ENSEMBLE COMPUTING: OPPORTUNITIES AND CHALLENGES

Contributors

Roy Want
Intel Corporation

Eve Schooler
Intel Corporation

Lenka Jelinek
Intel Corporation

Jaeyeon Jung
Intel Corporation

Dan Dahle
Intel Corporation

Uttam Sengupta
Intel Corporation

Index Words

Ensembles
Mobile Computing
Distributed Systems
Context-aware
Wireless Networking

Abstract

Since the advent of the first practical mobile computers more than 20 years ago, advances in the underlying technology have moved out of research and development and into both the enterprise and the area of consumer electronics. Today, mobile networked computers are ubiquitous. Trends that have enabled this revolution include improved processor performance, exponentially increasing storage density, and high-bandwidth standardized radios: all of these computers consume less power and are packaged in smaller form-factors than their predecessors. However, the formation of mobile collections of devices, or ensembles, can further build on the network effect to amplify ensembles' usefulness, a property first observed with telephony, and later the Internet. Ensembles can be created by using now commonly available wireless and wired standards, providing an aggregate value to a user that is greater than the sum of the component parts. In this article, we examine the opportunities for ensemble computing; classify the main usages and application domains; review enabling component technologies, and point out the research challenges that need to be addressed to realize the full potential of the ensemble concept.

Introduction

The history of modern computing can be divided into three distinct eras, each characterized by the number of computers available to an average user. During the first era, dominated by the mainframe (1960-1980), there were typically many people using a single computer. In the Personal Computer (PC) era (1980-2000), most people had their own PC. Today, we are in the third era (2000 onwards) in which it is now common for people to own and use many computers, sometimes referred to as the era of *ubiquitous computing* [1].

In addition to the trend of increasing numbers of computing devices per person per year, we have seen improved processor performance (driven by Moore's Law) and greater network connectivity for mobile computers that have been made possible through innovations in standardized wireless technologies. These standards provide high-bandwidth data-transmission, allowing seamless coordination between applications, and they enable fully distributed and replicated programming models. By a measure called the network effect, or Metcalfe's Law [2], N devices connected in this way create a total network value proportional to the number of connections they can make multiplied by the number of devices participating; this results in the network value growing at $O(N^2)$. Orchestrated collections of computers can thus provide greater aggregate value to users than would be expected from a simple sum of the component parts.

Although distributed programming has been the subject of research for some time with its foundation in networks of traditional servers or desktop computers, mobile computing brings dynamically changing topologies, context-awareness, and more diverse collections of computing resources, adding a new dimension to this field. We call dynamically coordinated collections of computers, which include both mobile and infrastructure components, *Device Ensembles* [3], and the techniques for programming and orchestrating their applications, *Ensemble Computing*.

"We call dynamically coordinated collections of computers, which include both mobile and infrastructure components, Device Ensembles.*"*

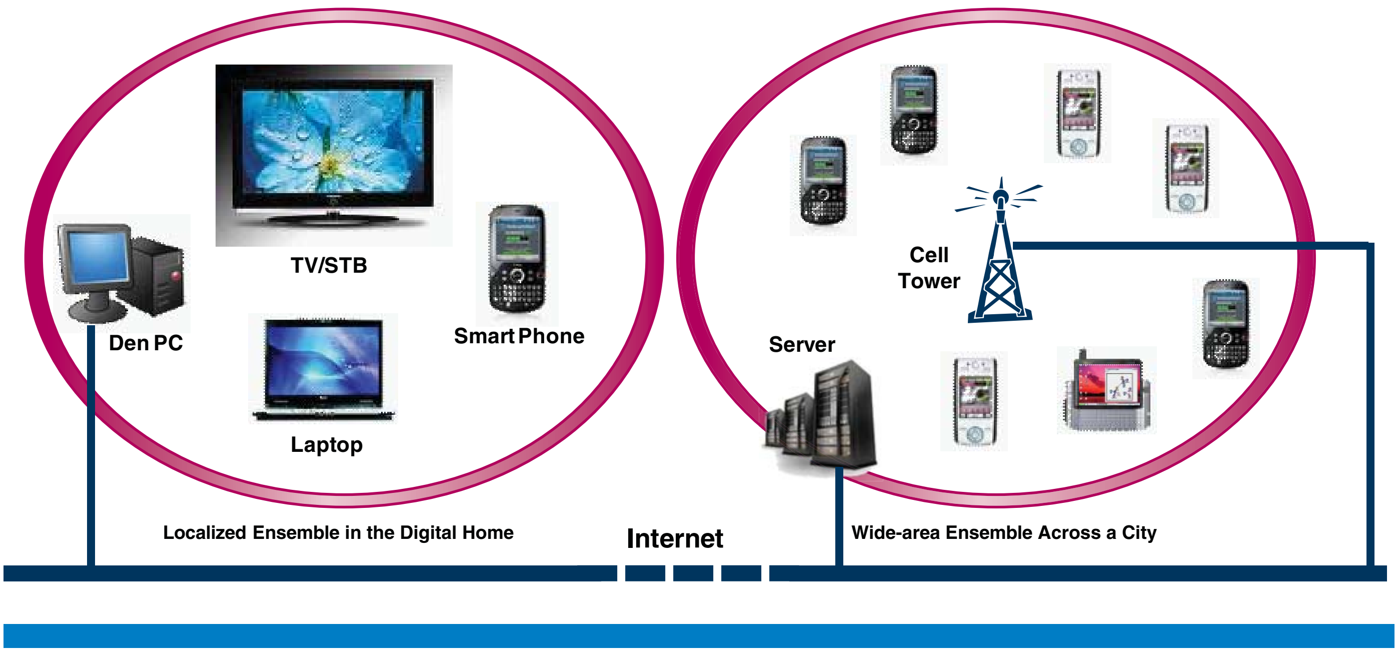

Figure 1: Ensembles of Devices a) Localized: Based on Wireless Local Area Network (WLAN), b) Wide-area: Based on Wireless Wide-area Network (WWAN), Properties: Dynamic Configuration/Heterogeneous/Context-aware
Source: Intel Corporation, 2010

Ensembles can be created for different scales of use, primarily determined by geographic separation. In close proximity (localized ensembles—Figure 1a), a computer can be enhanced by dynamically adding new wireless peripherals, e.g., a wireless headset. Alternatively, computers can mutually share their own system resources such as displays, networks, and disks. Processor sharing between devices is also possible through the migration of applications encapsulated in a virtual machine image, migrating them between low- and high-performance computers. Sharing resources in this way effectively builds logical computers by wirelessly piecing together new components on the fly, thus enabling a user to overcome the design limitations of their own computer by using whatever can be found nearby [4]. In localized ensembles, proximity is a prerequisite for components that support a user interface (e.g., display, camera, keyboard, mouse), as it only makes sense to share these components when they are physically accessible to a user.

"Sharing resources in this way effectively builds logical computers by wirelessly piecing together new components on the fly."

In contrast, ensemble applications based on highly distributed sets of heterogeneous devices may take advantage of geographic diversity (wide-area ensembles—Figure 1b), while still dynamically adapting their membership and taking advantage of their context of use. This information can be used to create ensemble-wide inferences based on their physical separation that would have been impossible from a localized viewpoint. For example, a smart-phone vehicular traffic-monitoring application [5] can be designed so that each smart phone enrolled in the task is programmed to periodically relay information about its location to a server, which in turn builds up a real-time view of traffic flow across all the city highways, and thus can highlight areas of congestion. The traffic-service relays this information back to each of the smart phones running a traffic-monitoring client, thus enabling users to make decisions about their preferred driving routes. The resulting application has greater utility than would be possible with a set of contextually isolated devices originally designed for the primary task of making phone calls.

