Faculty Publications

2004-03-21

Dynamic Autonomous Transport Selection in Heterogeneous Wireless Environments

Jeffrey M. Brown
jeff.brown@byu.edu

Heidi R. Duffin

Charles D. Knutson
knutson@cs.byu.edu

Shannon B. Barnes
barnes@byu.edu

Ryan W. Woodings
Follow this and additional works at: <https://scholarsarchive.byu.edu/facpub>

Part of the [Computer Sciences Commons](#)

Original Publication Citation

Charles D. Knutson, Ryan W. Woodings, Shannon B. Barnes, Heidi R. Duffin, and Jeffrey M. Brown. "Dynamic Autonomous Transport Selection in Heterogeneous Wireless Environments." Proceedings of the IEEE Wireless Communication and Networking Conference (WCNC 24), Atlanta, Georgia, March 21-25, 24.

BYU ScholarsArchive Citation

Brown, Jeffrey M.; Duffin, Heidi R.; Knutson, Charles D.; Barnes, Shannon B.; and Woodings, Ryan W., "Dynamic Autonomous Transport Selection in Heterogeneous Wireless Environments" (2004). *Faculty Publications*. 449.
<https://scholarsarchive.byu.edu/facpub/449>

This Peer-Reviewed Article is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Faculty Publications by an authorized administrator of BYU ScholarsArchive. For more information, please contact ellen_amatangelo@byu.edu.

Dynamic Autonomous Transport Selection in Heterogeneous Wireless Environments

Charles D. Knutson, Heidi R. Duffin,
and Jeffrey M. Brown
Mobile Computing Lab
Brigham Young University
Provo, UT, USA
{knutson, hduffin, jbrown}@cs.byu.edu

Shannon B. Barnes
Consumer Electronics Group
Intel Corporation
Chandler, Arizona, USA
shannon.b.barnes@intel.com

Ryan W. Woodings
Cypress Semiconductor
Boise, Idaho, USA
rww@cypress.com

Abstract—In this paper, we introduce Quality of Transport (QoT), an architecture for synergistically and autonomously managing session-layer protocol access to multiple transports in heterogeneous wireless environments. We present an overview of the QoT architecture including: 1) transport discovery, 2) service discovery, 3) object exchange, 4) transport switching, and 5) intelligent transport selection. Preliminary successes with our design and implementation of QoT suggest that dynamic intelligent autonomous transport switching can help to optimize user experience and session layer performance in multi-transport environments.

I. INTRODUCTION

It is now common for mobile devices to be equipped with multiple wireless transceivers, each suited to separate usage models and each designed for use with specific transport protocols. New transports and technologies continue to emerge. For example, 802.11 continues to evolve as a wireless networking standard [2], while Ultra-Wideband shows promise for interference-resistant short-distance high-bandwidth data communication [10] [16]. In the Wireless Personal Area Network (WPAN) arena, IrDA offers low-power point-to-point connectivity at speeds up to 16 Mbps [3], while Bluetooth facilitates a broader range of motion at lower throughput [1]. As stated in [9], “no single standard protocol or protocol family will become the universal protocol supported by all networked systems.” Intra-device heterogeneity can be exploited to optimize link quality via intelligent, dynamic transport selection independent of session layer protocols or user applications.

While a device may support multiple transports, and hence multiple (potentially synergistic) communication mechanisms, such potential is rarely exploited. Applications are normally written to a specific session or application layer protocol, which is typically transport-dependent (as are associated service registry and service discovery mechanisms). In situations for which multiple transports are natively available to a single session layer protocol (such as OBject EXchange (OBEX) over Bluetooth or IrDA [5]), user selection is normally required in advance to determine the transport mechanism for a specific connection. As an example, if a user begins an OBEX connection over IrDA in order to take advantage of its high-speed low-power features, but then needs to wander away from the one meter infrared cone, she would have to manually restart the connection and select Bluetooth as the desired transport. If the device were then able to come back into range of the IrDA

connection, the user would again have to cancel and start over in order to switch back.

