Mobile Networks and Applications 7, 269–278, 2002

Piecewise Network Awareness Service for Wireless/Mobile Pervasive Computing

LIANG CHENG * and IVAN MARSIC
Department of Electrical and Computer Engineering and the CAIP Center, Rutgers, The State University of New Jersey, 94 Brett Rd., Piscataway, NJ 08854-8058, USA

Abstract. This paper presents a piecewise framework for network awareness service (NAS) for wireless/mobile pervasive computing. We investigate how piecewise consideration of wired and wireless elements of the framework architecture benefits service advertisement and discovery and network-awareness techniques. We also discuss scalability of the NAS framework with respect to platform computing capabilities. The framework is suitable for a wide range of computing devices, from powerful ones with multi-tasking operating systems (OS) to small ones with lightweight OS. Case studies applying the NAS framework to sensor monitoring in home networks and data streaming in pervasive multimedia computing are presented. The analytical results on the performance of the NAS framework in these case studies show that it has significant advantages over traditional network-awareness frameworks in terms of reducing wireless bandwidth consumption and saving battery energy of mobile devices.

Keywords: pervasive computing, mobile computing, network awareness, sensor networks, adaptive applications

1. Introduction

1.1. Pervasive computing and network awareness

Pervasive computing [5,12] is the computing paradigm that enables network devices to be aware of their surroundings and peers, and to be capable of effectively providing services to, and using services from, peers. Advances in data networking and wireless communications, digital-system miniaturization, and novel user-interfaces are driving the research and development of pervasive computing. For example, in home networks [11], intelligent appliances with onboard-computers, from cell phone to microwave, may be wirelessly interconnected with the Internet, able to interact, and anticipate users' needs.

To realize pervasive computing, network devices and applications first need to be aware of network environments. Network awareness is a key component of pervasive computing. In this paper, *network awareness* is defined as the capability of devices and applications to be aware of the characteristics of the network. For example, a rate-adaptive application needs to be aware of the available bandwidth along its communication path. Awareness of dynamic context, such as the geographic location or proximity, may also be of interest, but it is not studied here.

1.2. Wireless/mobile pervasive computing paradigm

Pervasive computing increasingly involves mobile devices with wireless communication facilities. In this paper we only consider such cases, i.e., wireless/mobile devices with wireless network interface cards (NIC) or wireless modems taking part in computing tasks with other fixed and wired devices in the Internet, hereinafter referred to as pervasive computing for simplicity.

Figure 1 shows the generic architecture of the pervasive computing paradigm. The figure shows two main domains: the wireless/mobile domain connects to the fixed wired domain via a wireless link. The former domain may include mobile ad hoc networks (MANET) [18]. A dedicated proxy or gateway as a service point is generally used to enhance the performance of inter-domain communications. There are three reasons for this [19]: (i) the wireless connection quality is too poor to sustain a typical client-server application, (ii) the amount of data transferred to mobile devices must be filtered because of the orders of magnitude difference in bandwidth between the wired and wireless connections, and (iii) portable computing devices have display and processing limitations that must be addressed by the proxy before sending the filtered response to the mobile devices.

1.3. Piecewise design for network awareness

An important consideration in designing network awareness for the pervasive computing is that the cost of acquiring network awareness does not exceed its utility: if not obtained efficiently, network awareness could decrease performance, especially when inter-domain communications are involved, due to the resource consumption overheads in CPU, wireless bandwidth, and battery energy at mobile devices.

Our approach to network awareness takes advantage of the fact that the generic architecture (shown in figure 1) consists of a wired and a wireless part. Thus, all issues related to awareness about an end-to-end communication link are considered *piecewise*, i.e., the wired and wireless parts are considered separately and the end-to-end characteristics are derived by combining the piecewise characteristics, whereas a

* Corresponding author.

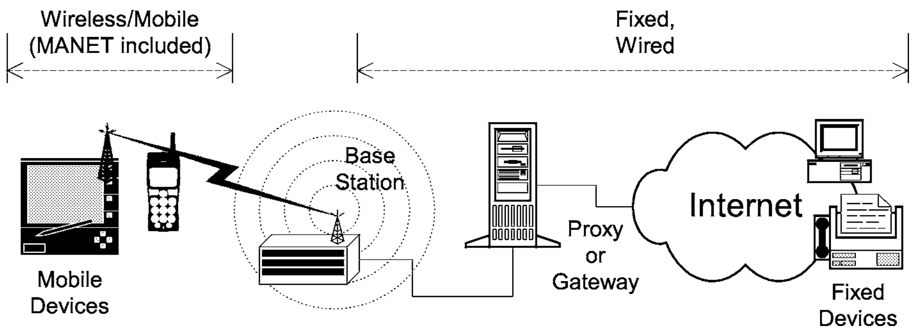


Figure 1. The generic architecture of wireless/mobile pervasive computing paradigm.

unitary approach considers the entire communication path for the awareness purposes as a single piece. The case studies and performance analysis presented in section 3 below show that this piecewise approach enhances the performance of pervasive computing by saving wireless bandwidth and battery energy and by reducing the inter-domain monitoring traffic for network awareness that is transmitted through wireless links. Moreover, piecewise network awareness is designed as a service, so that it is ubiquitously available in a pervasive computing environment.

1.4. Related work

Simple Network Management Protocol (SNMP) is a protocol governing network management and monitoring of network devices and their functions. It is formally specified in a series of related RFC documents, such as RFC 1067 [8], RFC 1212, and RFC 1213. The Network Weather Service [23] is a distributed system that periodically monitors and dynamically forecasts the performance that various network and computational resources can deliver over a given time interval. However, they each realize just part of the network awareness service, i.e., monitoring computing resources. For example, neither can support service advertisement and discovery of other available services in the environment. In addition, they do not take advantages of piecewiseness in the heterogeneous network architecture, which is the main characteristic of our NAS framework.

Network awareness is also studied in the research of adaptive mobile data access. The Odyssey project [16] develops a platform for adaptive mobile data access. In the Odyssey scheme, the operating system is responsible for monitoring resource availability and notifying applications of relevant changes. Each application decides how to best exploit the available resources [17]. However, Odyssey's architectural components reside on the mobile device, in either the user space or the operating system, which could impose a heavy computational load to portable devices with a lightweight OS and limited computing power. As described in section 4 below, our NAS framework is designed to be scalable to both high- and low-power-computing platforms. The network awareness in Odyssey is acquired at the mobile client; in contrast, network awareness acquisition in our framework is piecewise, which offers the advantages analyzed below.

