an application. This is what the BREW Emulator does. But, the most impressive part is its treatment of location sensitivity, quality of service, and telephony functionality. Not only does the BREW Emulator allow the developer to load and run the application on a custom configuration, but also it allows for the adjustment of various components of the network's connectivity, such as traffic up-delay and down-delay, so that the application may be tested under various QOS conditions. The emulator also allows for emulating location-sensitive applications by configuring a GPS output file manually and using it to simulate the location input to the device. Finally, it allows the developer to simulate various telephony events such as an incoming call or sending a Short Messaging Service (SMS) message. The BREW Emulator, though primitive in its look and feel, is perhaps the most complete emulator for applications that run on mobile devices.
4. *BREW Image Authoring Tool*: There is an image authoring tool that allows creation of images for BREW. This tool can use PNG or BMP files.
5. *BREW ARM Compiler*: Many mobile devices are based on the ARM or StrongARM hardware platform (registered trademarks of ARM Corporation). The ARM Compiler enables the BREW developers to compile their code for the mobile devices that carry the ARM-based technologies (16/32 bit RISC-based microprocessors). The ARM Compiler has a licensing fee associated with it.

6. *Image Converter*: The tool set provides an image converter to convert 4-bit bitmaps to 2-bit bitmaps as BREW only supports 2-bit bitmaps because of the limited resources on mobile devices
7. *BREW Resource Editor*: If you have worked with Java or C++ to build GUI client-side applications, then you are familiar with the concept of a resource bundle. Resource files in BREW are a collection of images, strings, and dialog look-and-feel components that allow changing the look and feel of the application for internationalization and similar purposes without changing the code base. The BREW Resource Editor gives the developers a GUI interface to manage the resource files.
8. *BREW Pure Voice Converter*: This command line utility allows the developers to convert wave files (audio) to Pure Voice files or vice versa.
9. *BREW AppLoader*: This tool allows the developer to deploy an application on a handset through a PC connector. This is a testing and not a deployment tool.
10. *BREW Grinder*: The Grinder generates a variety of inputs and tests the application.
11. *BREW TestSig Generator and AppSigner*: The TestSig tool provides the developer a mechanism to generate a test Class ID. The AppSigner uses the Class 3 Certification from Verisign (see Section 2.5.2 for how this plays into the development process) to authenticate and sign an application.

Windows CE

- Microsoft Windows CE (now officially known as Windows Embedded Compact and previously also known as Windows Embedded CE, and sometimes abbreviated WinCE) is an operating system developed by Microsoft for embedded systems. Windows CE is a distinct operating system and kernel, rather than a trimmed-down version of desktop Windows. It is not to be confused with Windows Embedded Standard which is an NT-based componentized version of desktop Microsoft Windows.

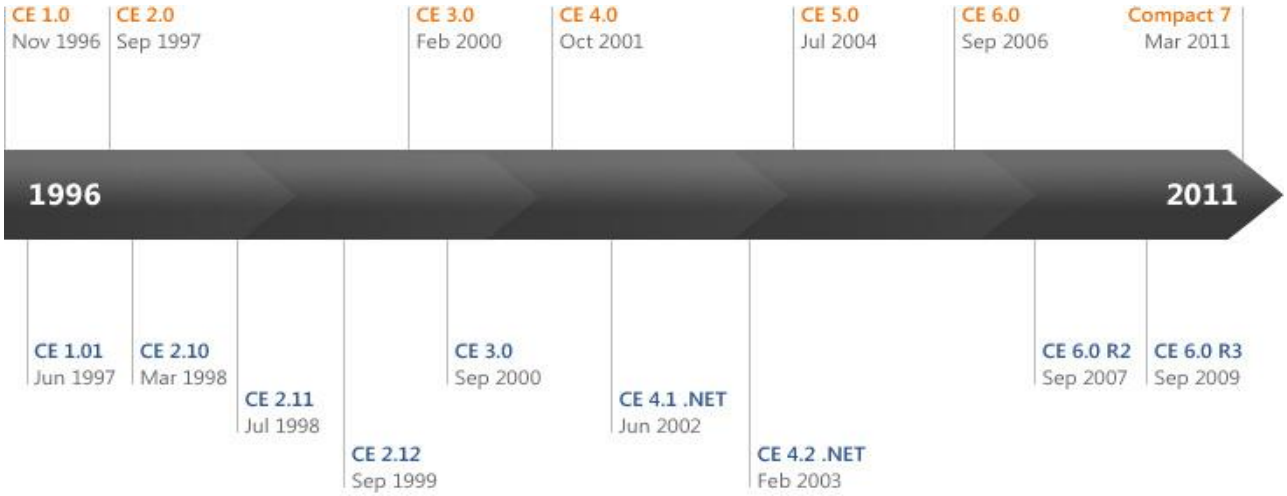
Features

- Windows CE is optimized for devices that have minimal storage; a Windows CE kernel may run in under a megabyte of memory. Devices are often configured without disk storage, and may be configured as a "closed" system that does not allow for end-user extension (for instance, it can be burned into ROM). Windows CE conforms to the definition of a real-time operating system, with a deterministic interrupt latency.
- Windows CE has evolved into a component-based, embedded, real-time operating system. It is no longer targeted solely at hand-held computers.
- Many platforms have been based on the core Windows CE operating system, including Microsoft's AutoPC, Pocket PC 2000, Pocket PC 2002, Windows Mobile 2003, Windows