In this paper, we introduce Quality of Transport (QoT), an architecture for managing session/application layer access to multiple transports in heterogeneous wireless environments. The goal of Quality of Transport (QoT) is to facilitate dynamic, transparent and autonomous transport switching for multi-transport devices in order to provide the highest quality data transfer capability within heterogeneous wireless environments. QoT does this by introducing an intelligent protocol layer between the session/application layers (such as HTTP, FTP and OBEX) and transport layers (such as IrDA, Bluetooth, and TCP/IP).

The remainder of this paper is structured as follows. Section II presents related work in multi-transport heterogeneous wireless data communications. Section III provides an overview of the QoT architecture. Sections IV and V discuss transport and service discovery in multi-transport environments. Sections VI and VII describe QoT mechanisms for object exchange and transport switching. Section VIII provides an overview of intelligent autonomous transport selection in QoT. In section IX we present conclusions.

II. RELATED WORK

A number of research projects have examined heterogeneous connection capabilities, though typically presuming a single usage model (normally Internet or local area network (LAN) access) and a single transport protocol (primarily TCP/IP).

The BARWAN project at the University of California explored the use of vertical handoffs in wireless overlay networks as a mechanism for intelligently and dynamically maintaining an active TCP/IP connection to a network infrastructure [17]. While the BARWAN project enabled transceiver switching, their solution presumed a single transport protocol (TCP/IP) and required that the protocol stack be modified, eliminating the ability to manage multiple transport protocols or transparently enhance existing systems.

The MosquitoNET project at Stanford University also studied continuous Internet connectivity to mobile hosts [6]. Their goal was “to switch seamlessly between different network devices to take advantage of whatever connectivity is available.” Their work assumed that the Internet would be used for device-to-device data transfers and therefore concentrated

on maintaining and optimizing Internet connectivity wherever possible.

Other researchers have studied periodic disconnections in mobile computing environments [11], abstracting the physical network medium from the software [14], and multimedia applications that cope with varying network connectivity and bandwidth [15]. These solutions tend to solve the transparent, persistent connection problem for a specific usage model and/or a small set of specific transports.

Research in the Mobile Computing Laboratory of Brigham Young University has explored intra-device heterogeneity in multi-transport environments. In [20] we demonstrated that devices equipped with both IrDA and Bluetooth transceivers and protocols could utilize the infrared link to discover and exchange Bluetooth connection parameters, resulting in a dramatic reduction in the costly Bluetooth discovery and device selection phases. In [12] we explored the use of inverse multiplexing to maximize throughput via simultaneous transmission over multiple shared transport mechanisms. In [7] we presented an algorithm whereby transport and service information may be dynamically obtained in wireless multi-transport environments.

III. QUALITY OF TRANSPORT (QoT) OVERVIEW

In this section we present key system requirements that influenced the design of QoT. We then present a high-level overview of QoT architecture and functionality.

A. Design goals

The following requirements have been integrated into the QoT architecture:

- 1) **No new APIs.** The QoT architecture must not require that applications and protocols be redesigned or rewritten in order to take advantage of its capabilities. QoT must enhance existing systems, rather than strictly future systems.
- 2) **Transparency to session layers, transport layers, and applications.** The QoT system must integrate with and enable existing applications and protocols without their awareness of its operation.
- 3) **Focus on asynchronous object exchange.** The initial QoT architecture is not concerned with synchronous streaming of real time data, but with the asynchronous exchange of data objects.
- 4) **Compatibility with non-QoT devices.** While QoT-enabled devices are able to enhance the quality of connections with other QoT devices, they must also interoperate with non-QoT devices to provide backward compatibility.
- 5) **Local and remote transport discovery.** QoT must be able to determine the transport protocols and transmission mechanisms available on a remote device, even if all such transports are not available at the time of discovery.
- 6) **Local and remote service discovery.** QoT must be able to discover services on remote devices, even when such

Fig. 1. QoT and the OSI Model.

services are tied to transports not currently accessible to the local device.

- 7) **Dynamic transport selection.** QoT must be able to dynamically switch between transport protocols in order to optimize the quality of the link.
- 8) **Intelligent and autonomous decision making.** Transport switching decisions must consider user preferences, link conditions, and device capabilities in making intelligent link decisions with minimal user intervention.