In [22], an architecture is presented for exporting environment awareness to mobile computing applications. A change in the environment is modeled as an asynchronous event and an associated event delivery framework is proposed. Unlike the NAS framework, the environment monitor resides in the mobile device and only the push model is deployed for event delivery. An additional framework for developing network-aware applications is presented in [6] but the techniques for awareness acquisition are not elaborated. The above frameworks are unitary, i.e., not piecewise. Our research is distinguished in that the NAS framework is piecewise and scalable to fit into the scenario of the pervasive computing paradigm with mobile devices of very low capabilities.

The paper is organized as follows. In section 2, we present a piecewise framework of network awareness service (NAS) by which network devices and applications can acquire inter-domain network characteristics efficiently. Case studies and performance analysis are presented in section 3, where multicasting in sensor monitoring and bandwidth-aware data streaming are studied. We discuss the platform-related scalability of the framework in section 4. Finally, section 5 concludes the paper.

2. Piecewise network awareness service

Figure 2 illustrates the NAS framework. The framework advertises the network awareness service. It acquires, measures, integrates, and distributes the parameters that reflect the current characteristics of the heterogeneous data networks. Network applications, protocols, and services that wish to use the NAS can acquire those parameters and apply them to leverage different pervasive computing tasks. The framework is transparent to network applications, services and protocols that do not explicitly use the NAS. It appears to them just as another network application.

2.1. Piecewiseness in framework architecture

The architecture of the NAS framework consists of four main components: the NAS daemon, the NAS data repository, the NAS manager, and the NAS agent. The SNMP (Simple Network Management Protocol) agent and manager in figure 2 are optional. Each piece realizes different functionality.

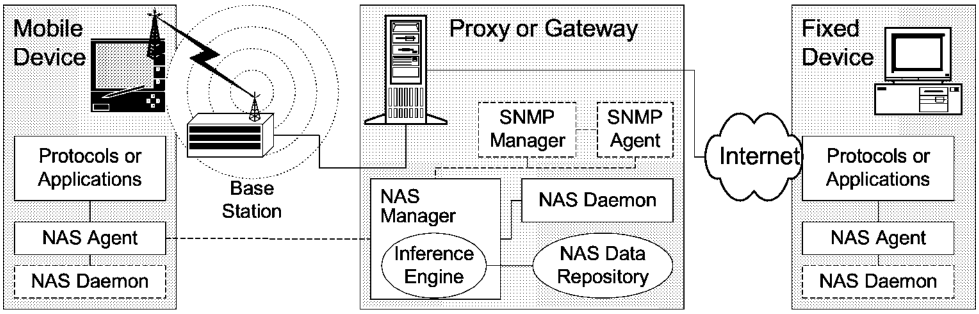


Figure 2. Piecewise NAS framework.

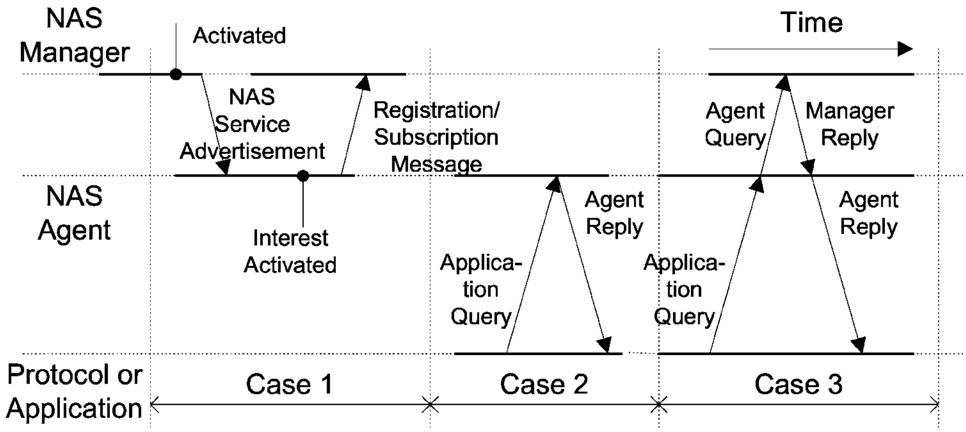


Figure 3. Communications between NAS components.

2.1.1. NAS daemon

An NAS daemon implements different network-awareness techniques, such as a method to obtain wireless-channel parameters or a method to measure the available bandwidth. It is an optional component at the mobile device, subject to resource limitations and the framework's scalability. Any network device with active NAS daemon(s) can be a service provider.

2.1.2. NAS data repository

The NAS data repository is a collection of all data of interest that reflect the current characteristics of the networks. Data are acquired from NAS daemons at the proxy or gateway, collected from the NAS agents at mobile devices, integrated via an inference engine, and distributed by the NAS manager. Some data may be selectively acquired from other service agents, such as SNMP agents and ICMP service. The repository has a hierarchical architecture conforming to the SNMP MIB standard for simplicity, extensibility, and compatibility.

2.1.3. NAS manager

The NAS manager manages the NAS data repository. It collects, integrates, and distributes the NAS data. The manager may communicate with other service agents, such as an SNMP agent or SNMP manager, to acquire additional information that the existing NAS daemons do not provide. Its inference engine integrates data collected from the NAS daemons and the SNMP manager or agent, if available. Also, when the manager receives a query message from an NAS agent, it replies with the corresponding NAS data. Once the NAS manager is activated at the proxy or gateway, it advertises NAS services through a well-known multicast channel defined in the NAS implementation (case 1 in figure 3). The advertisement includes the service name and description. The manager also listens to the service advertisement and service subscription messages from NAS agents.

2.1.4. NAS agent

The NAS agent is the portal for the network entities interested in the services provided by the NAS framework. If the local device computing capabilities permit, the NAS agent will activate all the local NAS daemons. Otherwise it may use a priority-based scheme for agent activation or may not run any local daemons at all, in which case all the service daemons will run at the proxy or gateway. In case there are local daemons active, the agent advertises their services by sending messages to the NAS manager through a well-known multicast channel.

A protocol or application may subscribe to the agent for services announced by the NAS manager in a *push* model of NAS data distribution. On the other hand, the NAS agent is explicitly queried in a *pull* model of data distribution. If the information requested in the query can be obtained from local active NAS daemons, the NAS agent will reply immediately (case 2 in figure 3); otherwise, the NAS agent will relay the query to the NAS manager to perform the awareness task and return the reply (case 3 in figure 3).

