Friend Relay: A Resource Sharing Framework for Mobile Wireless Devices

Hillary Caituiro-Monge, Kevin Almeroth, Maria del Mar Alvarez-Rohena
Department of Computer Science
University of California
Santa Barbara, CA 93106-5110

{hcaituiro, almeroth, malvarez}@cs.ucsb.edu

ABSTRACT

The rising popularity of wireless devices opens exciting possibilities for users to share their resources. However, there exists no end-to-end, general-purpose, resource sharing framework for such devices. Existing resource sharing solutions address only part of the overall challenge and barely consider mobile wireless devices' critical characteristics, such as mobility, CPU capability, and power limitations. Furthermore, they do not have mechanisms to *control* the degree to which resources are shared. This lack of control likely leads to over-usage of shared resources, therefore, reducing computing capacity and draining the batteries of devices that are sharing resources. In this paper, we present Friend Relay, a resource-sharing framework for mobile wireless devices that offers automatic publishing, discovery and configuration, as well as monitoring and control. Our contribution is to add monitoring and control as mechanisms to manage the utilization of shared resources. To evaluate Friend Relay, we quantify the benefits of monitoring and controlling resource usage for Internet access sharing. For our evaluation, we developed a prototype of Friend Relay. In addition, we prototyped "Internet Sharing", a service to allow users to share the ISP subscription of another user. Our evaluation focuses on the performance of this service with and without the Friend Relay monitoring and control mechanisms. Our results show that mobile devices that either share or use resources can benefit significantly from using our Friend Relay architecture.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communications

General Terms

Management

Keywords

Resource sharing, wireless, WLAN, control, monitoring

WMASH'06, September 29, 2006, Los Angeles, California, USA.
Copyright 2006 ACM 1-59593-470-7/06/0009...$5.00.

1. INTRODUCTION

In today's increasingly mobile world, laptops, Personal Digital Assistants (PDAs), cell phones, and numerous other mobile wireless handheld devices have become ubiquitous. People carry them everywhere and can use them at any time. As a result, mobile devices constantly come into contact with other mobile devices. As shown in Figure 1, this interaction can provide seamless access to well-known resources, as well as provide access to novel services, features, and content. The usefulness of a single device can be extended through access to and utilization of other devices. This idea has significant potential and leads to the possibility of several new use scenarios. For example, two friends are in a coffee shop where one of them has access to the Internet and is willing to share with the other. In another example, a conference participant offers to share his or her digital copy of the proceedings with a colleague. In an additional example, a group of friends traveling from Santa Barbara to Europe want to watch a movie that one of them has on her laptop. In a final example, a file sharing service runs to allow one friend to copy all of the pictures of another friend. These examples serve to illustrate the fact that resource sharing in mobile wireless networks is a powerful idea; there are potentially numerous resources that could be shared; and there are numerous ways in which resource sharing networks could be built.

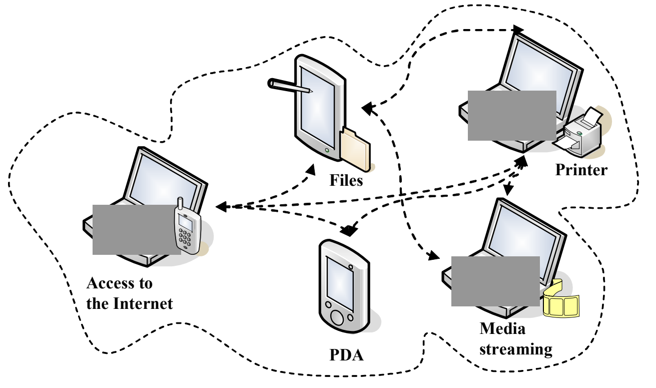

Figure 1 A mobile wireless network: devices with shared resources.

Sharing resources in mobile wireless networks poses several challenges that are different from those in structured wired networks. One difference is that, in the former case, there may not be access to services like the Dynamic Host Configuration Protocol (DHCP), the Domain Name System (DNS), or access to other servers whose function is to support resource sharing. This difference creates a difficult problem whose solution resides in zero configuration mechanisms [13]. In addition, and most importantly, in contrast to structured networks, mobile wireless network members and their resources are severely constrained due to their limited computing capacity and battery power, as well as their mobility. Sharing resources under these constrained conditions raises new challenges.