B. High Level Design

Quality of Transport (QoT) is implemented as a protocol layer residing between the transport and session layers of the OSI model (see 1). In order to accommodate the previously stated goals of requiring no new APIs plus providing transparency to existing protocols, QoT introduces upper and lower abstraction modules. The upper module is referred to as the Transport Proxy Module (TPM) and appears to a session layer as if it were an interface to a specific transport. The lower module is referred to as the Transport Abstraction Module (TAM) and interacts with the transport layer as if it were an arbitrary (but indeterminate) session protocol. The TAM can also be viewed as presenting a consistent transport interface to QoT, facilitating an extensible architecture from a transport perspective. By abstracting the upper and lower stack layers in this fashion, QoT is able to insert itself transparently into an existing data communication system.

Figure 2 illustrates a QoT-facilitated data exchange between Device 1 and Device 2 using OBEX as a session protocol. The devices each support multiple transports, two of them common (Bluetooth and IrDA). At the time represented by this figure, the highest quality link is provided by IrDA, so QoT routes the OBEX traffic via the IrDA stack. Should link conditions change such that Bluetooth provides a more desirable link, QoT would switch the underlying transport to Bluetooth without affecting either the session layer transports or the applications that depend upon them.

QoT's role is to optimize link quality by dynamically switching between shared transports in a manner transparent to the session protocols. We should note that QoT does

Fig. 2. Data Exchange Using QoT.

not provide a mechanism for permitting disparate session protocols from communicating with one another. For example, permitting an HTTP session to seamlessly interact with an OBEX or FTP session is a challenging task, especially considering the disparity in feature sets between session protocols. Supporting session-layer heterogeneity is beyond the initial scope of QoT.

There is also a disparity in the feature sets of transport protocols. For this reason, QoT defines a minimal set of features upon which it is dependent for the creation of a TAM/TPM pair. These features include: flow control, segmentation and reassembly, sequence preservation, and error correction. Guaranteed sequenced packet delivery capabilities such as those provided by Bluetooth, TCP/IP, and IrDA are sufficient for integration with QoT. Non-guaranteed delivery protocols such as UDP are not sufficient for integration with QoT.

The primary objectives of QoT are: 1) Identifying and tracking available transports on remote devices; 2) Identifying available session protocols on remote devices; 3) Exchanging data objects; 4) Managing transport switching; 5) Optimizing link quality via intelligent decision making. The remainder of the paper addresses each of these objectives in turn.

IV. TRANSPORT DISCOVERY

Multi-Transport Discovery is the mechanism that allows devices to discover common transports [7]. Figure 3 describes the mechanism for Multi-Transport Discovery. This algorithm is divided into four phases: Transport Probing, Transport Querying, Transport-to-Device Mapping, and Transport Accessibility. The Transport Probing and Transport Querying phases identify existing transports. The Transport-to-Device Mapping phase works in conjunction with Transport Querying to properly match transports with remote devices. The last phase, Transport Accessibility, is an ongoing process that continually checks discovered transports to ascertain their availability. Each of these phases utilizes a Remote Device Table (RDT) to store and track information regarding each device and its respective transports and services. The functions described in this section are facilitated within the QoT architecture via Transport Packets, which facilitate the exchange of transport information.

Fig. 3. Flow graph of a Multi-Transport Discovery Algorithm.

A. Transport probing and querying

The main objective of Transport Probing is to find at least one transport through which communication can be established with a remote device. After an initial connection is made, information regarding additional transports can then be exchanged over this link.

Transport Querying complements Transport Probing by discovering supported transports that were not visible during the probing phase. Transports that require line-of-sight (such as IrDA) or those that have limited range (such as Ultra-Wideband and Bluetooth) might remain undiscovered after Transport Probing. However, if at least one link can be established, querying for transports guarantees that a local device can be made aware of all transports supported by accessible remote devices. We designed the Transport Exchange Protocol (TEP) to facilitate Transport Querying within QoT¹.