The push and pull models can be considered as a tradeoff between simplicity of implementation (pull model) and simplicity of use (push model). Mobility in pervasive computing imposes problems to both models. In the push model, the NAS may have difficulties tracking the roaming user through different networks. In the pull model, due to dynamically varying network performance, the application may not know when the time is appropriate to pull the information. The push model provides timely updates of information, but there are security and privacy concerns about the push model, since it may reveal the current status of a mobile user. For this reason, the pull model may be advantageous. The NAS framework supports both models.

2.2. Piecewiseness in service advertisement and discovery

Service advertisement and discovery is an important component of network awareness in pervasive computing, since the main goal of pervasive computing is enabling network devices to provide services to peers and to be aware of and use the available services. The service advertisement and discovery problem has been extensively researched in technologies such as *Jini* [2] and the *service location protocol* [21]. The NAS framework follows their designs. Service advertisement and discovery are conducted by message distribution through a specified multicast channel in the wired network and infrastructure-based (non-MANET) wireless networks. In infrastructure-based wireless networks there is only one hop between the base station and a mobile device. The broadcast nature of wireless communication automatically enables multicasting in such wireless networks.

The advent of wireless communication technology has facilitated the development of mobile ad hoc networks, including sensor networks [13]. In this section, we consider MANET in the wireless/mobile domain of figure 1. MANET is a dynamically reconfigurable wireless network with no fixed infrastructure due to the nomadism and mobility of mobile devices. In addition, communication routes between mobile devices are often multihop because of limited radio propagation range. Thus, a separate protocol for service advertisement and discovery in MANET would be highly appropriate.

In [9], based on the on-demand multicast routing protocol (ODMRP) [14], we presented a protocol for service advertisement and discovery in MANETs. Compared to other multicast protocols for MANET, ODMRP is very effective and efficient in most simulation scenarios presented in [15]. Basically, service advertisement and discovery information is piggybacked in the ODMRP control packets (*Join Query* and *Join Reply*). Therefore, implementation workload and resource consumption is lightweight once mobile devices support multicast routing.

Since the protocol and the packet format of the ODMRP is different from that of the multicast scheme in the fixed wired domain, we must translate service advertisements between the domains. This is performed by the articulating point of the heterogeneous data networks, i.e., the proxy/gateway, which supports the query/reply mechanism. Thus, it bridges piecewiseness in service advertisement and discovery.

Note that the articulating point may not have overall information of all existing services in MANET. The *Time-to-Live* field in the service advertisement can control the geographical coverage of a service, so advertisement may not reach the articulating point. Once a device needs a service, it sends a query to a well-known multicast address for service discovery, which corresponds to the service query/reply multicast group, asking about the existence of the service. The multicast group consists of servers and devices which support the service query/reply mechanism, such as the proxy/gateway.

2.3. Piecewiseness in network awareness techniques

In traditional network awareness frameworks mentioned earlier, awareness tasks are performed in a *unitary* fashion. The awareness task is performed such that the entire communication path is considered as a single entity. Consider, for example, the pervasive multimedia computing paradigm, where a mobile device needs to send its on-site video stream to a remote host located in an enterprise office. The mobile device first acquires the average available bandwidth of its inter-domain communication path to configure the resolution and frame-rate of the video stream. In unitary frameworks, the mobile device may send a number of back-to-back packet-pairs to the remote host, the remote host would bounce back the packet-pairs, and the mobile host would receive the packet-pairs and use the inter-packet time and packet-size to compute the available bandwidth.

Unlike a unitary framework, techniques in the piecewise NAS framework acquire end-to-end network awareness information in a piecewise way. The network awareness is considered in two parts: the awareness of the wireless/mobile domain and the awareness of the fixed wired domain. The NAS manager or NAS agent integrates the intermediate results about these two domains and provides the final result to the application. Here we present three examples of piecewise techniques for end-to-end NAS.

2.3.1. Available bandwidth

The end-to-end available bandwidth is a frequently used parameter for adaptive applications. Generally, the end-to-end available bandwidth is the bottleneck bandwidth along the communication path. The essential idea of one method for available bandwidth awareness [7] uses inter-packet time to estimate the characteristics of the bottleneck link. If two packets (e.g., ICMP probe packets) travel together so that they are queued as a pair at the bottleneck link with no packet intervening between them, then their inter-packet spacing is proportional to the processing time required for the bottleneck link to transmit the second packet of the pair (figure 4).

We have implemented the inter-packet time method in the NAS framework. In our implementation, the probing packets can be payload packets as well as explicit ICMP probe packets. When an application queries the NAS agent about the available bandwidth, the agent delegates the NAS man-

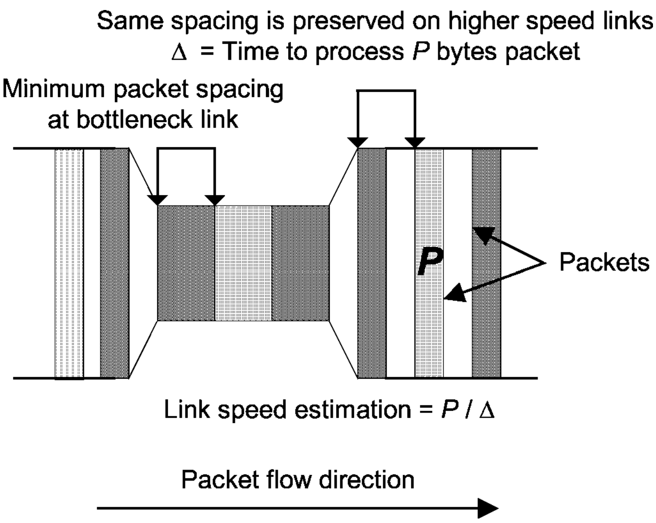


Figure 4. Available bandwidth awareness.

ager to perform this awareness task by sending a query with the address of the destination device. The manager divides the task into three steps. First, it activates the NAS daemon at the proxy/gateway responsible for the awareness service of available bandwidth to measure the available bandwidth between the mobile device and the proxy, say, b_1. Next, it gets the available bandwidth between the proxy and the remote Internet device from the same awareness service by the NAS daemon, say, b_2. Finally the manager computes the end-to-end available bandwidth b as

$$b = \min(b_1, b_2) \tag{1}$$

and sends b as the result back to the NAS agent. Note that the first and the second step can be done in parallel. Finally, the NAS agent replies with the result to the application.