Mobile 2003 SE, Windows Mobile 5.0, Windows Mobile 6, Smartphone 2002, Smartphone 2003, Portable Media Center, Zune, Windows Phone and many industrial devices and embedded systems. Windows CE even powered select games for the Dreamcast, was the operating system of the Gizmondo handheld, and can partially run on modified Xbox game consoles.

- A distinctive feature of Windows CE compared to other Microsoft operating systems is that large parts of it are offered in source code form. First, source code was offered to several vendors, so they could adjust it to their hardware. Then products like Platform Builder (an integrated environment for Windows CE OS image creation and integration, or customized operating system designs based on CE) offered several components in source code form to the general public. However, a number of core components that do not need adaptation to specific hardware environments (other than the CPU family) are still distributed in binary only form.

Versions

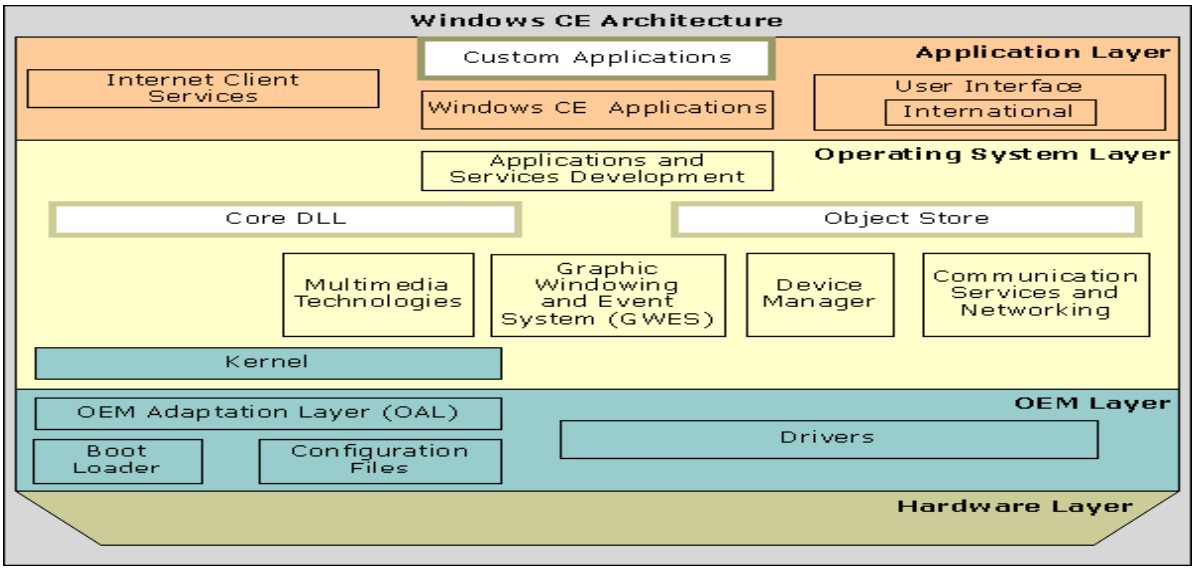


- Microsoft officially entered the embedded marketplace in November 1996 with the release of Windows Embedded CE 1.0.
- The first version—known during development under the code name "Pegasus"—featured a Windows-like GUI and a number of Microsoft's popular applications, all trimmed down for smaller storage, memory, and speed of the palmtops of the day.
- Windows Embedded CE was designed from the ground up to provide embedded developers with the ability to extend the sophisticated software environment of today's personal computer into the embedded world.
- Windows Embedded CE originally was developed for original equipment manufacturers (OEMs) building small, resource-constrained, handheld and Personal Information Manager (PIM) devices.
- In developing Windows Embedded CE, the embedded development team focused on four key areas: providing scalable wireless technologies to flexibly connect mobile devices;

providing reliable, core operating system services for demanding real-time designs; enabling rich personalized experiences that span devices, PCs, servers and Web services; and delivering a rich, easy-to-use, end-to-end tool set.