"In the case of WLANs, unlike wired networks, there is no connectivity between any of the peer layers in the protocol stack until a layer-2 discovery process has taken place."

Localized and wide-area ensemble applications of the types described here exist today, but there is no consistent programming strategy or architecture that can be used to capitalize on lessons learned from their design and implementation. This leaves engineers to reinvent the wheel across many of the application domains. Furthermore, these lessons are rarely captured in the tool-chains used to implement them, and therefore developers do not progressively build better tools for each generation of system design.

In the remainder of this article we examine background and related work, enabling technologies and opportunities for ensemble computing, and we discuss the challenges that need to be addressed.

Background

Distributed programming has been a subject of research and development since the 1970s with many well-understood programming models developed and in common use today.

Although wireless ensembles can make use of many proven distributed-computing abstractions such as sockets, remote procedure call (RPC), message passing, streaming, and client/server models in common use today, they differ in several important ways. In the case of WLANs, unlike wired networks, there is no connectivity between any of the peer layers in the protocol stack until a layer-2 discovery process has taken place, and until a mobile device is instructed to connect (manually or automatically). In the case of WWANs, devices are always connected but peer-to-peer messaging and discovery protocols are restricted by the rules of a particular service provider, limiting the types of protocols in use. However, in either type of wireless network, ensembles of devices can augment the discovery process further by including metadata that can be derived from their current context (e.g., location, or physical sensor data), and device capabilities (e.g., supported communication protocols). Based on these metadata, decisions can be made about the device groupings that can form. Context information can also be derived from user input, automatically sensed, or inferred from the behavior of applications. Shared metadata of this type can define and dynamically modify connections as the context changes. We now look in more detail at some of the solutions that give rise to ensembles in both the local and wide-area environment.

"In the case of WWANs, devices are always connected but peer-to-peer messaging and discovery protocols are restricted by the rules of a particular service provider, limiting the types of protocols in use."

Localized Ensembles

Radio technologies that are designed for short-range operation implicitly define a localized ensemble of devices by the nature of the proximate connections they can establish. The first digital radio standard to achieve this was Bluetooth*, providing an effective replacement for wired peripherals (up to 10 m). Bluetooth's most popular application has been to connect a cell phone to a wireless headset, but the protocol can in fact link up to 8-slave peripherals to a computer; thereby, serving as the master device [6]. Bluetooth application examples include on-body, in-room, and in-car networks supporting an ensemble of devices in those contexts.

Short-range radio systems have recently enabled new opportunities for patient care. Monitoring of chronically ill patients while out of a hospital, and hence mobile (ambulatory monitoring), has been a goal of physicians for many years. In recent years, inexpensive Body Sensor Networks (BSNs) have become viable. This is because power consumption has been reduced to a point where it is possible to perform signal processing, and the results can be wirelessly communicated within a power envelope that can be sustained by a small battery-powered device for several days [7]. BSNs can wirelessly connect a heterogeneous set of sensors through a Body Area Network (BAN) [8] to a central processor that can store, or wirelessly offload, the results to a hospital, for example, through a secondary radio, such as WiFi or a General Packet Radio Service (GPRS) [9]. However, special care needs to be taken with the antenna design, because its baseband frequency is 2.4 GHz, which is readily absorbed by water, and therefore wireless signals cannot travel through the body without being severely attenuated. Sensors, including EKGs, pulse-oximeters, and blood-pressure monitors, are now also readily available as commercial products and can be integrated into a BAN.

"Bluetooth application examples include on-body, in-room, and in-car networks supporting an ensemble of devices in those contexts."

"An ensemble, based on body-worn devices, such as a watch or a cell phone, in combination with environmental monitors in the home, can provide caregivers with the data they need."

Another use of BSNs, other than monitoring the body's vital signs, is to monitor physical activity. In recent years, there has been considerable interest in using technology to allow a larger fraction of the elderly population to remain in their homes as long as possible without needing to resort to institutional care, often referred to as aging-in-place. The problem is that aging may be accompanied by a slow mental decline, and it is often not clear to family and friends when an elderly person becomes a danger to him or herself [10]. An ensemble, based on body-worn devices, and embedded in day-to-day accessories such as a watch or a cell phone, in combination with environmental monitors in the home, can provide caregivers with the data they need. For example, when daily activity patterns change and become increasingly sedentary or erratic, these devices can provide vital clues that a person is no longer behaving in a healthful way, and that she needs further evaluation. The value proposition for the elderly is that in most cases they readily want to be able to stay in their own home as long as they are able, and a BSN provides caregivers, the physicians, and the elderly with the information needed to make this possible.

The WiFi radio standard and its various generations (b/a/g/n) also support a network of proximate wireless devices, but these standards are typically centered on an infrastructure node, or access point. Although these networks can be bridged to build a wireless network of arbitrary size, a common configuration makes use of a single access point serving one location up to a radius of 100 m. For example, in-home networks typically only need one access point, and all devices that discover and connect to it can then communicate with each other. The "digital home" is another example of where a coordinated ensemble of devices can better serve a user than a set of isolated devices. By coordinating the access and playback of digital media captured and consumed on a wide variety of consumer devices, a user can see or listen to their media on a device that is most convenient, or has the best physical capabilities (e.g., size, resolution, and fidelity).

Smart spaces, or *smart rooms*, are a fertile area for experimentation that make use of the tenets of ensemble computing. The essence of a smart space is one in which local infrastructure is readily made available as a service to users and their mobile computing devices. Usually, this is in the form of support for multimedia capabilities to enhance meetings and presentations, but may extend to highly-specialized capabilities, depending on the organization that developed it. Many of the concepts that underlie smart spaces are exemplified in the EasyLiving project at Microsoft [11], and the Stanford iRoom project [12]. For example, the iRoom supports multiple, wall-sized screens that can be linked together into one contiguous display, and at the same time, is wirelessly integrated with laptops and other heterogeneous mobile devices brought into the space. The system can also be commanded through tangible objects designed to be manipulated, or gestured with, to provide intuitive user controls. This smart space demonstrated effective use of the ensemble computing concept in support of group collaboration. The EasyLiving project had similar properties, but also made use of cameras and image processing to infer activities going on in the smart space.

"The essence of a smart space is one in which local infrastructure is readily made available as a service to users and their mobile computing devices."

Wide-Area Ensembles

Ensemble computing has its origin in Grid [13] and Cloud Computing frameworks [14]. These techniques can be used to support ensemble applications, and it is likely that new programming paradigms will be discovered that further enhance the value of ensembles in the future. However, many ensemble applications can also be built by using standard distributed computing techniques that are augmented to take advantage of mobile properties such as context.

"Ensemble computing has its origin in grid and cloud computing frameworks."

Social networking for mobile devices is an example where distributed computing merges with mobile and context-aware computing and clearly defines value for device ensembles. Consider some of the latest location-based computing applications, such as *Loopt** or *Overhere** that run on an iPhone*, and use cell-tower-based (or GPS-based) position finding. In these systems, a group of friends decide to share their location information, by allowing their devices to be tracked. Effectively, an ensemble of devices has been declared that is a proxy for group members. The resulting system lets you know if your friends are nearby so that you can serendipitously meet up over coffee or share a meal. In the case of Loopt, you can also periodically keep track of your friends' status through secondary services such as Twitter* notifications.

