International Journal of Advance Research in Science and Engineering
Volume No.06, Issue No. 12, December 2017
www.ijarse.com
IJARSE
ISSN: 2319-8354

A SURVEY ONMOBILE OPERATING SYSTEM AND MOBILE NETWORKS

Vignesh Kumar K[1], Nagarajan R[2]

(1Departmen of Computer Science, PhD Research Scholar, Sri Ramakrishna College of Arts And Science, India)

(2Department of Computer Science, Assistant Professor, Sri Ramakrishna College Of Arts And Science, India)

ABSTRACT

The use of smartphones is growing at an unprecedented rate and is projected to soon passlaptops as consumers' mobile platform of choice. The proliferation of these devices hascreated new opportunities for mobile researchers; however, when faced with hundreds ofdevices across nearly a dozen development platforms, selecting the ideal platform is often met with unanswered questions. This paper considers desirable characteristics of mobileplatforms necessary for mobile networks research.

Key words:smart phones,platforms, mobile networks,mobileplatforms.

I.INTRODUCTION

In a mobile network, position of MNs has been changing due todynamic nature. The dynamic movements of MNs are tracked regularlyby MM. To meet the QoS in mobile networks, the various issuesconsidered such as MM, handoff methods, call dropping, call blockingmethods, network throughput, routing overhead and PDR are discussed. In this paper I analyse the five most popular smartphone platforms: Android (Linux), BlackBerry, IPhone, Symbian, and Windows Mobile. Each has its own set of strengths and weaknesses; some platforms trade off security for openness, code portability for stability, and limit APIs for robustness. This analysis focuses on the APIs that platforms expose to applications; however in practice, smartphones are manufactured with different physical functionality. Therefore certain platform APIs may not be available on all smartphones.

II.MOBILITY MANAGEMENT

IP mobility management protocols proposed by Alnasouri et al (2007), Dell'Uomo and Scarrone (2002) and He and Cheng (2011) are compared in terms of handoff latency and packet loss during HM. Mobile IP is the current standard for supporting mobility in IP networks. But it lacks support for fast handoff control, real-time location tracking as per the investigation by Sami (1995) and Madhow (1994), and QoS which are critical to real-time services in cellular networks. Stevens- Navarro (2010) demonstrate that the Mobile IP also has high control overhead due to frequent notification to the Home Agent and lacks support for soft handoff that requires the ability to multicast traffic tomultiple base stations during handoff(in current and 3G cellular networks, soft handoff is done on link-layer.

III.REQUIREMENTS

Network scanning

To access a wireless network, each mobile device is required to scan for APs or cell towers. The data returned from a scan contain information needed to associate with the network. The results of a scan can also be used to approximate the location or assess the mobility of a user. Cell tower scans are typically automated, whereas Wi-Fi and Bluetooth scans are almost always user initiated. However, scanning can rarely be initiated by a third party application. The results of the scan are also typically inaccessible. The ability to programmatically scan and connect to a wireless network can significantly impact the ability of a mobile device to serve as research platforms.

The presence of multiple network interfaces is common on many current smartphones. The most common interfaces are cellular, Wi-Fi, and Bluetooth. Each wireless technology has a unique set of benefits and costs. For example, Wi-Fi has lowmonetary cost, but consumes more power than a cellular data service such as EDGE. When multiple interfaces are within coverage, most mobile platforms naively direct all data traffic over theWi-Fi interface to minimize monetary cost. Immediately routing data over the least expensive interface is not necessarily optimal [29]. Moreover, this strategy restricts experimentation and measurement across multiple interfaces. Researchers require the means to route data over a specific network interface

Bluetooth I/O

Bluetooth first appeared as a consumer technology in 2000 and it is still going strong. It's a wireless communication protocol for connecting devices through the air - it's slower than Wi-Fi but is often simpler to set up, and is usually preferred for device-to-device transfers.With just about every smartphone out there supporting Bluetooth, it's become the default way for connecting up speakers, headphones and other devices to phones without wires. It works across plenty of other devices too, including printers, remotes and computers.Bluetooth uses a closed-off network of 79 bands of radio waves. Bluetooth devices have the capability to automatically detect each other and you can connect up to 8 different devices at once, all without using up too much valuable battery power along the way. There are two flavours of Bluetooth, Bluetooth Basic Rate/Enhanced Data Rate (BR/EDR) and Bluetooth Low Energy (LE). Bluetooth BR/EDR is more limited in terms of range but is more suitable for keeping up a continuous connection (for streaming audio, for example).Bluetooth LE has more potential for Internet of Things (IoT) use, where briefer bursts of data are required, and where power conservation is more important. It's also cheap to implement, which is why it's managed to find its way into so many consumer gadgets.