- Windows Embedded CE saw significant improvements with subsequent versions of the embedded operating system, including a simplified wizard-based operating system configuration, export software development kits (SDKs) to enable application development, multimedia support with version 2.12, and enhanced Internet capabilities and support for hard real time with Windows CE 3.0.
- Version 3 and onward, the system supports 256 priority levels and uses priority inheritance for dealing with priority inversion. The fundamental unit of execution is the thread. This helps to simplify the interface and improve execution time. Microsoft says the letters instead imply a number of Windows CE design precepts, including "Compact, Connectable, Compatible, Companion, and Efficient."
- The fourth generation of Windows Embedded CE added emulation technology to enable developers to perform their development and testing using a Windows 2000 or Windows XP Professional workstation without additional hardware investment.
- Windows CE 5.0, which was released in July 2004, included many key Shared Source components; a program designed to enable OEMs to build better devices faster through source level access, and was considered the most open Microsoft OS to date. CE 5.0 gave developers the freedom to modify down to the kernel level, without the need to share changes with Microsoft or its competitors.
- In its sixth generation, Windows Embedded CE 6.0 featured a completely redesigned kernel, which supported more than 32K processes.
- Each process received 2 GB of virtual address space, compared with 32 MB in previous. It also provided a new file system that supported larger storage media, larger file sizes, removable media encryption, and more. With CE 6.0, a device maker could provide devices for the home, work, and field that consume media, share presentations, and connect to cellular networks.
- Windows Embedded Compact 7 is the latest release of the componentized, hard real-time operating system for small footprint devices. Compact continues the history of embedded innovation with:

- Silverlight for Windows Embedded, a UI framework included with Compact, combines the flexibility of declarative UIs with the performance of native code. Silverlight for Windows Embedded is based on Silverlight v3.0 and allows developers and designers to create and update device UIs using Microsoft Expression Blend.
- Compact also includes an updated Internet Explorer, built on the same core as IE in Microsoft Windows Phone 7 and includes support for Flash 10.1, panning and zooming, multi-touch, and viewing bookmarks using thumbnails.



Windows CE Architecture

Windows CE feature list

- Windows CE provides a Power Manager to manage device power and improve overall OS power efficiency
- Windows CE can reduce the power consumption of a target device and to maintain and preserve the file system in RAM during the reset, on, idle, and suspend power states.
- Complete micro kernel
- Enhanced reliability
- Enhanced scalability

Wireless Application Protocol

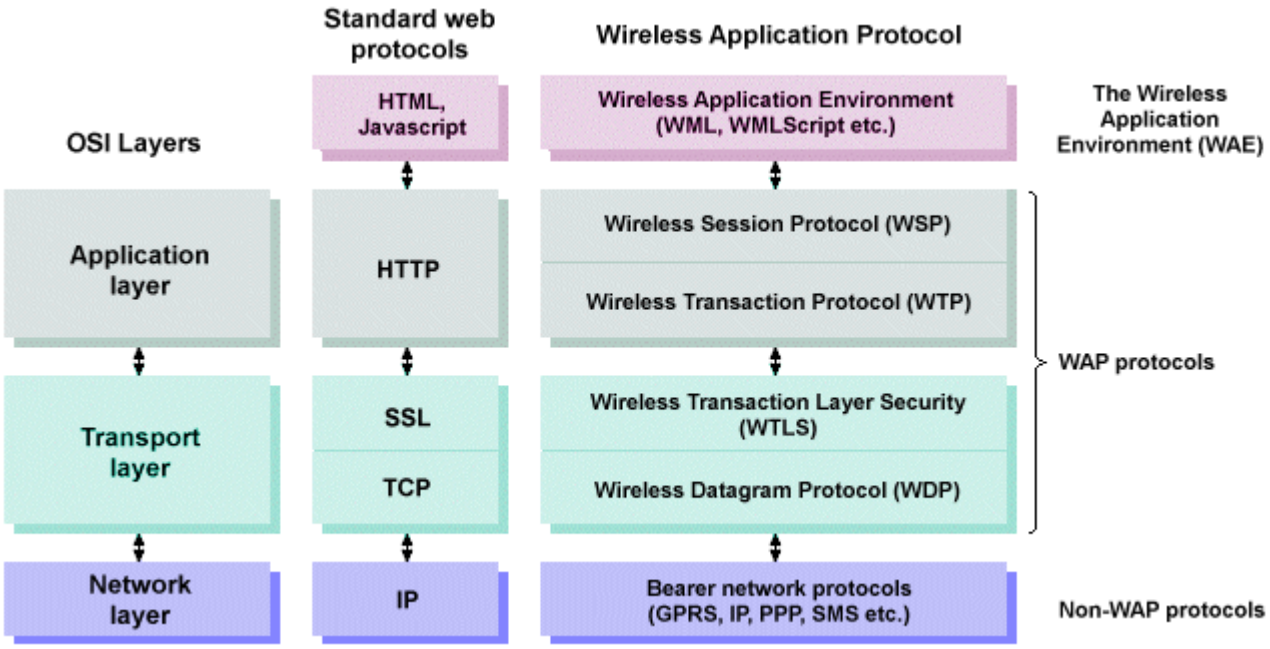
What is WAP?