Sensor Networks are another example of wide-area ensemble computing in which the underlying goal is for an ensemble of devices to work together to share environmental data in order to make sense of them, and understand larger scale trends [15]. Examples include micro-weather monitoring in order to make better weather forecasts, seismic activity monitoring to understand the effects of city-wide earthquake damage on buildings and structures, and wide-area pollution monitoring to advise on travel. Many sensor networks are composed of highly customized nodes (often referred to as motes) that sense and relay information across a wireless mesh of devices towards a collection point, or sink. From there it is routed to more capable processors that can store and analyze the data, and in turn infer the dominant trends. Through network servers the results can be redistributed to users running client programs, thereby allowing them to take appropriate action. An example of a sensor network in action is the Common Sense project [16] in Intel Labs that provides users with a device to monitor atmospheric pollution based on carbon monoxide, nitrogen oxides, and ozone. The entire system comprises both a local ensemble and a wide-area ensemble. Locally, pollution measurements are combined with location information from the monitor hardware via Bluetooth to a cell phone. Then by using GPRS, the cell phone periodically transmits these data over the wide-area network to a server where the information is aggregated; thereby, providing a service that supports all of the participating phones. In this way, users can be advised about travel options, and the airborne pollution levels currently contaminating particular routes. Such a system becomes valuable when a larger number of devices are participating in parallel across a large distributed area.

"Many sensor networks are composed of highly customized nodes (often referred to as motes) that sense and relay information across a wireless mesh of devices towards a collection point."

Enabling Technology and Trends

Ensembles have been enabled by several technology trends, each advancing the capabilities of mobile devices. The key advances are increasing processor performance (Figure 2a) at lower power, increasing portable memory density (Figure 2b), and increasing wireless networking bandwidth (Figure 2c), and all at smaller form factors for each new generation of device.

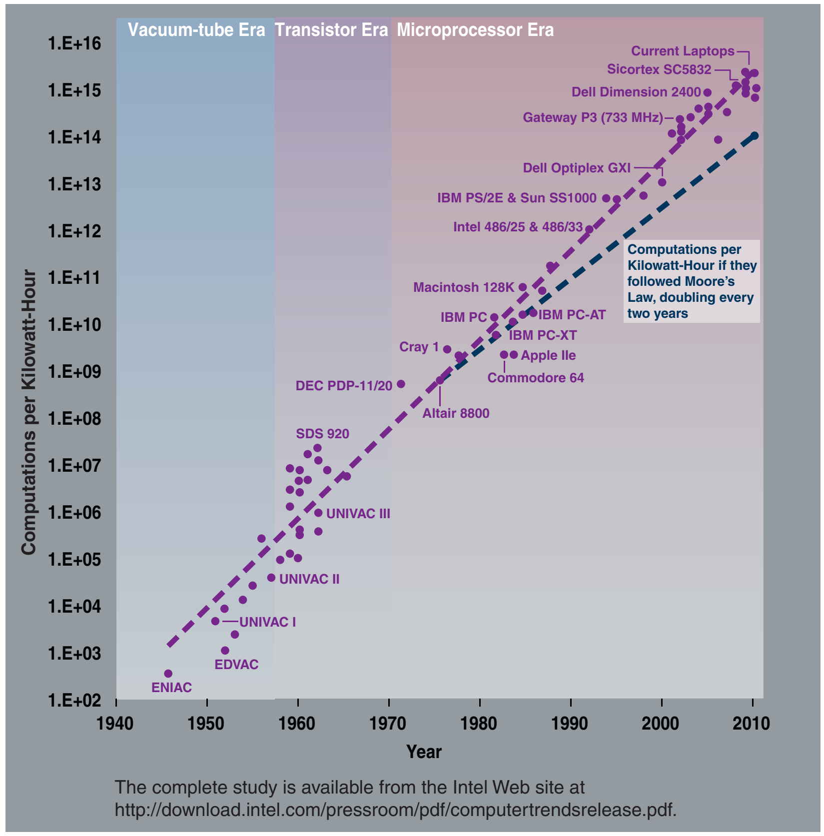

Figure 2a: Trends in Processor Instructions per kWhr by Year
Source: Intel Corporation, 2010

Figure 2a shows the dramatic progress that has been achieved in increasing processor performance in terms of ever increasing numbers of instructions that can be performed with the same amount of energy. The implication is that low-power processors can do more each year given the amount of energy that can be stored in a small portable battery—a technology whose energy density is improving at a far lower rate.

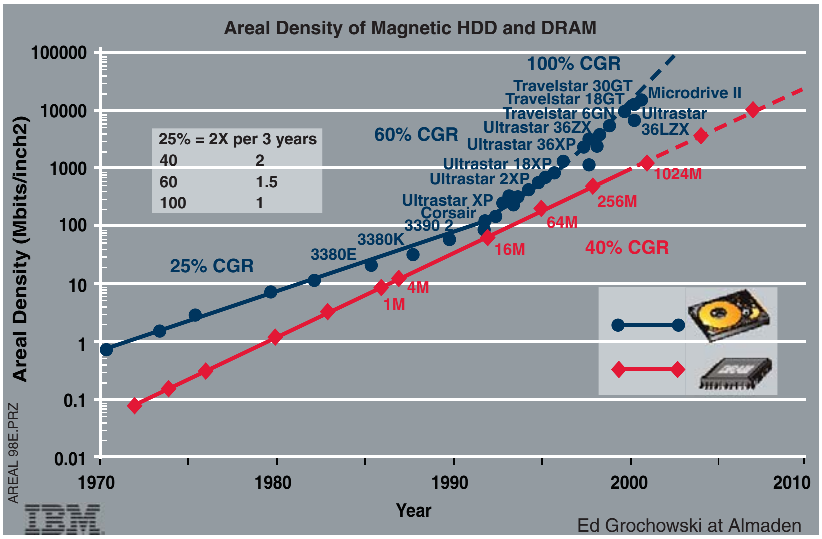

Figure 2b: Trends in Magnetic and Semiconductor Storage Density by Year
Source: Intel Corporation, 2010

Further, the yearly rate of increase in storage density for magnetic media, a rate which has been out-pacing Moore's Law, has been approximately doubling every year (Figure 2b), and this increase has also been a key enabler for ensembles. As a result, small mobile devices are able to store the gigabytes of data required by modern operating systems and their applications, and at an affordable price point. Today, many mobile computers are as capable as some of the most powerful desktop computers in use ten years ago while also running comparable operating systems.