Interface control:

Interface to the Cellular Network

Wi-Fi and Ethernet technologies are similar, and apps can run on either interchangeably. This is not true for cellular networks, where data transfer has different requirements. Whereas Wi-Fi connectivity is relatively inexpensive, the cost per byte of the same data transmitted over a cellular connection can be quite costly—for the carrier as well as the user. Cellular networks are quite complex in how they acquire and use radio (airwave/spectrum) resources. That complexity is expensive for the carrier and they pass those expenses on to the user.

Cellular Data Is Expensive

On most mobile devices, apps can use multiple forms of connectivity, each carrying different expenses. Yet the transition from one to the other is often outside of the users' awareness. In addition, to manage cellular connections, the device itself usually has multiple cellular data modes with different data rates and different power usage, also invisible to the user.

When a mobile device user out of Wi-Fi range opens an Internet app, it typically takes the exchange of eight to ten cellular network messages (depending on the protocol) before a single byte of data is transmitted. This is far more overhead than in a typical Ethernet/Wi-Fi LAN exchange and it is the principal reason that apps designed for Ethernet and Wi-Fi are often not as efficient when used over a cellular interface.

Background processing:

Experimentation often requires monitoring and recording information in parallel with user activities and mobility. Therefore the ability to run background applications is essential for researchers.

Energy monitoring:

For the foreseeable future, maximizing battery life will continue to be a key concern for mobile networks research. Scanning for APs, transmitting data, and location sensing all have an energycost that must be measured. As researchers, we require a means to programmatically measure, (or at least approximate), the energy consumed by a mobile device.

Power saving control:

Testing of Embedded system has always faced numerous problems with complex architecture of software and hardware combinations. Rapidly growing energy and battery consumption demands of mobile phones are not accomplished with advances in battery technology. Energy consumption is an important issue in android devices for mobile devices having advanced operating system. The lifetime of the battery is more and more important issue for users, for example if Wi-Fi or other applications are used continuously in the mobile, the battery of mobile discharges before its specified time. This paper presents a battery monitoring system which is capable of sensing and monitoring capability of battery of mobile phone, which is used to indicate the battery conditions in any numbers of standby powers. The battery capability is to monitor any number of cells utilizing a frequency-shift-keyed signal to sequentially interrogate each individual cell. This modulated tone responses are used to transmit the data back to the monitoring module. This paper introduces the background service name as Main activity. This service provides Power saver for automated power model construction method that uses built-in battery voltage sensors and knowledge of battery discharge behaviour to monitor power consumption. To save android applications energy, it is critical to monitor the energy consumption of each application. As a solution to this, process acts as a background process and used to monitor battery power for each application and overall system. It also helps to inform users about the battery usage information and restricting the application which uses excess power.

Location sensing

Low-power GPS receivers and cell tower/AP triangulation techniques [12] are now found in many mobile devices. Although location sensing comes with an energy cost, (and occasionally a monetary cost), location information is commonly used to provide context to a number of applications including: navigation, search, and social interaction. As these technologies continue to improve and scale to large numbers of devices, location sensing in mobile devices will undoubtedly become ubiquitous. As researchers, we require the means to enable and disable location sensing sub-systems and query the current location from within applications.

II.A. Other Platform Requirements

While this list covers most research application requirements, it clearly does not cover all of them. Forexample, in [20] the authors exploit audio APIs in Windows Mobile for proximity detection. In [13],combinations of vibration and audio are used to signal the presence of buddies. The accelerometer, afeature now found in every mobile platform, is used in [8] to detect potholes while driving. Camera APIsare used [16] to provide local service discovery. Mobilesocial networking applications may also use theaddress book and other personal information APIs toinfer trust and aid in content dissemination [1].

III. Mobile Platforms

In this section I analyse the five most popular mobile platforms and their success in meeting our research needs. The results of this analysis are summarized in Table 1. This platform analysis focuses purely on the functionality provided by the OS. With the exception of the iPhone, each platform depends on the features of the physical device to implement its APIs. For example, devices without Wi-Fi antenna cannot scan for Wi-Fi APs, even though the platform may support it.