The Transport Exchange Protocol provides two function calls, `Transport_Query` and `Transport_Info_Query`. The first function, `Transport_Query`, issues a QoT Transport Query Packet to the recipient over a link discovered during Transport Probing. The receiving device returns a Transport Query Response Packet, which contains a list of available transport types supported by the remote device. The local device can then obtain additional information about specific transports by calling `Transport_Info_Query` for a particular transport. Remote devices may then respond by sending a QoT Transport Info Response Packet, containing the parameters necessary to connect via the particular transport, and a structure that

¹It should be noted that certain transports support native service discovery protocols that can be used to store and exchange transport information in an ad hoc point-to-point fashion. However, not all transports provide standard ad hoc point-to-point service discovery mechanisms, hence the creation of an independent Transport Exchange Protocol was deemed necessary within the QoT architecture.

Let a be a transport address
Let t be a transport type
Let T be a list of transport types
Let d be a remote device containing $\{a_0, \dots, a_{n-1}\}$
where n is the number of transport addresses on a device
Let d_i be a local device containing $\{t_0, \dots, t_{m-1}\}$
where m is the number of supported transport types
Let D be a Remote Device Table containing $\{d_0, \dots, d_{r-1}\}$
where r is the number of remote devices
Let s represent a link's availability status

Algorithm Multi-Transport Discovery

Phase 1: Transport Probing

```

(1) for each  $t_i$  in  $d_i$ 
(2)    $a \leftarrow \text{probe}(t_i)$ 
(3)   if  $a$  not in  $D$ 
(4)      $D \leftarrow \text{insertNewDevice}(a)$ 

```

Phase 2 & 3: Transport Querying and Device Mapping

```

(5) for each  $d_i$  in  $D$ 
(6)    $T \leftarrow \text{Transport\_Query}(d_i, a_0)$ 
(7)   for each  $t_j$  in  $T$  where  $t_j$  is in  $d_i$ 
(8)      $a \leftarrow \text{Transport\_Info\_Query}(d_i, a_0, t_j)$ 
(9)      $D \leftarrow \text{updateTable}(d_i, a)$ 
(10)  for each  $d_k$  in  $D$  where  $k > i$ 
(11)    for each  $t_m$  in  $d_i$ 
(12)      if  $t_m$  equals  $d_{k,t_0}$ 
(13)         $D \leftarrow \text{deleteDevice}(d_k)$ 

```

Phase 4: Transport Accessibility

```

(14) repeat
(15)   for each  $d_i$  in  $D$ 
(16)     for each  $a_j$  in  $d_i$  such that  $a_j$  is not the current link
(17)        $s \leftarrow \text{TestAddress}(a_j)$ 
(18)        $D \leftarrow \text{updateTable}(a_j, s)$ 

```

Fig. 4. Multi-Transport Discovery Algorithm.

contains name/value pairs providing various link descriptors.

B. Address-to-device mapping and transport accessibility

During Transport Probing, newly discovered transports are initially assumed to refer to a unique device. The Address-to-Device Mapping phase associates discovered transports with specific devices in the RDT, resolving ambiguity and redundancy that may result from Transport Probing.

Once Address-to-Device Mapping has been completed, link connection and data exchange may commence between devices. The Transport Accessibility phase operates as a background process, periodically ascertaining link availability for various transports during the application session. The information provided by this phase enables QoT to determine the feasibility or desirability of dynamically switching transports during a session. The interval between link checks is dependent on the nature of the transport being checked and the nature of the link used in the active session. Optimal frequency of inquiry is a topic of on-going research.

C. Multi-transport discovery algorithm

Figure 4 presents the algorithm for Multi-Transport Discovery. During the probing phase, each of the local device's transports is subjected to the probe function (2), which performs the transport dependent method for discovery of remote devices. At this stage each remote device in the RTD is associated with only one transport. Each of these transports is subjected to a Transport_Query (6), which returns a list of transport types found on the remote device. Transport types supported

Fig. 5. Screen shot of a prototype Linux-based PDA application after performing Multi-Transport Discovery.

by both local and remote devices are then subjected to a Transport_Info_Query (8) which returns information needed to perform link establishment as well as attributes that enable the system to make intelligent decisions for dynamic transport switching. The RDT entry for the remote device is updated with new transport information (9). Each of the returned transports is then checked against the remaining probed transports for duplicates (12). By deleting any duplicate probed transports, we eliminate unnecessary querying calls and properly associate groups of transports to a particular device.