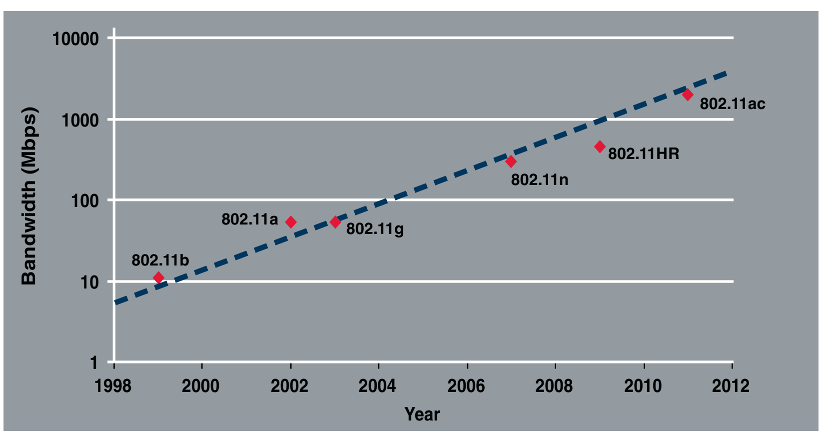

Figure 2c: Trends in Wireless Bandwidth for WLANS by Year
Source: Intel Corporation, 2010

Finally, inter-device communication, an essential component of ensemble computing, has also improved dramatically. It was only in 1999 that the first draft of the Bluetooth standard and the first implementations of the IEEE 802.11b WiFi [17] (11 Mbps) standard were established. Today, 802.11HR approaches 450 Mbps, with further extensions planned (Figure 2c). Wide-area, wireless-data transport services such as GPRS, and later 3G [18], also became available during this time. Each technology supports communication at different geographic scales and bandwidth and can be classified into various types of networks now commonly available to support ensemble computing:

- Personal Area Networks (PANs) connecting computers and peripheral components over a short range (10 m) e.g., Bluetooth 1.0 (1 Mbps).
- WLANs connecting computers to local infrastructure and other nearby computers (100 m) e.g., WiFi/802.11g (54 Mbps).
- WWANs connecting mobile computers to cell towers networked across a city, e.g., GPRS (2000 m) 20/80kbps.

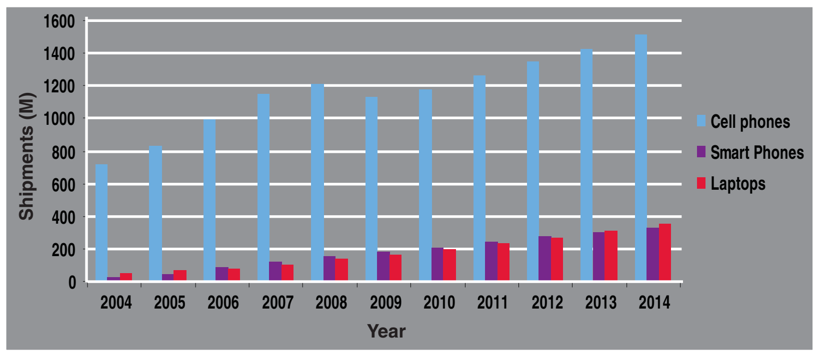

Figure 3: Shipments of Mobile Devices by Year: Actual 2004-2009 and Forecast 2010-2014
Source: Intel Corporation, 2010

Furthermore, the value proposition for mobile users has continued to increase; i.e., more capability at lower cost, resulting in the spectacular increase in market size for cell phones, smart phones, and laptop computers that we have witnessed during the last five years. Moreover, the trend is still forecasted to continue despite the recent recession in 2009 (Figure 3). The large-scale proliferation of these devices has become a key enabler for a wide variety of ensemble applications.

Opportunities

In this section we consider three major opportunities for realizing greater value for ensemble solutions in the following areas: convertible resources, super-charging performance, and added value from new emergent behaviors.

Convertible Resources

Although short-range wireless standards provide a core capability to enable localized ensembles, a higher-level construct is needed to take advantage of them. The Dynamic Composable Computing (DCC) project at Intel [4] is one model for overcoming the resource limitations of a computer, especially a small mobile device, augmenting its capabilities by utilizing the resources of more capable devices nearby. The model for DCC is to abstract away the underlying devices by using a client/server service model enabling a remote peripheral to appear as if local. There are several industry solutions to enable network access to remote displays, such as Virtual Network Computing (VNC) [19], storage [20], and USB peripherals [21], but none that coordinate all these services into a unified system. The DCC composition framework utilizes many existing solutions for a category of peripherals, advertising their presence, and coordinating operations for the user via an intuitive GUI. The opportunity for composable systems is for every computer to be able to convert itself into a much more capable device by using the best of the available resources nearby.

"The DCC composition framework utilizes many existing solutions for a category of peripherals, advertising their presence, and coordinating operations for the user via an intuitive GUI."

Super-Charging Performance

It is also possible to use high-performance computers to augment the processing capabilities of nearby small mobile devices, thus going beyond sharing system peripherals, to sharing a processor among an ensemble of devices. However, unlike peripheral sharing mechanisms such as VNC, super-charging processing has not been widely explored to date in support of mobile computing.

One of the most promising approaches to enable this capability is in the realm of virtual machine (VM) migration. It has only recently been practical to run a VM on a handheld mobile computer with the advent of low-power mobile x86 processors (i.e., the Intel® Atom™ family of processors). Mobile Internet Devices (MIDs) and Netbooks, both based on the Intel Atom processor, also run standard operating systems based on Windows* and Linux*. The next generation of Intel Atom processors will target smart phones, and run standard operating systems by using x86 applications without modification. The resulting common instruction-set allows the same code to run on mobile computers that are based on the Intel Atom processor, as it does on most desktop computers, and thus standard VM hypervisors, and the associated VM migration techniques, can be employed to support the migration of computation between the two.

"The next generation of Intel Atom processors will target smart phones, and run standard operating systems by using x86 applications without modification."

Consider the benefits of VM migration in an everyday scenario (Figure 4) in which a mobile professional, Jane, wishes to maintain a consistent computing experience as her computing activities flow between home, office, and mobile working. While in her home office, Jane was sending e-mail, surfing the Web, and referring to documents about a product she wanted to research. When mobile, she would like to continue these tasks on her smart phone even while waiting for a friend in a parking lot. Later she finds the product in a store and takes some pictures with her mobile so she can study it more before buying it. As is often the case with smart phones, the camera has poor resolution, and so she employs a super-resolution [22] application in which several pictures are taken in quick succession, and then they are combined to create a higher resolution picture (a compute- and power-intensive activity).

This scenario can be implemented by running Jane's entire office PC's computational state in a VM that can be suspended, and then transferred to her smart phone where it is resumed. While mobile, her e-mail and documents can be accessed as before, and the mobile phone's camera can be used to take photos. When returning home, or at her workplace, her smart phone can suspend its VM again and transfer it back to a desktop PC where it can resume and execute at higher performance. In this example, VM migration enables effective mobility by providing session transfer and by super-charging processor performance for CPU- and power-intensive operations. By utilizing a mains-powered desktop computer, it is possible to reduce program execution time on the smart phone, while also saving power on a battery-constrained device.