	Android (Linux)	BlackBerry	iPhone	Symbian	Windows Mobile
Bluetooth I/O	Not satisfied	Partially satisfied	Not satisfied	Partially satisfied:	Partially satisfied:
Interface control	satisfied	satisfied	Not satisfied	Not satisfied	satisfied
Background processing	satisfied	satisfied	satisfied	satisfied	satisfied
Energy	satisfied	satisfied	satisfied	satisfied	satisfied

monitoring					
Power saving control	satisfied	satisfied	Partially satisfied	satisfied	satisfied
Low-memory management	satisfied	satisfied	satisfied	satisfied	satisfied
Location sensing	satisfied	Partially satisfied	satisfied	satisfied	satisfied

TABLE : 1 Summary of mobile platforms requirements.

III.A. Android (Linux)

There's really only one argument in favour of classifying Android as Linux, but it's a strong one: every Android smartphone or tablet contains a Linux kernel. You can even see what version of the kernel is installed on your device, by opening its 'Settings' app and navigating to 'About Device > Software info.'

In order to create an operating system that meets the unique needs of mobile devices, the Android team made a number of changes to the Linux kernel, including the addition of specialized libraries, APIs and tools that are mostly BSD-derived or written from scratch, specifically for Android.

Since the entire argument in favour of classifying Android as a Linux distort hinges on the fact that Android uses the Linux kernel, this point might seem like it's the end of the argument, but it's not *that* unusual for a Linux distro to make changes to the kernel.

The Linux kernel is released under the GNU General Public License, so anyone is free to modify its source code, which many Linux distorts have done. When it comes to the question of just how drastically the Android team modified the Linux kernel, the Embedded Linux wiki concludes that the amount of changes implemented by the Android team "is not extremely large, and is on the order of changes that are customarily made to the Linux kernel by embedded developers."

While it's typically pretty easy to modify a Linux distribution at the operating system level, by default Android owners cannot access the underlying operating system on their smartphone or tablet, and sensitive partitions are locked down tight. However, the *by default* is important, as you can gain access to areas of your device that are normally closed off, by exploiting security flaws in the Android system, in a process known as *rooting*.

So, while Android is considerably less customizable than your typical Linux distribution, there are ways to gain access to the underlying operating system.

III.B. BlackBerry

BlackBerry is a mobile operating system originally created by Research In Motion (RIM). RIM now goes by the name BlackBerry. The BlackBerry name refers to the unique physical keyboard design on early devices that resembledthe fruit of the same name. Widely popular in the business world, BlackBerry is best known for its email features and BlackBerry Messenger (BBM™).

BlackBerry OS is a proprietary mobile operating system designed specifically for Research in Motion's (RIM) BlackBerry devices. The BlackBerry OS runs on Blackberry variant phones like the BlackBerry Bold, Curve, Pearl and Storm series.

The BlackBerry OS is designed for smartphone environments and is best known for its robust support for push Internet email. This push email functionality is carried out through the dedicated BlackBerry Enterprise Server (BES), which has versions for Microsoft Exchange. Other mobile operating systems like Android, Microsoft Windows Mobile and Symbian can run on different brands of mobile phones; the BlackBerry OS can run only on BlackBerry phones. BlackBerry OS is similar to Apple's iOS in this regard.

Traditionally, BlackBerry applications are written using Java, particularly the Java Micro Edition (Java ME) platform. However, RIM introduced the BlackBerry Web development platform in 2010, which makes use of the widget software development kit (SDK) to create small standalone Web apps made up of HTML, CSS andJavaScriptcode.

Those who opt to develop through Java can use the BlackBerry Java Development Environment (JDE), an integrated development environment (IDE) that comes with an editor, debugger, device simulator and memory viewer. The JDE can be downloaded from the BlackBerry website. It can be used either as a standalone or as a plug-in for Eclipse, a graphical IDE. Other tools that are used in conjunction with the JDE are the RAPC compiler and the Sun Java compiler.

There are three ways of installing apps on a BlackBerry OS: downloading an app from BlackBerry App World, over-the-air through the device's built-in browser and via the BlackBerry Desktop Manager.

III.C. iPhone (Mac OS X)

Like the BlackBerry, Apple's Mac OS X is tightly coupled with its device: the iPhone. The iPhone is an exceptional case in our survey. Out of the box, the iPhone and SDK provided by Apple are severely limited. The current SDK does not allow applications to initiate a Wi-Fi network scan or retrieve information about neighbouring cell towers. The SDK allows applications to detect if the iPhone has Wi-Ficonnectivity; however, applications cannot transmit over aspecific network. Moreover, Apple prohibits bandwidthintensive applications to be installed, nor canthird party applications run in the background. TheSDK does not allow applications to query the state ofthe battery or the level of available RAM. On the plus side, the iPhone SDK does provide persistent storagethrough conventional files and an integrated SQLitedatabase. The platform also provides location sensingthrough the 'Core Location Framework', whichdetermines its position using an integral GPS or cell tower/Wi-Fitriangulation. As a research platform, the functionality provided by the Apple SDK is insufficient. Fortunately, the iPhone has been liberated by members of the black hatcommunity. Unlocking, or 'Jail Breaking', the iPhone takes a matter of minutes using tools such as 'iPlus'. After installing the GNU or BSD sub-systems [7], the iPhone has all the capabilities of a standard Unix system. The iPhone OS provides APIs for both cell tower and Wi-Fi scanning, and network connections are made using conventional BSD sockets. However, I found no evidence in Apple documentation or in a survey of third party application that indicates that applicationsmay enable and disable network interfaces. The iPhone also provides no support for