2.3.2. Round-trip time

Similar to the awareness of the available bandwidth, the NAS agent delegates this task to the NAS manager. The RTTs are measured between the mobile device and the proxy/gateway (rtt_1) and between the proxy/gateway and the remote Internet device (rtt_2). The manager computes the end-to-end round-trip time rtt as

$$rtt = rtt_1 + rtt_2 \tag{2}$$

and sends rtt back to the NAS agent, which forwards it to the application.

2.3.3. Packet loss type

In the wired Internet, all losses are assumed to be congestion-type losses (e.g., by TCP). In the pervasive computing paradigm, however, losses on wireless links occur frequently due to signal corruption or for reasons other than congestion, and such losses are called transmission losses. Congestion loss is a relatively sustained phenomenon compared to transmission loss. It is beneficial and sometimes essential for network applications to be aware of the different loss types, i.e., to differentiate between transmission losses and congestion losses. For example, since TCP assumes that all losses are of congestion-type, its performance degrades when temporary transmission losses are mistakenly considered as congestion losses, which unduly triggers TCP congestion control algorithms such as slow start or window decrease [19].

The piecewise NAS framework can be used to enable network applications at mobile devices to differentiate such losses. Packets of communication sessions are snooped by an NAS daemon at the proxy/gateway. Once a network application at the mobile device observes a packet loss, it may query the NAS agent about whether the NAS also observed the same packet loss. If affirmative, then this packet loss is due to congestion loss, otherwise it is due to transmission loss. Thus the application is aware of the type of the packet loss and can react correctly.

3. Case studies

3.1. Multicasting in sensor monitoring

3.1.1. Scenario overview

With the development of small, low-power devices that combine general purpose computing with sensing and wireless communication capability, sensors are becoming key components in pervasive computing, especially in the home networking and MANET scenarios. Sensor monitoring refers to receiving data reported by a sensor, such as a temperature surveillance sensor.

For example, in a home network, a wireless surveillance sensor measures the temperature of a certain space in the home. It periodically reports the temperature value to several monitoring stations, such as the central temperature control station in the home network, a monitoring station at a nearby community fire department, a monitoring program running at the home-owner's office computer, and so on. Because of the intrinsic characteristics of one-to-many communications in this scenario, multicast is commonly used to provide efficient network resource utilization by sending packets from one to N group members rather than sending N unicast packets. The way the multicast group is formed can affect the performance of the sensor-monitoring task. Here we illustrate the performance enhancement in terms of wireless bandwidth saving in such scenario by using the piecewise NAS framework.

3.1.2. Performance analysis model

Suppose that there are N fixed devices acting as receivers that need to get the data from a wireless/mobile data source. The analysis presented below can also be carried out, with favorable outcome, for the case where the sender is a fixed wired device and the receivers are wireless/mobile devices. In the sensor monitoring example, temperature reports are sent periodically in unreliable UDP packets via a multicast group. However, the fixed wired monitoring devices require reliable temperature reports from the sensor.

Even though the wireless communication in MANET may be multihop, the data path from a mobile device to the

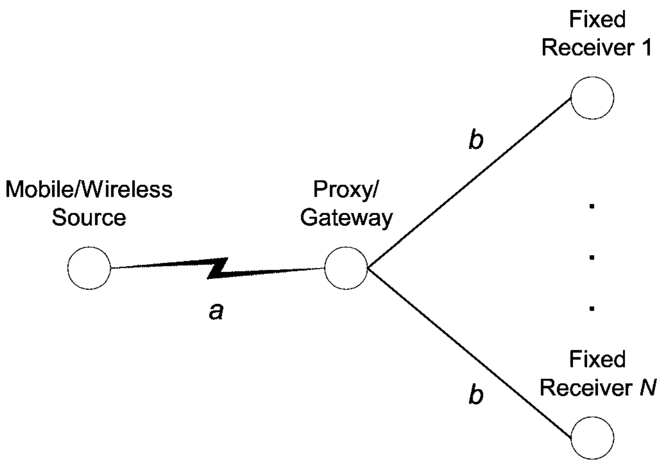


Figure 5. Performance analysis model for sensor monitoring.

proxy/gateway can still be abstracted as a one-hop link by taking into consideration the packet-loss characteristics. The performance analysis model is shown in figure 5. Similar to figure 1, there is a wireless/mobile domain and a fixed wired domain. The loss rate of the wireless/mobile domain is a, which is the probability of a packet being incorrectly transmitted/received between the mobile/wireless device and the proxy/gateway. We assume that all wired links in the fixed wired domain have the same packet loss rate, b, which is the probability of a packet being incorrectly transmitted/received between the proxy/gateway and the fixed devices, e.g., monitoring stations. We assume further that the packet loss rates of these links are independent.

In practice, a and b change dynamically. However, in this performance analysis we assume that the heterogeneous data network is in a steady state so that a and b remain the same over a certain period of time, say, 10 RTTs. Note that the above assumptions are about the link state and the following analysis is independent of the source traffic pattern. Now, assume that each report by the temperature surveillance sensor fits in one data packet.

3.1.3. Wireless bandwidth consumption by unitary framework

In a unitary network awareness framework, the source is a member of the multicast group and sends its packets, e.g., temperature reports, to the multicast group. To increase the probability of all receivers correctly receiving a packet, the packet needs to be sent to the receivers multiple times, considering the packet loss rate a and b. The awareness procedure in this case can be represented as a discrete-time Markov chain, where the state is the number of receivers that have received the report packet correctly.

Given that i receivers have received a report packet correctly at the end of the nth transmission time, the probability that $i + j$ receivers receive the packet correctly at the end of the $(n + 1)$st transmission time is given by

$$p(i, i+j) = \binom{N-i}{j}(1-r)^j r^{(N-i-j)}, \qquad (3)$$

where r is the packet loss rate between the sender and the receiver, which is

$$r = 1 - (1-a)(1-b). \qquad (4)$$

Let $P(k, n)$ denote the probability that k receivers have received the packet by the end of the nth transmission time. Thus,

$$P(k, n) = \sum_{i=0}^{k} p(i, k) P(i, n-1), \qquad (5)$$

where

$$P(i, 0) = \begin{cases} 1, & i = 0, \\ 0, & i > 0. \end{cases} \qquad (6)$$

The probability that all N receivers have received the report correctly by the end of nth transmission is $P(N, n)$. Since the dominant IP packet loss rate in the US part of the Internet is around 5–10% [3,4], we assume that $a = b = 0.1$, thus $r = 0.19$. The probability of $P(N, n)$ is illustrated in figure 6.