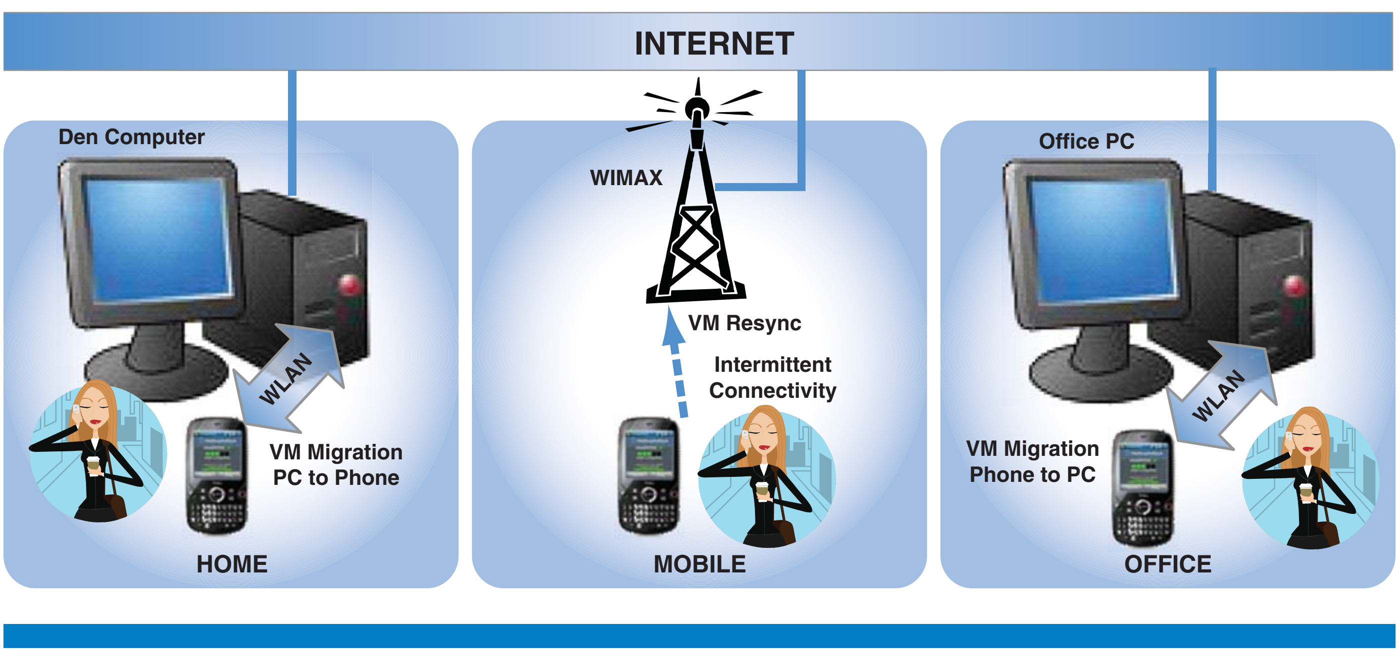

Figure 4: A Smart Phone Migrating Processing Between Home, Office PC, and Mobile Use, Encapsulated in a Virtual Machine
Source: Intel Corporation, 2010

Smart phones, based on the Intel Atom processor, make this mobile use model a reality. In a collaboration between the University of Toronto and Carnegie Mellon University (CMU), the Horatio system [23] has been developed expanding the earlier work on Internet Suspend Resume (ISR) [24] to include migration to a smart phone. This system demonstrates the opportunity for using either a network server, or a mobile device, to transfer a VM image to a secondary PC. However, the first version of the Horatio system used Advanced Risk Machines (ARM) based smart phones and therefore could not resume the VM image while mobile. In a few years, x86 smart phones will become more common, thus enabling the more complete use model.

"VM migration is not a practical option as the large image size (4-20GB) results in transfer times that would likely take too long over a WWAN or other long-haul network."

In addition to local infrastructure, cloud computing [14] offers another opportunity to super-charge the execution performance of a mobile device. In this case computation can be moved from a mobile to a high-performance computing (HPC) center. However, VM migration is not a practical option as the large image size (4-20GB) results in transfer times that would likely take too long over a WWAN or other long-haul network. On the other hand, ensembles that are designed for specialized tasks may include HPC services, in combination with mobile devices that send only selected data elements. These might include media, documents, and context sent to services in the cloud for processing, e.g., photos sent to the cloud for image identification, or documents translated form one language to another. In both cases these services are improved by utilizing context knowledge about the media's creation, which can be provided by the ensemble.

Added Value from Emergent Behaviors

Although the number of compute-capable devices in use is predicted to reach 15 billion by 2015 [25], and the number of devices connected to the Internet will more than double from 2010 to 2013 [26], the sheer availability of devices to communicate does not equate with their intelligent coordination. The opportunity to harness the collective resources of multiple devices has been maturing for some time, yet the opportunity to harness the collective *wisdom* of groups of devices awaits us.

"Increasingly, devices find themselves in a sea of other devices, each of which has the ability to measure its own behaviors, to sense the environment or context in which it is operating, and to share its findings with other devices."

Increasingly, devices find themselves in a sea of other devices, each of which has the ability to measure its own behaviors, to sense the environment or context in which it is operating, and to share its findings with other devices. As a consequence of gossiping data, not only are devices able to build local approximate images of global system state, but also to exhibit global properties that emerge from local properties. In these emergent collaborative systems, metrics have been shown to improve with the size of the group. In the security realm, ensembles of end-hosts collaboratively defend the enterprise network, offering improved accuracy and sensitivity of malware detection [27]. In the wireless world, ensembles have been used to great effect to mitigate wireless interference [28]. Additionally, in the arena of neighborhood-level networking, distributed trust models have demonstrated the simplification of self-configuration [29].

Challenges

Despite the many attractive opportunities, there is a variety of challenges when applying ensemble concepts to practical problems. Here we consider difficulties in manageability, new programming models, wireless network limitations, usability, power, and security and privacy.

Manageability Challenges

One challenge often faced by wide-area ensembles is the stringent requirement for homogeneity, i.e., that all machines agree on the same software, protocols, formats, and versioning. Thus, machine federations are often managed by one owner, who isolates the solutions from one another and can thwart efforts to support a more broadly generalized service [30]. However, proprietary solutions often reduce security and privacy concerns that arise when device federations span multiple owners and/or multiple administrative domains, even though they may not have the most effective solutions.

With the proliferation of mobile and embedded devices, there is an opportunity to revisit how we manage them. In the absence of an IT department, computers are at liberty to bootstrap services from nearby [4, 31] or like-minded devices [32, 33, 34], and/or from reputable peers [35], rather than from preconfigured servers or proxies. In other words, ensembles can take advantage of peer-to-peer mechanisms in a context where no infrastructure exists. However, ensembles also underscore the challenges inherent in establishing device trustworthiness [36], and the challenge of preserving a device's identity along with its data (see later section on Security and Privacy). For wide acceptance, these issues need to be resolved, and the level of sharing needs to be adapted to the context before enterprise IT departments will allow their mobile computers to participate in generalized ensemble applications.

New Programming Models

Platform-independent programming such as Java*, Flash*, and Silverlight* can be adapted to ensemble programming in which applications run across devices with different operating systems and heterogeneous hardware platforms. However, ensemble computing is inherently distributed across multiple devices (unlike the traditional client-server model), and it should dynamically adapt to the presence of nearby resources. This environment, unlike the static configuration of wired computer networks, extends existing platform-independent programming models that currently may not suffice to take advantage of all the potential benefits of an ensemble.

"Ensemble computing is inherently distributed across multiple devices and it should dynamically adapt to the presence of nearby resources."

We envision that requirements for new programming models to support ensemble systems will bear similarities to those for data-parallel technologies, such as Globus* [37] or MapReduce* [38] with additional support for running algorithms efficiently on an unreliable, heterogeneous collection of mobile devices. For instance, when a failure occurs e.g., a participating device suddenly leaves the area, the departing code should be re-executed on a device that is still present.

Second, context-aware programming models [39, 40] can also be useful to write ensemble applications. These programming models allow developers to define tuples that connect a current context, derived from sensor data, to a corresponding action that allows programs to naturally adapt to their environment, and to the dynamically-changing membership of an ensemble. A friend-finder application is a good example as it uses sensed location-data to initiate an action that notifies users that are near one another in a city.

Wireless Limitations

While wireless connectivity provides great benefits in convenience and usability, it also presents significant challenges in reliability and form factor.