WAP, Wireless Application Protocol, is a world wide standard which allows Internet based services to be accessed on mobile terminals like mobile WAP phones. Besides allowing the user to browse the Internet, the WAP terminal allows one to browse Intranet and Extranet services and to send/receive emails. Basically WAP offers all the standard Internet services but in a flexible and mobile wireless environment.

Wireless Application Protocol

- The *Wireless Application Protocol* (WAP) is an open standard first developed by the members of the *WAP Forum* in 1997 for mobile applications in a wireless communications environment.

- The goal was to develop a technology that would operate over any kind of mobile wireless network, including GSM, CDMA and TDMA-based networks, and 3G technologies such as UMTS. The WAP forum, whose original members included Nokia, Ericsson, Unwired Planet and Motorola, is now part of the *Open Mobile Alliance* (OMA). The forum's membership now includes players such as Microsoft, IBM, Oracle and Intel.
- WAP technology is primarily designed to enable mobile devices such as mobile phones and *personal data assistants* (PDAs) to access the World Wide Web and other Internet services such as email, information services and media content. It takes account of the need to optimize the use of the limited bandwidth available to such devices, minimize power requirements, and build in tolerance for the often unpredictable nature of wireless network availability.
- A *micro-browser* on the mobile device provides similar functionality to a web browser on a desktop computer, although its capabilities are limited due to the constraints imposed by the architecture of the mobile device (limited display space, processing power and memory).
- WAP content was originally created using the *Wireless Markup Language* (WML), an XML-based version of HTML optimized for wireless environment, and hosted on a standard web server. WAP also provided *WMLScript*, the wireless equivalent of JavaScript.
- The original WAP standard was version 1.0, released in 1998. It was closely followed by version 1.1 in 1999 and version1.2 in 2000. These early versions employed a protocol stack that was intended to completely replace the application and transport layer protocols used by the World Wide Web (primarily HTTP and TCP or UDP).
- The replacement protocols were similar in operation, and provided the same services as their standard counterparts, but were optimized for the constraints of a mobile wireless environment and designed to operate over any bearer network technology as opposed to only IP-based networks. The original WAP protocol stack is illustrated below.



The WAP protocol stack

- With version 1.x of the wireless application protocol, a request generated by a mobile device for a WML document will normally be routed via a *WAP gateway*.
- The gateway provides the interface between the web server, to which it is connected via the Internet, and the mobile client, to which it connects via the client's wireless bearer network. It performs the required translation between the WAP protocols used by the client device and the HTTP and TCP/IP protocols used by the web server. It also compresses downstream data (i.e. WML and WMLScript) for transmission across the wireless channel. The micro-browser on the client device decompresses and interprets the resulting byte-code and displays the results on the mobile device's display. The role of the WAP gateway is illustrated below.

- The WAP gateway carries out the necessary protocol translation and content encoding The WAP 1.x protocols enable communication to take place between the WAP gateway

and the micro-browser. They are described below (note that the network layer bearer protocols are not a part of the WAP specification).

WAP Protocols

WSP (Wireless Session Protocol)

Provides the application layer of WAP with a consistent interface for two session services. A connection-oriented service that operates above the transaction layer protocol WTP. A connectionless service that operates above a secure or non-secure datagram service (WDP).

WTP (Wireless Transaction Protocol)

Provide efficient request/reply based transport mechanism suitable for devices with limited resources over networks with low to medium bandwidth. WTP Push mode allows server to "push" data to a client without request (e.g. notification of stock hitting target price) WTP/WDP uses less than half the packets that TCP/IP uses to transfer the same amount of data.

WTLS (Wireless Transport Layer Security)

A security protocol based upon the industry-standard Transport Layer Security (TLS) protocol, formerly known as Secure Sockets Layer (SSL). WTLS is intended for use with the WAP transport protocols and has been optimized for use over narrow-band communication channels.

WDP (Wireless Datagram Protocol)

The Transport layer protocol in the WAP architecture Provides a common interface to the Security, Session, and Application layers Allows these upper layers to function independently of the underlying wireless network. This is the key to global interoperability.This first version of WAP was not an immediate success in either Europe or North America, although it fared rather better in Japan (possibly due to a more innovative approach to marketing and content provision). Authoring tools for WAP content were not widely available, and the level of support provided by service providers varied considerably. WAP version 2.0 went a long way towards remedying this situation, however, and WAP saw a revival of its fortunes between 2003 and 2004.