The minimum number of transmissions, M_t, to achieve a certain probability, β, that all N receivers have received the packet correctly can be computed by $P(N, n) > \beta$. Since the consumption of wireless bandwidth is proportional to the number of transmissions, we use M_t as the performance index. For example, if $\beta = 99.99\%$, then the relationship between M_t and the total number of receivers, N, is shown in figure 7.

3.1.4. Wireless bandwidth consumption by piecewise framework

In the piecewise framework, the wireless/mobile device is not a member of the multicast group. Instead, it sends the packet, e.g., a temperature report, to the proxy/gateway that is a member of the multicast group, and the proxy/gateway relays the report to the multicast group. The proxy/gateway can repetitively relay the temperature report as many times as necessary to achieve success probability of approximately 1 in the wired network without consuming the wireless bandwidth. After the first packet transmission, the probability, s_p, of all receivers receiving the report correctly is

$$s_p = 1 - a. \qquad (7)$$

After the nth transmission, the probability, P_p, of all receivers receiving the packet correctly is

$$P_p = 1 - (1 - s_p)^n. \qquad (8)$$

The relationship between P_p and n, with $s_p = 0.9$, is shown in figure 8. In order to achieve the probability of successful awareness approximately equal to 1, n should be large. To achieve the same probability of success β, $\beta < 1$, as with the unitary framework, at least M_p packets should be sent, where

$$M_p = \mathrm{ceil}\big(\log_{(1-s_p)}(1-\beta)\big), \qquad (9)$$

because $\beta < 1 - (1 - s_p)^n$, if $s_p < 1$, where ceil$(\cdot)$ is the ceiling operator.

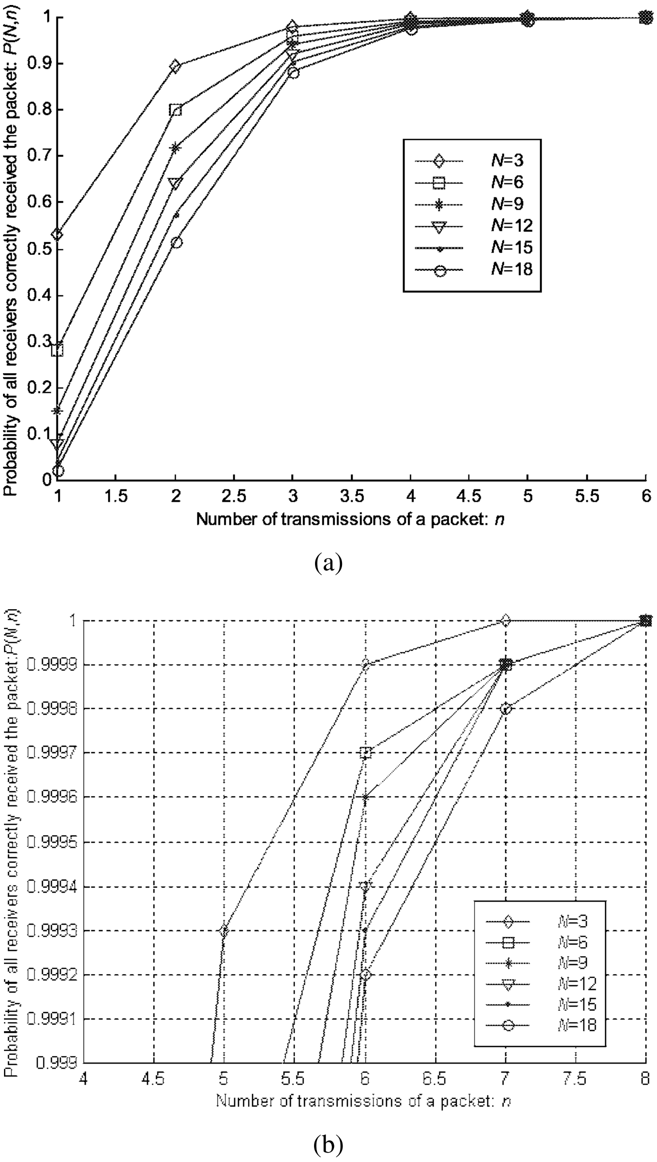


Figure 6. Probability $P(N, n)$. (a) Zoom-out view. (b) Zoom-in view.

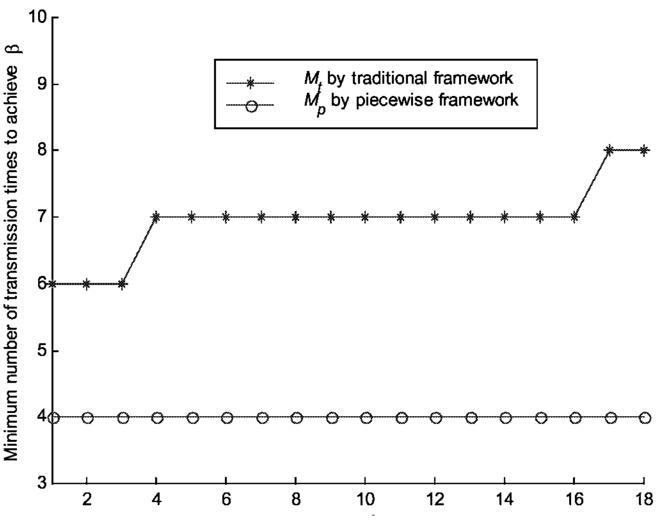


Figure 7. Relationship between N, M_t, and M_p with $\beta = 99.99\%$.