Reliable Connectivity

The reliability of a connection between ensemble devices is impacted by a number of factors in the ensemble's environment. Physical objects can block the wireless signal between ensemble devices and drop the connection, much like a dropped cellular call. When ensembles include mobile nodes, reliable communication can be problematic. Also, radio frequency interference (RFI) may be caused by other wireless devices within range of the ensemble. The household microwave oven is a typical example, contributing interference in the license-free 2.4 GHz band, in which most home and consumer wireless devices around the world operate. For wide-area ensembles there are a larger number of other devices to cooperate with (and interfere with) than for local-area ensembles, and this situation can also lead to potential security risks (see Section on Security and Privacy). Furthermore, large distances between ensemble devices can affect the quality and reliability of connection. Devices that connect near the limits of their range must process weaker signals that are more difficult to decode reliably. Research on Delay Tolerant Networking [41] suggests techniques to mitigate communication link failure and boost reliability in these situations.

"Achieving greater range typically requires a more powerful transmitter that translates to larger batteries, bigger antennas, and a larger physical device."

Another approach for increasing the reliability of wireless communication is to design devices, or infrastructural communication hubs, with greater operational range. However, achieving greater range typically requires a more powerful transmitter that translates to larger batteries, bigger antennas, and a larger physical device. These factors are a challenge when designing mobile devices, resulting in a compromise of utility versus form factor.

Form Factor

By definition, given the dynamic nature of ensembles and group formation, a significant number of participating devices are likely to be highly mobile, with small user-friendly form factors. These ultra-mobile designs provide significant challenges when accommodating the antenna and power requirements of a radio subsystem that provides discovery and connectivity. High-capacity batteries that can support high-power radios, or long transmission times, incur additional device weight and size. Increasingly, mobile devices also support several communication technologies (3G, BT, WiFi) in the same package, operating at a variety of frequencies. In turn, this increases the complexity of antenna designs, also requiring shielding between each radio, and these factors contribute to larger packaging requirements for a mobile device.

Usability Issues

"Devices that transmit weak signals may frequently drop connections with other ensemble devices, thereby causing lost data and slow response times."

The broad availability and spectrum of devices that can form an ensemble, ***and do so with minimal user intervention***, will dictate the usability and pervasiveness of the technology. A successful wired precedent came with the introduction of the universal serial bus (USB), which allowed users to painlessly transfer data between devices. Notably at CES 2010, manufacturers started showing concepts for wirelessly pairing mobile devices with Consumer Electronic (CE) equipment (such as TVs) to share content and execute applications. We can easily imagine this notion evolving to include ensembles between (and within) automobiles, peripherals, and home appliances. Key ingredients for making an ensemble computing paradigm work well are minimal and consistent user-interfaces (UI), preference-based configuration and intelligent set-up, and context-based adaptation and connection. Further, in an optimal world, these would all require minimal user intervention to discover, connect, and share resources among participating computers.

Some of the primary advantages of wireless systems also create additional challenges in usage compared with wired networks. As described in the section on Wireless Limitations, devices that transmit weak signals may frequently drop connections with other ensemble devices, thereby causing lost data and slow response times, and these further limit connection quality, throughput, and reliability. These disruptions interrupt natural usage that negatively affects the user experience. Wireless device connections lack the visual physical indicators (i.e., a visible cable) of wired connections.

The extent of secure communication is also unclear, as eavesdropping on a wired network requires physical access to the cable, whereas eavesdropping on a wireless network only requires that the device be in range. Because wireless is invisible, these issues need to be addressed through connection status and configuration indicators to guide and support the usability model for a device and the ensemble. User interface standards in these areas have not been developed at this time. Furthermore, if a device is connected to a wire, it can be designed to be powered by the same wire, simplifying the form factor design constraints, or when an internal battery is present, providing a mechanism for charging the device. Ensembles are less likely to benefit from this design approach.

Power Constraints

In the case of ensembles, where value is maximized by the seamless orchestration of multiple devices, power becomes critical.

"Eavesdropping on a wireless network only requires that the device be in range."

We have all experienced the frustration of using a cell phone in a car and the phone battery dies when you're in the middle of getting directions. In this mobile situation we are further frustrated because we cannot easily use our hands to plug in a charger and redial. The reality today is that we increasingly rely on our mobile battery-powered devices to be continuously available and provide critical communication, content capture, I/O control, and storage. As the interdependency, and thus criticality, of these devices continues to grow, the expectation that they are powered and available increases. The solution is not simple; moreover, the problem is exacerbated by our expectations of ever-shrinking form factors, and the desire for minimal dependency on plugging in. However, devices can adapt when they join an ensemble, for example, by turning off unnecessary duplicate peripheral capabilities already provided by peer devices across the wireless connection.

Mobility of Tasks within Ensembles Based on Power Availability

One of the advantages of an ensemble is the opportunity for reliability through redundancy. In the dropped phone-call example just cited, there is a potential solution. By detecting the cell phone's low-power state and seamlessly rerouting the phone call (including headset re-pairing) to a VOIP [42] system running on a laptop in the car, an automatic handover could be achieved. Characterization and discovery of device capabilities in an ensemble, and the ability to cross device boundaries to take advantage of opportunities such as more available power, are key capabilities for enabling this approach. Techniques such as VM migration (described earlier) can also provide the infrastructure needed for process migration to devices that have sufficient power to complete a task.

"By detecting the cell phone's low-power state and seamlessly rerouting the phone call (including headset re-pairing) to a VOIP system running on a laptop in the car, an automatic handover could be achieved."

"To form an ensemble, mobile devices should be able to discover other ensemble-capable devices as they enter mutual communication range."

Mobility of Physical Power among Ensemble Members

An alternate approach uses power as a resource to be shared among devices. In this scenario, power is transferred to the most critical part of the ensemble. To continue with the dropped call on the cell phone example, the ensemble detects the low-power state of the cell phone, and it moves power from the user's laptop, or even the car itself, to the cell phone. Users can do this today under some conditions by using a USB cable; other technologies, such as wireless power, are under development. Wireless charging through inductive coupling or spatial (longer range) technologies will increasingly be adapted and improved for use in commercial applications. However, it is only likely between larger power-rich devices, and nearby low-power mobile computers, due to the current inefficiency of free-space power-transfer.

Security and Privacy

Securing an ensemble system as a whole requires more effort than securing individually participating devices. In this section we discuss security and privacy issues that are uniquely associated with implementing and running ensemble systems.

Ensemble Formation

To form an ensemble, mobile devices should be able to discover other ensemble-capable devices as they enter mutual communication range. For example, an access point periodically broadcasts its SSID to allow other 802.11 clients to find the resource (and vice versa). While this type of beaconing is commonly used for automatic resource discovery, the discovery protocol could inadvertently expose the device's identity to attackers who snoop on broadcast traffic for the purpose of tracking the mobility patterns of each device. An approach for securing discovery protocols is to employ cryptographic algorithms to limit such risks [43, 44].

"A new approach to control the use of private sensitive data is via a trusted virtual domain accessible to the remote parties. Once an ensemble dissociates, any cached code or data in the trusted domain can also be deleted for added system security."

Once nearby devices are located, the next step is to establish trust between the devices before executing coordinated tasks. Given that many usage cases assume little or no institutional support for the ensemble infrastructure, both certificates and trust verification processes used in traditional networks would be impractical or too heavyweight to be applied to ensemble systems. However, existing solutions for mobile ad-hoc networks [45] could be adapted to establishing trust with ephemeral devices, but the heterogeneity of participating devices complicates distributed key management.

Ensemble Computation

If sensitive data are involved in an ensemble computation, or the code implementing applications needs to be protected from reverse engineering, it is possible to build an additional layer of security leveraging trusted hardware. For instance, a new approach to control the use of private sensitive data is via a trusted virtual domain accessible to the remote parties [46]. Once an ensemble dissociates, any cached code or data in the trusted domain can also be deleted for added system security.