The TestAddress function (17) takes into account the type of transport being tested in determining the proper delay to be incorporated. Once the status is returned, the RDT is updated.

Figure 5 shows a screen shot of an implementation of the Multi-Transport Discovery Algorithm on a Linux-based PDA. This device includes a USB port (LAN), a Bluetooth Compact Flash (CF) card, and an integrated IrDA transceiver.

V. SERVICE DISCOVERY

Service discovery protocols play a vital role in dynamic wireless systems. These mechanisms facilitate service connection in the absence of a priori information, permitting devices to negotiate relevant connection parameters with minimal user intervention [13]. WPAN protocol stacks in particular tend to have integrated service discovery protocols. For example, the Information Access Service (IAS) is part of IrDA's Link Management Protocol (IrLMP) [4] and Bluetooth's Service Discovery Protocol (SDP) is tightly coupled with the Logical Link Control and Adaptation Layer Protocol (L2CAP) [1]. The integration of service discovery protocols within transport layers presents a challenge in abstracting underlying transports from network applications.

A. Tunnelling service discovery protocols

We apply protocol tunnelling as a method to manage multiple service discovery mechanisms in a transport switch-

Fig. 6. Tunnelling an IrDA service discovery call through Bluetooth SDP.

ing environment. In this case, we are encapsulating a non-compatible service discovery protocol within the payload of a transport protocol.

Figure 6 illustrates tunnelling in an IrDA IAS call over Bluetooth. The figure shows two QoT devices that support multiple transports (Bluetooth, IrDA, and TCP/IP over 802.11b). In this figure, device A exchanges an electronic business card with device B. QoT initiates the connection over Bluetooth because the users are at different ends of a room, beyond IrDA's maximum range of one meter. However, the personal information manager (PIM) software was designed to exchange information using OBEX over IrDA and needs to initiate an IAS query to ascertain the proper LsapSel value.

At (1) the OBEX client performs a GetValueByClass IAS call. At (2), the IrDA TAM responds to the API call and adds the search parameters into a QoT packet. The QoT packet-header identifies the packet as an IrDA Information Access Protocol (IAP) frame and returns the packet to the QoT layer. At (3), QoT forwards the packet to the preferred transport (in this case Bluetooth). Once the peer device receives the packet (4), its QoT layer examines the QoT packet-header, determines that it is an IrDA IAS query, and forwards the packet to the IrDA TAM of Device B. The TAM is familiar with the intricacies of IAS and is able to parse the IAP frame and perform a localized search based upon the search parameters (5).

The IAS GetValueByClass response is returned to Device A in a similar approach. The IrDA TAM inserts the search results into a QoT packet and sends it back to the inquiring device. The local device's QoT layer identifies the packet as an IAS call and forwards the packet to its IrDA TAM. The TAM extracts the return value and advances the value to the OBEX session layer.

A similar encapsulation scheme can be used for Bluetooth SDP and TCP/IP service discovery protocols such as SLP (Service Location Protocol). Because TCP/IP service discovery protocols reside in the session layer, at (4) QoT would locally query the SLP repository that resides above the QoT layer.

VI. OBJECT EXCHANGE

As stated earlier, the preliminary design of QoT is focused on facilitating asynchronous data object exchange. The issues involved in synchronous data exchange (such as streaming

multimedia) are important, but will be dealt with in future versions of QoT. To facilitate object exchange, QoT employs Connection and Data packets.

Connection Packets are used to establish and tear down QoT connections. These packets originate in the TPM and specify the transport intended for use by the session layer, such as an HTTP client requesting a connection over the TCP (802.11b) transport. Once the connection is established, the session layer assumes the data is being routed over the requested transport, although QoT may in fact be rerouting data over a different transport. Packet types that are connection-dependent, such as Data and Switch packets (explained later), can only be sent after a QoT connection is established and before the connection is dropped. Connection-independent packet types, such as Transport Discovery packets, can be sent at any time.