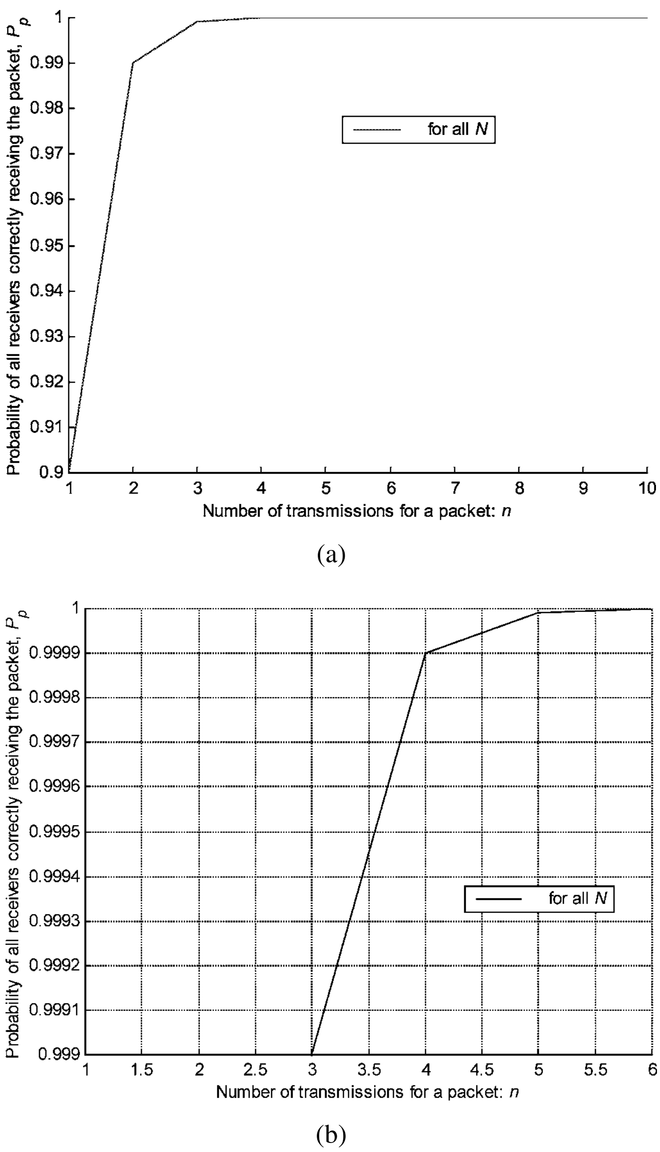


Figure 8. Probability P_p. (a) Zoom-out view. (b) Zoom-in view.

We use M_p as the performance index for the consumption of wireless bandwidth. For $\beta = 99.99\%$, the relationship between M_p and the total number of receivers, N, is shown in figure 7. The graphs show that $M_t > M_p$ for the same β. Therefore the performance enhancement in terms of wireless bandwidth consumption is $(M_t - M_p)/M_p$, which is illustrated in figure 9. The performance gain of piecewise framework over unitary framework increases as the number of receivers, N, increases.

3.2. Bandwidth-aware data streaming in pervasive multimedia computing

3.2.1. Scenario overview

An important application in wireless/mobile pervasive computing is telecollaboration. For example, an on-site worker collaborates with an office-bound expert using multimedia applications. Multimedia applications, such as audio and video ones, are helpful to enhance the collaboration experience. It is

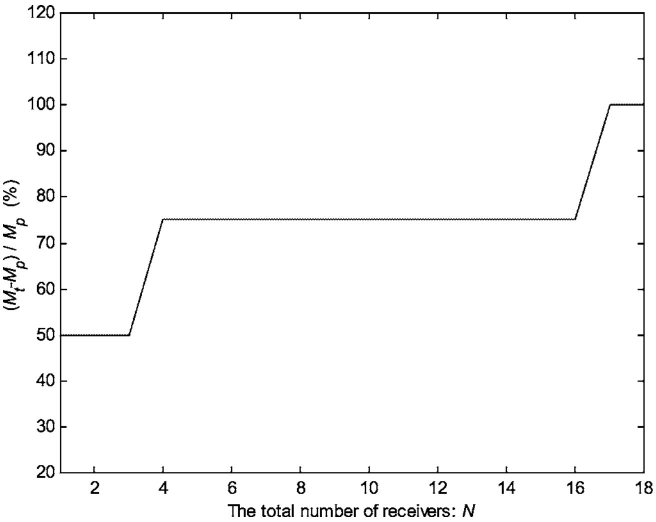


Figure 9. Performance enhancement with $\beta = 99.99\%$.

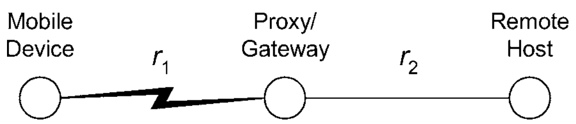


Figure 10. Performance analysis model for bandwidth awareness.

important to improve the performance of the multimedia data streaming in this scenario. One approach is to use bandwidth-aware data streaming. For instance, a multimedia application learns about the average available bandwidth of its inter-domain communication path to configure its codec, e.g., the compression ratio and frame-rate. Another common example of data streaming in pervasive computing is mobile video-on-demand (VoD). It is again helpful for the VoD application to be aware of the available bandwidth.

In this case study, the awareness of the available bandwidth is used for performance analysis of the piecewise framework. More specifically, the performance in terms of wireless bandwidth consumption will be compared between the piecewise and the unitary frameworks as they provide bandwidth awareness service to multimedia applications.

3.2.2. *Performance analysis model*

Based on figure 1, the performance analysis model is abstracted as in figure 10. The end-to-end path of the multimedia data stream is between the mobile host and the remote host, i.e., an inter-domain communication path. Either of them could be the sender or the receiver. Assume that the path between the sender and the receiver is symmetric, meaning that the communication path characteristics are the same in both directions. In practice this is not true, though often the difference is small.

We denote the loss rate of the wireless/mobile domain as r_1, which is the probability of a packet being incorrectly transmitted/received between the mobile device and the proxy/gateway. The loss rate of the fixed wired domain is r_2, which is the probability of a packet being incorrectly transmitted/received between the proxy/gateway and the remote host. As in the first case study, assume that r_1 and r_2 are constant and they are packet-independent. The probability of a packet being successfully transmitted/received between the mobile device and the remote host is

$$p = (1 - r_1)(1 - r_2). \tag{10}$$

According to the available bandwidth measurement algorithm (section 2.3.1), only when both packets in a back-to-back packet-pair travel a round trip successfully can a correct value of the available bandwidth be computed. For simplicity, assume that the query and the reply packets between the NAS agent and NAS manager have the same size as that of the probe packet.

3.2.3. *Wireless bandwidth consumption by unitary framework*

In a unitary framework, the available bandwidth is determined on the mobile device. The mobile device sends out the back-to-back probe packet-pair, receives the bounced packet pair from the remote host, and computes the available bandwidth. Based on equation (10), the probability of obtaining a correct measurement, which is denoted as α, if only one packet-pair is used can be expressed as

$$\alpha = p^4. \tag{11}$$

If multiple, say m, packet-pairs are used, then the probability of obtaining a correct measurement increases, which is expressed as

$$1 - \left(1 - p^4\right)^m. \tag{12}$$

Therefore, to achieve the probability approximately equal to 1, m should be large. To achieve the probability β of successful service, at least M_t packet pairs should be sent, where

$$M_\mathrm{t} = \mathrm{ceil}\left(\log_{(1-p^4)}(1-\beta)\right), \tag{13}$$

because $\beta < 1 - (1 - p^4)^m$, if $p < 1$. Since the consumption of wireless bandwidth is proportional to the number of packet-pairs, we use M_t as the index.