WAP version 2.0

- The radically re-engineered version of the Wireless Application Protocol was released in 2002 as version 2.0, and essentially replaces the original WAP protocol stack with standard Internet protocols. It does not therefore require a WAP gateway, but a proxy server is often used to optimize communication.
- A proxy can also improve access times and make more efficient use of network bandwidth by storing copies of frequently accessed resources. Current mobile devices have a much higher specification than the devices in use when WAP first appeared, enabling both HTTP and TCP to be used (although performance is often further enhanced through the use of the *wireless profile* version of both protocols).
- The Wireless Markup Language (WML) has been replaced by *XHTML Mobile Profile* (XHTML MP), a cut-down version of XHTML. This, together with improved availability of web authoring tools that support XHTML MP, means that it has become easier to create web content that is suitable for mobile devices. The use of HTTP at the application layer has the additional advantage that multimedia content can be sent to a WAP device

using the *Multimedia Message Service* (MMS), which has evolved from the *Simple Messaging Service* (SMS) used by millions of people every day to send text messages to each other. Another important concept introduced with WAP 2.0 is that of *push* – a mechanism that allows network servers to initiate the delivery of content to a WAP client device (as opposed to the normal *pull* model in which the server must wait for a request from the client device).

Current Constraints of Current Constraints of Wireless Interfaces:

Less Bandwidth
High Latency
Less Stable Connections
Less Predictable Availability
Diverse range of network standards

Current Constraints of Mobile Devices:

Less CPU Power
Less Memory and Storage
Restricted Power Consumption
Small / Variable Sized Displays Variable Input Types (Keypad, Pen, etc)

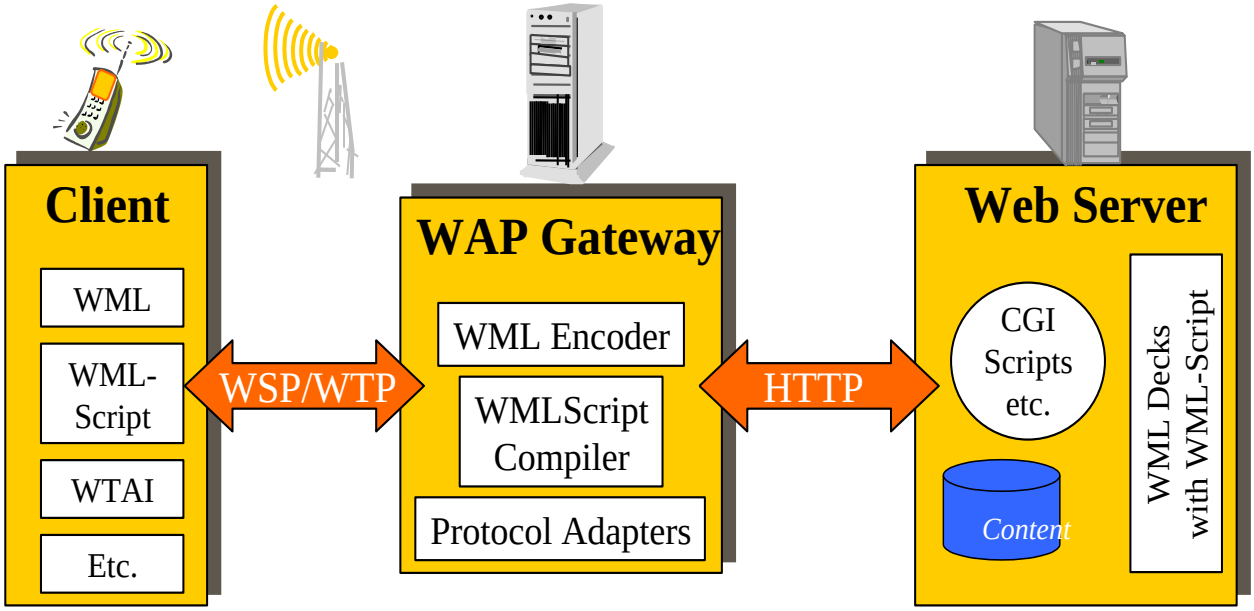
Use WAP

Wireless networks and phones have specific needs and requirements not addressed by existing Internet technologies.

WAP enables any data transport TCP/IP, UDP/IP, GUTS (IS-135/6), SMS, or USSD.

WAP Architecture

The WAP architecture has several modular entities which together form a fully compliant Internet entity all WML content is accessed via HTTP 1.1 requests. WAP utilizes standard Internet markup language technology (XML) Optimizing the content and air link protocols The WML UI components map well onto existing mobile phone user interfaces no re-education of the end-users leveraging market penetration of mobile devices WAP utilizes plain Web HTTP 1.1 servers leveraging existing development methodologies CGI, ASP, NSAPI, JAVA, Servlets, etc.