Data packets are used to transfer session data between devices. In order to recover from dropped transport connections, QoT must be capable of rolling back to the last known point of successfully transmitted data and continue the data transmission. This requires the devices to establish synchronization points during the data transmission. The purpose of sync points is to acknowledge successful reception of data by the receiving device so that the sending device can free up the buffer space that is storing the data in case a retransmission is necessary. Synchronization points are based on Data Packet IDs, which are sequentially assigned to packets as they are transmitted.

VII. TRANSPORT SWITCHING

The goal of QoT is to maintain a connection between two devices over the best transport. Sometimes the change in transport involves improving the nature of the connection even when the existing connection is functioning satisfactorily. An *upgrade* in QoT occurs when a connection is already in place and functioning, but a better choice becomes available, such as moving from Bluetooth to IrDA FIR to optimize speed. When a connection is forced to a lower quality link because the current link ceases to be available, or ceases performing well, we refer to this situation as a *downgrade* in QoT. Downgrades are generally preferred to dropping the overall connection. Upgrade and downgrade transport switching is handled through QoT Switch and Resume packets.

Switch packets are used to coordinate switching the underlying transport during a transport upgrade or downgrade operation. A device desiring a transport switch first queries the remote device to see if a transport switch is possible. It then must create a transport connection on the requested transport to the remote device (if a transport connection is not already established) before sending a Switch_Request packet. If a Switch_Request is rejected, the local device may request a different transport, continue with the current transport, or disconnect the connection. Switch packets are connection dependent, but not transport-dependent. In other words, Switch packets are not required to be sent over the currently active transport.

If the active transport is abruptly disconnected during a QoT connection, the primary device is responsible for establishing

another transport connection using any available transport. After a new transport connection has been established, the two devices exchange data sync points via `Resume_Request` and `Resume_Accept` packets. This packet exchange allows the devices to continue communicating without losing the overall session connection. If a dropped connection is not resumed within a certain time (implementation specific), QoT may notify the session layer that the connection has been dropped.

VIII. INTELLIGENT TRANSPORT SELECTION

When a new transport becomes available a switch should only occur if at least one of the two devices considers it an upgrade. Perceiving a transport as an upgrade may not be consistent across devices since the selection of the “best” transport depends highly on user preferences. For some users, the preferred transport is the fastest, for others, the most power conservative, or perhaps the least costly, or the one with longest signal range. Decision making in a transport switching system must take user input on criteria such as these and translate that input into a decision on a single preferred transport. Some have approached the problem using policy-based schemes [18], others with fuzzy logic [8]. QoT performs optimal link selection with a novel approach to decision making which we call Prioritized Soft Constraint Satisfaction (PSCS).

The interface to PSCS allows users to set a range of constraints over performance values that are desirable, acceptable, or unacceptable for each criteria. For example, regarding speed, a user may select 8 - 54 Mbps to be desirable, 4 - 7.9 Mbps to be acceptable and less than 4 Mbps to be unacceptable. Additionally, the user may attach priority to each criterion.

PSCS computes the preferred transport in a negotiation style. If no transport meets all the desirable settings then PSCS compromises on the preferences of the lowest priority criteria first until a satisfying solution is reached. Performance measurement of the current connection enables QoT to decide when to switch transports. If the measurement falls into a lower range of constraints for a sufficient time period, then QoT may attempt to switch transports if one is available with better performance potential.

IX. CONCLUSION

Our research into Quality of Transport (QoT) has yielded: 1) An initial specification for the QoT architecture, including the ability to dynamically and transparently switch transport protocols. [19]; 2) Prototype implementations on several handheld device platforms involving selected transport protocols using disparate physical layers; 3) Demonstration of transport discovery, service discovery, data object exchange, and transport switching in heterogeneous wireless environments; 4) A mechanism for intelligent autonomous transport switching.

A number of issues remain for further investigation, including: 1) A study of QoT performance issues including discovery and data exchange phases; 2) A study of the optimality of transport switching rates, including the trade-offs inherent in

packet exchange duration versus switching requirements; 3) A study of the optimal time delay between performing transport testing for non-active transports.

Wireless multi-transport devices will continue to proliferate. Preliminary successes with our design and implementation of QoT suggest that dynamic intelligent autonomous transport switching can help to optimize user experience and session layer performance in such heterogeneous environments.