Another challenge for ensemble system security is to develop dynamic security policies that can adapt to available power, platform resources, and the collection of devices present. However, the security of the ensemble is only as strong as the weakest member device; hence, striking the right balance between flexibility and security is important to secure the entire system.

Conclusion

Ensemble computing extends the general notion of distributed computing with the issues that arise from introducing collections of mobile devices into the mix. Key differentiators are the dynamic nature of the connections between devices, the use of location and other context information, and the heterogeneous nature of devices and their processing and system resources.

As a result, there are many opportunities that arise from the existence of ensembles that include converting (or augmenting) a computer by sharing remote wireless resources. Alternatively, processing can be super-charged by migrating computation from under-powered mobile devices to higher-performance infrastructure computers. Emergent behaviors in an ensemble may also solve problems that cannot be solved locally: some solutions will only be found by taking advantage of the shared wisdom of large numbers of people.

"Many opportunities arise from the existence of ensembles that include converting (or augmenting) a computer by sharing remote wireless resources."

Ensemble systems also face many challenges: managing large numbers of devices, defining new contextual programming models, living with the limitations of wireless communication, designing intuitive user interfaces, mitigating power constraints, and all the while ensuring security and privacy. These problems are not intractable, and there are many promising solutions that will move us forward in this field.

Ensemble systems that exploit the available opportunities while finding new ways to effectively handle the challenges are an ongoing subject of research. At Intel Labs we continue to define new research projects that explore these issues. Time and time again, we find examples in which the aggregate value of ensembles is greater than the simple sum of the component parts.

References

[1] Weiser, M. "The Computer of the 21st Century." *Scientific American.* September No. 9, pages 66-75, 1991.

[2] Shapiro, C. and Varian H.R. 1999. "Information Rules." *Harvard Business Press.* ISBN 087584863X. [Note: the original work was not archived: this reference cites a presentation in 1980]

[3] Schilit, B. and Sengupta, U. "Device Ensembles." *IEEE Computer*, Volume 37, No. 12, pages 56-64, 2004.

[4] Want, T., Pering T., Sud S., Rosario B. "Dynamic Composable Computing." *ACM HotMobile*, Silverado Resort, Napa Valley, California, 2008.

[5] Lunde, T. and Larsen, A. "KISS the Tram: Exploring the PDA as Support for Everyday Activities." In *Proceedings of the Third International Conference on Ubiquitous Computing*, pages 232-239, 2001.

[6] Haartsen, J. C. "The Bluetooth Radio System." *IEEE Personal Communications*, pages 28-36, 2000.

[7] Yang, G. "Body Sensor Networks." *MA: Spring Science & Business Media.* ISBN 0-84628-272-1, 2006.

[8] Hanson, M. A., Powell Jr., H. C., Barth, T. T., Ringgenberg, K., Calhoun, B.H., Aylor, J.H., and John Lach. "Body Area Sensor Networks." *University of Virginia, IEEE Computer*, pages 58-65, January 2009.

[9] Samjani, A. "General Packet Radio Service (GPRS)." *IEEE Potentials*, pages 12-15, April/May, 2002.

[10] Philipose, M., Fishkin, K., Perkowitz, M., Patterson, D., Fox, D., Kautz, H., and Hahnel, D. "Inferring activities from interactions with objects." *Pervasive Computing*, 3(4), pages 50-56, 2004.

[11] Brumitt, B., Meyers, B., Krumm, J., Kern, A. and Shafer, S. "EasyLiving: Technologies for Intelligent Environments." *Handheld and Ubiquitous Computing (HUC)*, Bristol, UK, pages 97-119, September 2000.

[12] Fox, A., Johanson, B., Hanrahan, P. and Winograd, T. "Integrating Information Appliances into an Interactive Workspace." *IEEE Computer Graphics and Applications*, Volume 20, No. 3, pages 54-65, May 2000.

[13] Foster, I., Kesselman, C. and Tuecke, S. "The Anatomy of the Grid: Enabling Scalable Virtual Organizations." *International Journal Supercomputer Applications*, 15(3), 2001.

[14] "Amazon Elastic Compute Cloud." (Amazon EC2). Available at ***http://aws.amazon.com/ec2/***

[15] Akyildiz, I. F., Su, W. and Sankarasubramaniam, Y. "A Survey on Sensor Networks." *IEEE Communications*, pages 102-114, August 2002.

[16] Dutta, P., Aoki, P. M., Kumar, N., Mainwaring, A., Myers, C., Willett, W. and Woodruff, A. "Common Sense: Participatory Urban Sensing Using a Network of Handheld Air Quality Monitors." (demonstration). In Proceedings *SenSys'09*, Berkeley, CA, pages 349-350, November 2009.

[17] "WiFi." *IEEE 802.11*, February 2010. Available at ***http://grouper.ieee.org/groups/802/11/index.html***

[18] 3GPP. "3rd Generation Mobile System Release 4 Specifications." *TS 21.102*. Available at ***http://www.2gpp.org/ftp/Specs***

[19] Li, S., Spiteri, M., Bates, J. and Hopper A. "Capturing and Indexing Computer-based Activities with Virtual Network Computing." *ACM Symposium on Applied Computing*, Como, Italy, pages 601-603, March 2000.

[20] "Samba: An Introduction." 2010. Available at ***http://samba.org/samba/docs/SambaIntro.html***

[21] "Redirector: product from Incentivespro." Available at ***http://www.incentivespro.com/usb-redirector.html***

[22] Pelltier, S. "Acceleration Methods for Image Super-resolution." *Thesis*, Department of Electrical & Computer Engineering, McGill University Montreal, Canada, November 2009.

[23] Smaldone, S., Gilbert, B. Bila, N., Iftode, L., de Lara, E. and Satyanarayanan, M. "Leveraging smart phones to reduce mobility footprints." In *Proceedings ACM MobiSys'09*, Krakow, Poland, pages 109-122, June 2009.

[24] Satyanarayanan, M., Gilbert, B., Toups, M., Tolia, N., Surie, A., O'Hallaron, D. R., Wolbach, A., Harkes, J. Perrig, A., Farber, D., Kozuch, M., Helfrich, C., Nath, P., Lagar-Cavilla, H. "Pervasive Personal Computing in an Internet Suspend/Resume System." *IEEE Internet Computing*, Vol. 11, No. 2, March/April 2007.

[25] IDC. 2009. Available at ***http://community.edc.intel.com/t5/Applications-Other-Blog/15-Billion-Connected-Devices-Powered-by-the-Embedded-Internet/ba-p/613***

[26] IDC. December 2009. Available at ***http://news.softpedia.com/news/Over-One-Billion-Mobile-Devices-to-Access-the-Web-by-2013-129261.shtml.***

[27] Dash, D., Kveton, B., Agosta, JM., Schooler, E. M., Chandrashekar, J., Bachrach, A. and Newman, A. "When Gossip is Good: Distributed Probabilistic Inference for Detection of Slow Network Intrusions." *AAAI*, 2006.

[28] Huang, J., Berry, R, and Honig, M. "Distributed Interference Compensation for Wireless Networks." *IEEE Journal on Selected Areas in Communications*, Volume 24, No. 5, pages 1074-1084, May 2006.

[29] Zage, D., Livadas, C. and Schooler, E. M. "A Network-Aware Distributed Membership Protocol for Collaborative Defense." *International Conference on Computational Science and Engineering*, Volume 4, pages 1123-1130, 2009.