Symbian

- Symbian is a mobile operating system (OS) and computing platform designed for Smart phones and currently maintained by Accenture. Symbian was originally developed by Symbian Ltd.,as a descendant of Psion's EPOC and runs exclusively on ARM processors, although an unreleased x86 port existed.
- The current form of Symbian is an open-source platform developed by Symbian Foundation in 2009, as the successor of the original Symbian OS. Symbian was used by many major mobile phone brands, like Samsung, Motorola, Sony Ericsson, and above all by Nokia. It was the most popular Smartphone OS on a worldwide average until the end of 2010, when it was overtaken by Android, although in some developing nations, Symbian is still the biggest.
- Symbian rose to fame thanks to the S60 platform built by Nokia, first released in 2002 and powering pretty much every Nokia Smartphone. UIQ, another Symbian platform, ran in parallel, but these two platforms were not compatible with each other. Symbian^3, was officially released in Q4 2010 as the successor of S60 and UIQ, first used in the Nokia N8, to use a single platform for the OS. In May 2011 an update, Symbian Anna, was officially announced, followed by Nokia Belle (previously Symbian Belle) in August 2011. The latest (and last) phone with Symbian is the Nokia 808 PureView
- On 11 February 2011, Nokia announced that it would use Microsoft's Windows Phone OS as its primary Smartphone platform, and Symbian will be its franchise platform, dropping Symbian as its main Smartphone OS of choice. On 22 June 2011 Nokia made an agreement with Accenture for an outsourcing program.
- Accenture will provide Symbian-based software development and support services to Nokia through 2016; about 2,800 Nokia employees became Accenture employees as of

October 2011. The transfer was completed on 30 September 2011. The Nokia 808 Pure View was officially the last Symbian Smartphone.

Features

User interface

- Symbian has had a native graphics toolkit since its inception, known as AVKON (formerly known as Series 60). S60 was designed to be manipulated by a keyboard-like interface metaphor, such as the ~15-key augmented telephone keypad, or the mini-QWERTY keyboards. AVKON-based software is binary-compatible with Symbian versions up to and including Symbian^3.
- Symbian^3 includes the Qt framework, which is now the recommended user interface toolkit for new applications. Qt can also be installed on older Symbian devices.
- Symbian^4 was planned to introduce a new GUI library framework specifically designed for a touch-based interface, known as "UI Extensions for Mobile" or UIEMO (internal project name "Orbit"), which was built on top of Qt Widget; a preview was released in January 2010, however in October 2010 Nokia announced that Orbit/UIEMO has been cancelled.
- Nokia currently recommends that developers use Qt Quick with QML, the new high-level declarative UI and scripting framework for creating visually rich touch screen interfaces that allows development for both Symbian and MeeGo; it will be delivered to existing Symbian^3 devices as a Qt update. When more applications gradually feature a user interface reworked in Qt, the legacy S60 framework (AVKON) will be deprecated and no longer included with new devices at some point, thus breaking binary compatibility with older S60 applications.

Browser

- Symbian^3 and earlier have a built-in Web Kit based browser; indeed, Symbian was the first mobile platform to make use of Web Kit (in June 2005).[26] Some older Symbian models have Opera Mobile as their default browser.
- Nokia released a new browser with the release of Symbian Anna with improved speed and an improved user interface.
- Multiple language support Symbian has strong localization support enabling manufacturers and 3rd party application developers to localize their Symbian based products in order to support global distribution.

Application development

- QtAs of 2010, the SDK for Symbian is standard C++, using Qt. It can be used with either Qt Creator, or Carbide (the older IDE previously used for Symbian development).[28][32] A phone simulator allows testing of Qt apps. Apps compiled for the simulator are compiled to native code for the development platform, rather than having to be emulated.[33] Application development can either use C++ or QML.
- Symbian C++As Symbian OS is written in C++ using Symbian Software's coding standards, it is naturally possible to develop using Symbian C++, although it is not a

standard implementation. Before the release of the Qt SDK, this was the standard development environment. There were multiple platforms based on Symbian OS that provided software development kit (SDKs) for application developers wishing to target Symbian OS devices, the main ones being UIQ and S60. Individual phone products, or families, often had SDKs or SDK extensions downloadable from the maker's website too.