Bluetooth I/O. As in previous systems, the iPhone OS provides APIs to query the battery capacity, voltage, and charging status.

Unlike previous systems, the iPhone has a virtual memory system with paging. Low-memory conditions are therefore less critical to the iPhone; however, one should be aware that frequent paging in conjunction with persistent flash memory I/O will have an adverse effect on battery life. IPhone is also far more aggressive in power management than other devices. By default, when a user is not interacting with the device, the screen is disabled and background processes are suspended. To run applications in the background of an unlocked iPhone, it is necessary to override the power saving setting and force the device to stay on. Developing on an unlocked iPhone has initial challenges;documentation is generally poor and it will probably take several attempts to successfully unlockthe device and install the desired sub-systems. The iPhone has a final caveat, future versions of the iPhone may restrict this openness, but given Apple's recent endorsement of iPhone unlockers [11], it is unlikely.

III.D. Symbian (S60)

All modern computing devices, big or small, require an Operating System, also known as an OS. The OS contains all the basic instructions that the computer needs in order to do things.

The most widely-used OS on home computers is Microsoft Windows. The most widely-used OS on smartphones is Symbian.

Symbian is the operating system used on the Nokia, on all of Nokia's smartphones, and on smartphones from many other manufacturers too such as Samsung, Sony Ericsson and LG.

S60, also known as Series 60, is a software platform and interface that sits on top of Symbian. In the past there were other flavours of Symbian such as Symbian UIQ, Symbian Series 80 and Symbian Series 90. However, nowadays all Symbian phone manufacturers use Symbian S60. Because all Symbian phones now use S60, the two terms are often used interchangeably.

Nokia's efforts in Power Management are notable. Of the five platforms surveyed, Nokia is only oneWith a developer tool, the 'Nokia Energy Profiler', that actively monitors the power consumption on the device. The tool is free to download, easy to use, but requires a Nokia device running the latest version of S60. On the device side, Symbian contains a framework for power management that allows applications to lock the power state, receive notifications of changes to power modes, adjust power requirements, or wakeup/shutdown the Symbian kernel.

Symbian also offers a Java runtime environment; however, unlike Android and BlackBerry, Symbian's Java environment is constrained to the Java 'Mobile Information Device Profile' (MIDP). Constraining Symbian Java applications to MIDP makes them highly portable, but the set of available APIs are very limited and do not satisfy the needs of mobile researchers.

III.E. Windows Mobile

To my great surprise, Microsoft's Windows Mobile satisfies every one of our research requirements. The OS provides facilities to scan for cellular, Bluetooth, and Wi-Fi networks, establish connections on a specific network interface, enable and disable interfaces, determine the current location using GPS, and to run applications in the background. Like the iPhone, Windows employs a virtual memory system and memory is therefore not a concern for applications. However, since Windows Mobile is commonly deployed on devices with less than 64 MB of total Flash memory, applications should be careful not to over indulge in memory. Bluetooth on Windows Mobile has distinct advantagesover BlackBerry and Symbian. The Windows Bluetooth API does not require devices to bepaired! Although these 'feature' exposesWindows Mobile users to a range of interesting attacks, it has huge advantages when prototyping mobile systems: devices can connect to each other without user intervention.

Windows provides applications with standard file I/O facilities and an integral database engine. Applications may also maintain state in a persistent global system registry consisting of key pairs (a mechanism similar to the Windows registry). As a researcher writing experimental code, I have found that the registry is an invaluable tool for storing state across countless crashes.

IV. SUMMARY AND CONCLUSION

To the average consumer, each platform is functionally equivalent. They all support making phone calls, saving files, taking pictures, and other common tasks; however, there is more to a mobile platform than meets the eye. Each platform has its own development environment that supports different sets of programming languages and APIs. Android has a rich set of APIs, but without Bluetooth it currently lacks the necessary requirements for multi-NIC related research. BlackBerry is the most robust mobile platform in our survey. It lacks someof the features of other platforms, but makes up for it with ease of development and stability. In five years developing applications for BlackBerry, I have never experienced a crash. Out of the box, iPhone is a substandard research platform; however, unlocking it exposes a rich set of APIs from its Mac OS X foundation.