REFERENCES

- [1] Bluetooth Special Interest Group, Specification of the Bluetooth System, version 1.1, volumes 1 and 2, available from <http://www.bluetooth.com>, Feb. 2001.
- [2] Institute of Electrical and Electronics Engineers, Wireless Standards Package (802.11), available from <http://www.ieee802.org/11>, 1999.
- [3] IrDA, Serial Infrared Link Access Protocol, IrLAP, Version 1.1, Infrared Data Association, Walnut Creek, California.
- [4] IrDA, Link Management Protocol, IrLMP, Version 1.1, Infrared Data Association, Walnut Creek, California.
- [5] IrDA, IrDA Object Exchange Protocol, IrOBEX, Version 1.3, Infrared Data Association, Walnut Creek, California.
- [6] M. Baker, X. Zhao, and J. Stone, “Supporting mobility in MosquitoNet.” Proceedings of the 1996 USENIX Technical Conference, Jan. 1996, pp. 120-127.
- [7] S.B. Barnes, R.W. Woodings, C.D. Knutson. “Transport discovery in wireless multi-transport environments.” Proceedings of the IEEE Wireless Communications and Networking Conference 2003 (WCNC '03), New Orleans, Louisiana, March 17-19, 2003.
- [8] P.M.L. Chan, Y.F. Hu and R.E. Sheriff. “Implementation of fuzzy multiple objective decision making algorithm in a heterogeneous mobile environment.” Proceedings of the IEEE Wireless Communications and Networking Conference, Orlando, FL, USA, March 2002, pp. 17-21.
- [9] R.J. Clark, M.H. Ammar, and K.L. Calvert, “Protocol discovery in multiprotocol networks.” Mobile Networks and Applications, vol. 2, no. 3, 1997, pp. 271-284.
- [10] J. Foerster, E. Green, S. Somayazulu, and D. Leeper, “Ultra-wideband technology for short or medium-range wireless communications.” Intel Technology Journal, no. Q2, May 2001, p. 11
- [11] G.H. Forman and J. Zahorian, “The challenges of mobile computing.” IEEE Computer, April 1994, pp. 38-47
- [12] J.C. Funk, H.R. Duffin, L. Dai, C.D. Knutson. “Inverse multiplexing in short-range multi-transport wireless communications.” Proceedings of the IEEE Wireless Communications and Networking Conference 2003 (WCNC '03), New Orleans, Louisiana, March 17-19, 2003.
- [13] S. Graham, B. Miller, and J. Rusnak, IBM Corporation, Discovering Devices and Services in Home Networks, <http://www-3.ibm.com/software/pervasive/tech/networking.shtml>, June 1999
- [14] J. Inouye, J. Binkley, and J. Walpole, “Physical media independence: System support for dynamically available network interfaces.” OGI CSE Technical Report 97-001, Jan. 1997.
- [15] B. Noble, “System support for mobile, adaptive applications.” IEEE Personal Communications, Feb. 2000, pp. 44-49.
- [16] K. Siwiak, “Ultra-wide band radio: Introducing a new technology.” Proceedings of the IEEE Semiannual Vehicular Technology Conference (VTC), vol. 2, Rhodes, Greece, May 2001, pp. 1088-1093.
- [17] M. Stemm and R.H. Katz, “Vertical handoffs in wireless overlay networks.” ACM Mobile Networking and Applications, no. 4. ACM, 1999, pp. 335-350.
- [18] H. Wang, R.H. Katz, and J. Giese. “Policy-enabled handoff across heterogeneous wireless networks.” 2nd IEEE Workshop on Mobile Computing and Applications (WMCSA 1999), New Orleans, February 1999.
- [19] R.W. Woodings, S.B. Barnes, C.D. Knutson. Quality of Transport (QoT) Protocol Specification v1.0. Mobile Computing Laboratory, Brigham Young University, Provo, Utah, Aug. 2002.
- [20] R.W. Woodings, D. Joos, T. Clifton and C.D. Knutson. “Rapid heterogeneous ad hoc connection establishment: accelerating bluetooth inquiry using IrDA.” Proceedings of the IEEE Wireless Communications and Networking Conference 2002 (WCNC '02), Orlando, Florida, March 18-21, 2002.