[30] Campbell, R., Gupta, I., Heath, M., Steven, K. Y., Kozuch M., Kunze M., Kwan, T., Lai, K., Lee Y., Lyons M., Milojicic, D., O'Hallaron D. and Soh Y. "Open CirrusTM Cloud Computing Testbed: Federated Data Centers for Open Source Systems and Services Research." HP Laboratories, *HPL-2009-134*, 2009.

[31] "Wi-Fi Direct." *Wi-Fi Alliance*, 2010. Available at ***http://www.wi-fi.org/files/20091019_WiFi_Direct_FAQ.pdf***

[32] "Wi-Di." *Wireless Display*. Available at ***http://www.intel.com/consumer/products/technology/wirelessdisplay.htm***

[33] Scheifler, R. and Gettys, J. "The X Window System." *ACM Transactions on Graphics* (TOG), 1986.

[34] Jakubczak, S., Andersen, D., Kaminsky, M., Papagiannaki, K. and Seshan, S. "Link-alike: Using Wireless to Share Network Resources in a Neighborhood." *ACM SIGMOBILE MC2R*. Volume 12, No. 4, October 2008.

[35] Devine, R. "Design and Implementation of DDH: A Distributed Dynamic Hashing Algorithm." In *Proceedings of 4th International Conference on Foundations of Data Organizations and Algorithms*, 1993.

[36] Chan, H. and Perrig A. "Security and Privacy in Sensor Networks." *Carnegie Mellon University*, IEEE Computer, pages 99-101, October 2003.

[37] Allcock, W., Bresnahan, J., Kettimuthu, R. and Link, J. "The Globus eXtensible Input/Output System (XIO): A Protocol Independent IO System for the Grid." In *Proceedings of 19th IEEE International Parallel and Distributed Processing Symposium* (IPDPS'05) - Workshop 4, 2005.

[38] Dean J., and Ghemawat S. "MapReduce: Simplified Data Processing on Large Clusters." In *Proceedings OSDI*, 2004.

[39] Min, X., Jizhong, Z., Yong, Q., Hui, H., Ming, L. and Wei, W. "Isotope Programming Model: a Kind of Program Model for Context-Aware Applications." In *Proceedings International Conference on Multimedia and Ubiquitous Engineering* (MUE'07), 2007.

[40] Weichang, D. and Lei, W. "Context-aware application programming for mobile devices." In *Proceedings 2008 C3S2E conference*, 2008.

[41] Fall, K. "A delay-tolerant network architecture for challenged internets." In *Proceedings ACM SIGCOMM*, Karlsruhe, Germany, ISBN:1-58113-735-4, pages 27–34, 2003.

[42] Kanter, T., Olrog, C., "VoIP in Applications for Wireless Access." *10th IEEE Workshop on Local and Metropolitan Area Networks*, pages 122-125, 1999.

[43] Greenstein, B., Mccoy, D., Pang, J., Kohno, T., Seshan, S. and Wetherall, D. "Improving wireless privacy with an identifier-free link layer protocol." In *Proceedings ACM MobiSys'08*, Breacon Ridge, Colorado, 2008.

[44] Gruteser, M. and Grunwald, D. "Enhancing location privacy in wireless LAN through disposable interface identifiers: a quantitative analysis." *Mobile Networks and Applications*, Vol. 10, No. 3, pages 315-325, 2003.

[45] Eschenauer, L., Gligor, V. and Baras, J.S. "On Trust Establishment in Mobile Ad-Hoc Networks." *Security Protocols, Lecture Notes in Computer Science*, 2002.

[46] Löhr, H., Sadeghi, A., Vishik, C., and Winandy, M. "Trusted Privacy Domains – Challenges for Trusted Computing in Privacy-Protecting Information Sharing." *Lecture Notes in Computer Science*, volume 5451, pages 396-407, 2009.

Acknowledgements

We thank Dina Papagiannaki, Trevor Pering, John Gustafson, Lakshman Krishnamurthy, Anthony Joseph, and Herman D'Hooge for their technical feedback.

Author Biographies

Roy Want is a Senior Principal Engineer at Intel Labs, Santa Clara, California, and Associate Director for XTR. His research interests include mobile and ubiquitous computing, wireless protocols, embedded and distributed systems, automatic identification, and MEMS. Want received his PhD degree in Computer Science from Churchill College, Cambridge, UK in 1983. He is currently the Chair of ACM SIGMOBILE, and a Fellow of both the IEEE and ACM.

Eve Schooler joined Intel in 2005. She is a Principal Engineer in Intel Labs within its new Energy Systems Research Lab, where she is designing an intelligent control plane for Smart Homes and Smart micro-Grids. Previously she led the Distributed Detection and Inference (DDI) project, an effort that focused on collaborative anomaly detection to secure large-scale networks. Her broad interests lie at the intersection of networking, distributed systems, and scalable group algorithm design. Eve served on the Transport Directorate of the IETF, co-founded and co-chaired the IETF MMUSIC working group for many years, and is a co-author of the SIP protocol that is widely used for Internet telephony. Eve obtained an BSc degree from Yale University, an MSc degree from UCLA, and a PhD degree from Caltech, all in Computer Science. Prior to Intel, she held positions at Apollo Computers, Information Sciences Institute (ISI), AT&T Labs-Research and Pollere LLC.

Lenka Jelinek. Lenka's passion is incubating technologies, growing ideas from an inspiration into a market reality. Her most recent focus has been on making power cords obsolete through wireless power. In her prior 15 years of experience at Intel, Lenka led research on intuitive systems and low-power IA platforms. Lenka was a co-founder of the Intel Smart Toy Lab and a lead architect on the award-winning Intel® Play™ QX3™ Computer Microscope. She served as the TA to Andrew Chien, VP and Director of Intel Research. Lenka holds degrees in Electrical Engineering and Computer Science from the University of Wisconsin-Madison.

Jaeyeon Jung received her PhD degree in Computer Science from MIT in 2006, and her Bachelor's and Master's degrees from the Korea Advanced Institute of Science and Technology (KAIST). Her research focuses on many aspects of networked systems and system security. Prior to joining Intel Labs, Seattle, Jaeyeon worked at Mazu Networks and implemented her scan-detection algorithms in Mazu's Profiler product. She is an affiliate faculty member at the University of Washington and KAIST.

Dan Dahle is a Sr. Strategic Architect in Intel Labs responsible for developing and executing strategies for moving Intel research technologies into the marketplace. In the past, Dan led a team chartered with enabling technologies for anytime, anywhere computing. The team also worked in the areas of communications in mobile networks, including Wi-Fi (802.11) hotspots, WiMax, and cellular networks, addressing security, authentication, authorization and accounting (AAA) for safer, simpler, seamless access for mobile clients. Prior to Intel, Dan ran his own marketing consulting company. He started his career with AT&T in product engineering, product management, worldwide sales, and business development. A licensed professional engineer, Dahle holds a BS degree in Mechanical Engineering and Material Science and Engineering from the University of California at Berkeley, as well as an MBA from Indiana University.

Uttam Sengupta is a principal engineer in the Ultra Mobility Group and is responsible for defining security, manageability and virtualization capabilities of handheld platforms. Uttam's other interests include ensemble computing, location & context sensing and related technologies required for a service ready platform. Prior to Intel, Uttam was a principal engineer at Motorola, working in the area of GSM cellular technologies. He received a PhD in computer science from Arizona State University. Uttam is a member of IEEE Computer Society.

Copyright