The unlocked iPhone currently lacks the developer tools found on other platforms, which makesit more difficult to develop and debug applications. Conversely, Symbian includes an excellent set of developer tools and the platform supports many of our requirements. Unfortunately, Symbian is widely regarded as the most difficult platform to develop on. This fact alone negates many of its qualities and hurts its viability as a research platform. Finally, Windows Mobile, running on a wide range of devices and hardware configurations, has a rich set of APIs; unfortunately, support for those APIs is OEM dependent. Moreover, there is little incentive for OEMs to support niche APIs such as programmatic power management and network scanning.

REFERENCES

[1] G. Ananthanarayanan, R. Venkatesan, P. Naldurg,S. Blagsvedt, and A. Hemakumar. SPACE:Secure Protocol for Address-Book based Connection Establishment. 2006.

[2] Ganesh Ananthanarayanan, Venkata N. Padmanabhan, Lenin Ravindranath, and ChandramohanA. Thekkath. Combine: leveraging the power of wireless peers through collaborativedownloading. In *MobiSys '07: Proceedings of the 5th international conference on Mobile systems,applications and services*, pages 286–298, New York, NY, USA, 2007. ACM.

[3] Android.*http://code.google.com/android/*.

[4] Trevor Armstrong, Olivier Trescases, Cristiana Amza, and Eyal de Lara. Efficient and transparent dynamic content updates for mobile clients. In *MobiSys '06: Proceedings of the 4th internationalConference on Mobile systems, applications and services*, pages 56–68, New York, NY, USA, 2006.

[5] N. Banerjee, A. Rahmati, M.D. Corner, S. Rollins, and L. Zhong. Users and Batteries: Interactions and Adaptive Energy Management in Mobile Systems.

[6] BlackBerry. *http://blackberry.com/developers/*.

[7] iPhone. *http://www.saurik.com/id/1*.

[8] Jakob Eriksson, Lewis Girod, Bret Hull, Ryan Newton, Samuel Madden, and HariBalakrishnan.The pothole patrol: using a mobile sensor network for road surface monitoring. In*MobiSys '08: proceeding of the 6th international conference on Mobile systems, applications, and services*,
pages 29–39, New York, NY, USA, 2008.

[9] Ken Hinckley, Jeff Pierce, Mike Sinclair, and Eric Horvitz.Sensing techniques for mobile interaction. In *UIST '00: Proceedings of the 13th annual ACM symposium on User interface softwareand technology*, pages 91–100, New York, NY, USA, 2000. ACM.

[10] iPhone. *http://developer.apple.com/iphone/*.

[11] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard,J. Hughes, F. Potter, et al. Place Lab: Device Positioning Using Radio Beacons in the Wild. *Proceedings of Pervasive*, 3468:116–133, 2005.

[12] Kevin A. Li, Timothy Y. Sohn, Steven Huang, and William G. Griswold. Peopletones: a systemfor the detection and notification of buddy proximity on mobile phones. In *MobiSys '08:Proceeding of the 6th international conference on Mobile systems, applications, and services*,pages 160–173, New York, NY, USA, 2008.

[13] *http://limofoundation.org/*.

[14] *http://maemo.org*.

[15] MSDN Blog. *http://blogs.msdn.com support-boundaries-for-windows-mobileprogramming- developing-drivers-for-exampleor- even-wifi-programming.aspx*.

[16] *http://openmoko.org*.

[17] Anna-KaisaPietil¨ainen, Earl Oliver, Jason Le- Brun, George Varghese, Jon Crowcroft, andChristophe Diot. Experiments in mobile social networking. Technical Report CR-PRL-2008- 02-0003, Thomson, February 2008.

[18] QTopia. Last visited 12/11/2008.*http://qtopia.net*.

[19] Ahmad Rahmati and Lin Zhong.Contextfor- wireless: context-sensitive energy-efficientwireless data transfer. In *MobiSys '07: Proceedings of the 5th international conference on Mobilesystems, applications and services*, pages 165–178, New York, NY, USA, 2007. ACM.

[20] J. Scott, P. Hui, J. Crowcroft, and C. Diot. Haggle: A networking architecture designed aroundMobile users. *Proceedings of the Third Annual Conference on Wireless On demand Net work Systems and Services (WONS 2006)*, 2006.