3.2.4. *Wireless bandwidth consumption by the piecewise framework*

In the piecewise framework, the available bandwidth measurement in the fixed wired part does not consume the wireless bandwidth. Only the measurement in the mobile wireless part contributes to wireless bandwidth consumption. Based on the previous analysis, to achieve the same success probability β as for the unitary framework, at least M_p packet pairs should be used:

$$M_\mathrm{p} = \mathrm{ceil}\left(\log_{(1-q^4)}(1-\beta)\right) + 0.5, \tag{14}$$

where $q = 1 - r_1$. The 0.5 packet-pair comes from a query packet plus a reply packet between the NAS manager and the NAS agent.

Since $p < q$, it follows that $M_\mathrm{t} > M_\mathrm{p}$. Therefore the performance enhancement in terms of wireless bandwidth consumption is $(M_\mathrm{t} - M_\mathrm{p})/M_\mathrm{p}$. This can be substantial in the

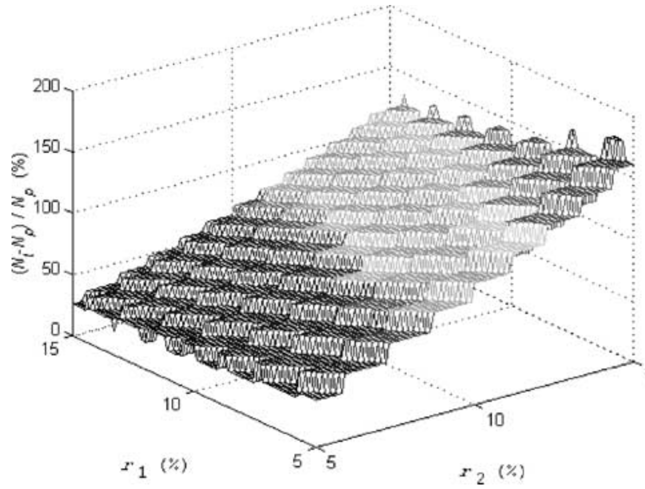


Figure 11. Performance enhancement with $\beta = 99.99\%$.

presence of unreliable communication as is the case in the Internet. As mentioned before, the dominant packet loss rate in the US part of the Internet is around 5–10% [3,4]. The performance enhancement for realistic loss rates is shown in figure 11. An additional benefit of the performance enhancement is the battery energy saving, since the power amplifier of a radio device draws a much greater amount of power during the transmission [10].

3.2.5. Analysis of awareness agility

The piecewise NAS framework involves communication between the NAS agent and the NAS manager, between the NAS agent and the application, and between the NAS manager and the NAS daemon. Therefore it is necessary to examine whether the piecewise framework impairs the agility of network awareness, i.e., the timely measurement and, ultimately, the application's response.

Assume that the RTT between the proxy/gateway and the remote host is t_1, and the RTT between the mobile host and the proxy/gateway is t_2. Also, denote the communication time between the NAS agent and application as t_3 and between the NAS manager and the NAS daemon as t_4. We ignore the computing time. Finally, suppose that the interval between the packet-pairs is so small that the duration from sending out the first packet-pair to the last packet-pair in a measurement can be neglected.

In the unitary framework, the time t_t to achieve successful bandwidth awareness is $(t_1 + t_2)$.While in the piecewise framework, the timet_p to achieve successful bandwidth awareness is expressed as

$$t_p = \max(t_1, t_2) + t_3 + t_4 + t_2, \tag{15}$$

because the measurement between the mobile device and the proxy/gateway and the measurement between the proxy/gateway and the remote host are conducted in parallel. The times t_3 and t_4 can be neglected, since the RTT between different hosts is several orders of magnitude greater than the interprocess communication time between different applications on the same host.

Therefore, assuming that the transmission times for M_t or M_p packet-pairs are negligible, the time difference between the piecewise framework and the unitary framework is

$$\max(t_1, t_2) - t_1, \tag{16}$$

which is zero for $t_1 \geqslant t_2$, or $t_2 - t_1$ for $t_1 < t_2$. In the latter case the piecewise framework impairs the application's agility. However, since the RTTs are almost always in the range of milliseconds, the impact of the information exchange between NAS entities on awareness agility is negligible.

4. Platform scalability of NAS framework

4.1. Scaling up to a multi-tasking OS

As higher-performance and energy-efficient processors and memory chips emerge, more and more mobile devices will support multitasking operating systems. Thus multiple network applications can be active simultaneously at a mobile device. Suppose that the wireless link is the bandwidth bottleneck. In a unitary framework N applications are adaptive to the available bandwidth and perform the measurements independently. To achieve the probability β of successful awareness, the total wireless bandwidth consumption will be $N \times M_t$, if there is no correlation among their individual measurement successes.

In contrast, to achieve the same success probability β in the piecewise NAS framework, the total wireless bandwidth consumed will be only M_p. An application sends the query to the NAS agent that delegates the task to the NAS manager. The manager obtains the value of the available bandwidth, b, from the NAS daemon at the proxy/gateway, and feeds it back to the agent. Finally, the agent gets the traffic-multiplexing ratio, γ, of that application from the traffic scheduler, and replies with the final result, $\gamma \times b$, to the application. Therefore, the other applications can share the measurement result b multiplied by their corresponding traffic-multiplexing ratios.

4.2. Scaling down to a lightweight OS

The minimum footprint of the NAS framework at the mobile device is the NAS agent, which is the service portal communicating with the NAS manager. The NAS daemons can all run at the NAS manager's host. The agent is a simple proxy for invoking NAS manager's services, thus making the framework downward scalable.

5. Conclusions

This paper presents a piecewise framework for network awareness service (NAS) for mobile pervasive computing, which is based on the intrinsic piecewiseness of heterogeneous data networks. Piecewiseness in framework architecture, service advertisement and discovery, and network-awareness techniques in the NAS framework are investigated. Scalability of the framework is also discussed with regard to platform capabilities. The framework scales up to powerful computing devices with multi-tasking OS and scales down to small devices with lightweight OS.