- The SDKs contain documentation, the header files and library files needed to build Symbian OS software, and a Windows-based emulator ("WINS"). Up until Symbian OS version 8, the SDKs also included a version of the GNU Compiler Collection (GCC) compiler (a cross-compiler) needed to build software to work on the device.
- Symbian OS 9 and the Symbian platform use a new application binary interface (ABI) and needed a different compiler. A choice of compilers is available including a newer version of GCC (see external links below).
- Unfortunately, Symbian C++ programming has a steep learning curve, as Symbian C++ requires the use of special techniques such as descriptors, active objects and the cleanup stack. This can make even relatively simple programs initially harder to implement than in other environments.
- It is possible that the techniques, developed for the much more restricted mobile hardware and compilers of the 1990s, caused extra complexity in source code because programmers are required to concentrate on low-level details instead of more application-specific features. As of 2010, these issues are no longer the case when using standard C++, with the Qt SDK.
- Symbian C++ programming is commonly done with an integrated development environment (IDE). For earlier versions of Symbian OS, the commercial IDE CodeWarrior for Symbian OS was favored. The CodeWarrior tools were replaced during 2006 by Carbide.c++, an Eclipse-based IDE developed by Nokia. Carbide.c++ is offered in four different versions: Express, Developer, Professional, and OEM, with increasing levels of capability.
- Fully featured software can be created and released with the Express edition, which is free. Features such as UI design, crash debugging etc. are available in the other, charged-for, editions. Microsoft Visual Studio 2003 and 2005 are also supported via the Carbide vs plug-in.

Other languages

- Symbian v9.1 with a S60v3 interface, on a Nokia E61Symbian devices can also be programmed using Python, Java ME, Flash Lite, Ruby, .NET, Web Runtime (WRT) Widgets and Standard C/C++.[34]
- Visual Basic programmers can use NS Basic to develop apps for S60 3rd Edition and UIQ 3 devices.
- In the past, Visual Basic, Visual Basic .NET, and C# development for Symbian were possible through AppForge Crossfire, a plug-in for Microsoft Visual Studio. On 13 March 2007 AppForge ceased operations; Oracle purchased the intellectual property, but announced that they did not plan to sell or provide support for former AppForge

products. Net60, a .NET compact framework for Symbian, which is developed by redFIVElabs, is sold as a commercial product. With Net60, VB.NET and C# (and other) source code is compiled into an intermediate language (IL) which is executed within the Symbian OS using a just-in-time compiler. (As of 18/1/10 RedFiveLabs has ceased development of Net60 with this announcement on their landing page: "At this stage we are pursuing some options to sell the IP so that Net60 may continue to have a future".)

- There is also a version of a Borland IDE for Symbian OS. Symbian OS development is also possible on Linux and Mac OS X using tools and methods developed by the community, partly enabled by Symbian releasing the source code for key tools. A plug-in that allows development of Symbian OS applications in Apple's Xcode IDE for Mac OS X was available.[35]
- Java ME applications for Symbian OS are developed using standard techniques and tools such as the Sun Java Wireless Toolkit (formerly the J2ME Wireless Toolkit). They are packaged as JAR (and possibly JAD) files. Both CLDC and CDC applications can be created with Net Beans. Other tools include SuperWaba, which can be used to build Symbian 7.0 and 7.0s programs using Java.
- Nokia S60 phones can also run Python scripts when the interpreter Python for S60 is installed, with a custom made API that allows for Bluetooth support and such. There is also an interactive console to allow the user to write Python scripts directly from the phone.

Deployment

- Symbian Belle, on a Nokia C7Once developed, Symbian applications need to find a route to customers' mobile phones. They are packaged in SIS files which may be installed over-the-air, via PC connect, Bluetooth or on a memory card.
- An alternative is to partner with a phone manufacturer and have the software included on the phone itself. Applications must be Symbian Signed for Symbian OS 9.x in order to make use of certain capabilities (system capabilities, restricted capabilities and device manufacturer capabilities).Applications can now be signed for free.
- Architecture Technology domains and packages Symbian's design is subdivided into technology domains, each of which comprises a number of software packages. Each technology domain has its own roadmap, and the Symbian Foundation has a team of technology managers who manage these technology domain roadmaps.
- Every package is allocated to exactly one technology domain, based on the general functional area to which the package contributes and by which it may be influenced. By grouping related packages by themes, the Symbian Foundation hopes to encourage a strong community to form around them and to generate discussion and review.
- The Symbian System Mode illustrates the scope of each of the technology domains across the platform packages.
- Packages are owned and maintained by a package owner, a named individual from an organization member of the Symbian Foundation, who accepts code contributions from the wider Symbian community and is responsible for package.

- Symbian kernel The Symbian kernel (EKA2) supports sufficiently fast real-time response to build a single-core phone around it—that is, a phone in which a single processor core executes both the user applications and the signaling stack.[41]
- The real-time kernel has a microkernel architecture containing only the minimum, most basic primitives and functionality, for maximum robustness, availability and responsiveness. It has been termed a nanokernel, because it needs an extended kernel to implement any other abstractions. It contains a scheduler, memory management and device drivers, with networking, telephony and file system support services in the OS Services Layer or the Base Services Layer. The inclusion of device drivers means the kernel is not a true microkernel.
- Design Symbian features pre-emptive multitasking and memory protection, like other operating systems (especially those created for use on desktop computers). EPOC's approach to multitasking was inspired by VMS and is based on asynchronous server-based events.

Symbian OS was created with three systems design principles in mind:

1. The integrity and security of user data is paramount
2. User time must not be wasted
3. All resources are scarce

- To best follow these principles, Symbian uses a microkernel, has a request-and-callback approach to services, and maintains separation between user interface and engine. The OS is optimized for low-power battery-based devices and for ROM-based systems (e.g. features like XIP and reentrancy in shared libraries). Applications, and the OS itself, follow an object-oriented design: Model-view-controller (MVC).
- Later OS iterations diluted this approach in response to market demands, notably with the introduction of a real-time kernel and a platform security model in versions 8 and 9.
- There is a strong emphasis on conserving resources which is exemplified by Symbian-specific programming idioms like descriptors and a cleanup stack. Similar methods exist to conserve disk space, though disks on Symbian devices are usually flash memory. Further, all Symbian programming is event-based, and the central processing unit (CPU) is switched into a low power mode when applications are not directly dealing with an event. This is done via a programming idiom called active objects. Similarly the Symbian approach to threads and processes is driven by reducing overheads.

Operating system The All over Model contains the following layers, from top to bottom:

- UI Framework Layer
- Application Services Layer

- Java ME
- OS Services Layer
- generic OS services
- communications services
- multimedia and graphics services
- connectivity services
- Base Services Layer
- Kernel Services & Hardware Interface Layer

Symbian includes a reference user-interface called "TechView." It provides a basis for starting customization and is the environment in which much Symbian test and example code runs. It is very similar to the user interface from the Psion Series 5 personal organizer and is not used for any production phone user interface.

EPOC Based- Handheld Computers

EPOC is a versatile operating system designed for usage in various mobile devices. Beside sub-notebooks, it is also applied for phones. EPOC has been developed by the Symbian consortium, founded by Ericsson, Motorola, Nokia, and Psion. EPOC based sub-notebooks are available from Ericsson, Oregon Scientific's OsariS, and Psion's 5mx (Figure 2.7), Series 7, and Revo. All of them are based on the current version of EPOC, EPOC Release 5, except for the OsariS, which still uses EPOC Release 4.

Remarkable is, that the EPOC operating system is very stable and users only experience crashes every few years. These devices contain a rich set of applications for mobile workers. Besides the PIM applications, they feature a word processor, a spreadsheet application, a web browser which supports Java applets, a communication suite with email, fax and SMS, as well as an application to backup and synchronize the device with a PC.

Data entered on an EPOC device can be exported to PC documents and imported in applications like Lotus SmartSuite, WordPerfect or Microsoft Word. There is plenty of additional third-party software available for EPOC devices. Maps, databases, dictionaries, tools for internet, email, fax and SMS, financial calculators, and accounting applications, just to name a few of them.

Table 2.1: Comparing Handheld Computers

Device Class	Palm	Pocket PC	EPOC
A sample Product	Palm i705	Casio E-200	Psion Series 5mx
Operating System	Palm OS 4.1	PocketPC 2002	EPOC 32 bit
Size	8 x 12 x 2 cm	13 x 7.8 x 1.6 cm	17 x 9 x 2 cm
Weight	140 – 160 g	255 g	350 g
Data Entry	Stylus	Stylus	Keyboard
Display Size	7.8 cm	21 – 25 cm	12 – 14 cm
Display Resolution	160 x 160 pixels Monochrome or 256 colors touch screen	320 x 240 pixels 65000 colors touch screen	640 x 240 pixels Monochrome touch screen
RAM	8 MB	64 MB	16 MB
Processor	Motorola Dragonball VZ33, 33 MHz	Intel Strong ARM 1110, 206 MHz	32 bit RISC ARM710T, 36 MHz
Device Class	**Palm**	**Pocket PC**	**EPOC**
Battery charge	> 10 hours	12 hours	> 10 hours
Peripherals	Serial, Infrared, Extension Card slot	Serial, USB, Infrared, Compact Flash., Voice recording, Audio speaker,	Serial Port, Infrared, Compact Flash Voice recording
Applications	PIM, Email HotSync, Graffiti,	PIM, ActiveSync, Handwriting recognition, Internet Explorer, Office Suite, PocketOutlook Media player	PIM, Email Internet Browser, Database, Spreadsheet, WAP (MC218 only)