Case studies of applying the NAS framework to sensor monitoring in home networks and data streaming in pervasive multimedia computing are presented. The analytical results about the performance of the piecewise NAS framework in these case studies show that it can gain significant advantages over unitary network-awareness frameworks in terms of reducing wireless bandwidth consumption and saving limited battery energy of mobile devices. Our continuing work includes integrating the schemes for prediction of network parameters.

Acknowledgements

This research is supported by NSF KDI Contract No. IIS-98-72995, US Army CECOM Contract No. DAAB07-00-D-G505 and by the Rutgers Center for Advanced Information Processing (CAIP). The authors would like to thank Dr. Edward J. Devinney and anonymous reviewers for their valuable comments.

References

[1] IrDA Specifications, http://www.irda.org/standards/specifications.asp

[2] Jini Network Technology, http://www.sun.com/jini/

[3] MPEG-4 Overview, ISO/IEC JTC1/SC29/WG11 N3536 (July 2000) http://www.iis.fhg.de/amm/techinf/mpeg4/mp4_overv.pdf

[4] Tutorial on Internet Monitoring & PingER at SLAC, http://www.slac.stanford.edu/comp/net/wan-mon/tutorial.html

[5] G. Banavar, J. Beck, E. Gluzberg, J. Munson, J. Sussman and D. Zukowski, Challenges: An application model for pervasive computing, in: *Proc. 6th Annual International Conference on Mobile Computing and Networking (MobiCom2000)*, Boston, MA (August 2000) pp. 266–274.

[6] J. Bolliger and T. Gross, A framework-based approach to the development of network-aware applications, IEEE Transactions on Software Engineering 24(5) (May 1998) 376–390.

[7] R.L. Carter and M.E. Crovella, Measuring bottleneck link speed in packet-switched networks, Technical Report BU-CS-96-006, Computer Science Department, Boston University (March 1996).

[8] J. Case, M. Fedor, M. Schoffstall and J. Davin, A simple network management protocol, RFC 1067 (August 1988) http://www.faqs.org/rfcs/rfc1067.html

[9] L. Cheng and I. Marsic, Service discovery and invocation for mobile ad hoc networked appliances, in: *Proc. 2nd International Workshop on Networked Appliances (IWNA'2000)*, New Brunswick, NJ (December 2000).

[10] C.F. Chiasserini and R.R. Rao, Pulse battery discharge in: communication devices, in: *Proc. 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'99)*, Seattle, WA (August 1999) pp. 88–95.

[11] A. Dutta-Roy, Networks for homes, IEEE Spectrum 36(12) (December 1999) 26–33.

[12] R. Grimm, T. Anderson, B. Bershad and D. Wetherall, A system architecture for pervasive computing, in: *Proc. 9th ACM SIGOPS European Workshop*, Kolding, Denmark (September 2000) pp. 177–182.

[13] J. Kulik, W. Rabiner and H. Balakrishnan, Adaptive protocols for information dissemination in wireless sensor networks, in: *Proc. 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'99)*, Seattle, WA (August 1999) pp. 174–185.

[14] S. Lee, M. Gerla and C. Toh, On-demand multicast routing protocol (ODMRP) for ad hoc networks, Internet draft, work in progress (June 1999) http://www.ietf.org/proceedings/99jul/I-D/draft-ietf-manet-odmrp-01.txt

[15] S. Lee, W. Su, J. Hsu, M. Gerla and R. Bagrodia, A performance comparison study of ad hoc wireless multicast protocols, in: *Proc. IEEE Infocom'2000*, Tel-Aviv, Israel (March 2000) pp. 565–574.

[16] B. Noble, System support for mobile, adaptive applications, IEEE Personal Communications 7(1) (February 2000) 44–49.

[17] B.D. Noble and M. Satyanarayanan, Experience with adaptive mobile applications in Odyssey, Mobile Networks and Applications 4(4) (July/August 1999) 245–254.

[18] C.E. Perkins, *Ad Hoc Networking* (Addison-Wesley, 2001).

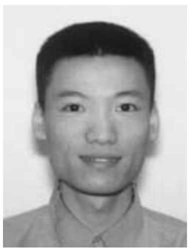
[19] P. Sinha, N. Venkitaraman, R. Sivakumar and V. Bharghavan, WTCP: A reliable transport protocol for wireless wide-area networks, in: *Proc. 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'99)*, Seattle, WA (August 1999) pp. 231–241.

[20] D. Tang, C. Chang, K. Tanaka and M. Baker, Resource discovery in ad hoc networks, Technical report CSL-TR-98-769, Stanford University (August 1998).

[21] J. Veizades, E. Guttman, C. Perkins and S. Kaplan, Service location protocol, RFC-2165 (June 1997) http://www.ietf.org/rfc/rfc2165.txt

[22] G. Welling and B.R. Badrinath, An architecture for exporting environment awareness to mobile computing applications, IEEE Transactions on Software Engineering 24(5) (May 1998) 391–400.

[23] R. Wolski, N.T. Spring and J. Hayes, The network weather service: A distributed resource performance forecasting service for metacomputing, Future Generation Computer Systems 15(5–6) (October 1999) 757–768. Also UCSD Technical report TR-CS98-599 (September 1998).



Liang Cheng is a Ph.D. candidate and graduate fellow of electrical and computer engineering at Rutgers University. He received the B.S. degree from Huazhong University of Science and Technology, Wuhan, P.R. China, in 1994, and the M.S. degree from Tsinghua University, Beijing, P.R.China, in 1997. His current research interests include wireless data communications, multimedia networks, distributed computing, network performance and control, and intelligent agents for information processing.
E-mail: chengl@caip.rutgers.edu

Ivan Marsic is an Assistant Professor of Electrical and Computer Engineering at Rutgers University. He received the B.S. and M.S. degrees in computer engineering from University of Zagreb, Croatia, in 1982 and 1987, respectively, and Ph.D. degree in biomedical engineering from Rutgers University, New Brunswick, NJ, in 1994. Prof. Marsic is the chief architect of the DISCIPLE system, an advanced groupware system that enables teams, consisting of individuals with specific roles, to collaboratively access, manipulate, analyze, and evaluate multimedia data using networked information systems having disparate computing and communication capabilities. He has authored over 70 journal and conference papers and 3 book chapters. His current research interests include mobile computing, groupware, computer networks, and human–computer interfaces. Prof. Marsic is a member of IEEE and ACM and has been a consultant to industry and government.
E-mail: marsic@caip.rutgers.edu