As part of resource sharing, devices share their computing power and battery power. Users of the devices that are sharing should be able to observe the status and resource consumption levels of those using the resources. Furthermore, users should be able to control the level at which their resources are used. These requirements create the need to establish monitoring and control mechanisms to ensure the feasibility of resource sharing. The lack of these mechanisms is likely to result in a decline in the number of users willing to share due to the uncertainty of what is happening to their devices while participating in the sharing process. Similarly, a lack of control mechanisms could prevent users from sharing their resources because they will not be able to control how much of their resources are available for sharing. Monitoring and control create an observable and controllable environment for users when sharing their resources. In addition, as a final component of a sharing system, there is a real need to encourage users to share their resources with their mobile neighbors through incentive mechanisms.

The idea of sharing resources in mobile wireless networks creates several challenges, including: auto publishing, discovery and configuration, as well monitoring and control [2] [4] [13] [9]. In fact, the first three efforts are core parts of the work of the Internet Engineering Task Force (IETF) ZeroConf group [13]. Apple's Bonjour is a successful and widely used implementation of the ideas addressed in the IETF ZeroConf group [9]. Although, the ideas and solutions are valid and function correctly for both structured and unstructured networks, the ideas are not fully suitable for mobile wireless networks. Neither ZeroConf nor Bonjour address monitoring and control for resource sharing in mobile wireless networks. In addition, other work has addressed resource sharing in mobile wireless networks for specific tasks such as web sharing [11] and file sharing [12]. However, their only focus is to minimize energy consumption. Another effort presents a solution for general resource sharing as a "meta-protocol" for ad-hoc sharing [10]. However, this solution does not provide a general-purpose framework for resource sharing in mobile wireless networks.

In this paper, we present Friend Relay, a resource-sharing framework for mobile wireless networks. Friend Relay provides monitoring and control as its major contributions in addition to any zero configuration features; in particular, we start with automatic publishing, discovery, and configuration. Furthermore, as a way of balancing responsibility, Friend Relay distributes monitoring and control tasks between a resource provider and a resource consumer. For simplicity, we will use the term "provider" meaning a device that shares a resource and the term "consumer" meaning a device that uses a shared resource.

To evaluate Friend Relay, we have investigated the benefits of monitoring and control functionality. In particular, our goals are to confirm that fine-grained control can be implemented, and to quantify its overall effectiveness. To this end, we have developed a prototype of Friend Relay. In addition, we developed support for Internet sharing, a service that allows access to the Internet through the relay of packets, a service similar to what a Network Address Translator (NAT) provides, but with Internet sharing, the goal is to share a friend's ISP subscription. The evaluation results, obtained by sharing Internet access with and without Friend Relay monitoring and control, show that mobile devices either sharing or using resources can benefit significantly from our Friend Relay architecture.

Our contribution in this paper then, is to introduce monitoring and control as mechanisms to manage the usage of shared resources. Monitoring tasks enable users and applications to observe the status of shared resources by gathering and visualizing relevant data. Control tasks enforce policies intended to limit resource usage in order to preserve a device's computing capacity and battery power.

The remainder of this paper is organized as follows. Section 2 covers background for this work by presenting a detailed scenario and discussing the related work in the areas of resource sharing and discovery. Section 3 describes our Friend Relay framework requirements. Section 4 covers the design and implementation of Friend Relay and presents our service for Internet sharing. Section 5 presents the evaluation of our solution. Section 6 describes some broader implications of resource sharing. The paper is concluded in Section 7.

2. BACKGROUND

To motivate the need for monitoring and control in resource sharing, we first present a detailed scenario. We then review related work to put our efforts into context.

2.1 Scenario

The following example describes a scenario in which three friends with mobile devices share their resources. First, we establish the context for the scenario and the mobile devices. Next, we explain how the users share their resources using current technology and mechanisms. We then describe the problems of these solutions. Finally, we describe an alternative solution that will become the basis for our approach.

Three students, Mary, Lisa, and Joe often see each other during the course of a day, from joint classes, to time in the lab, and even outside of school. Mary likes to carry her iPod and PDA almost all of the time, and her PowerBook G4 some of the time. Lisa usually has her small laptop and printer with her when she is attending classes and conferences. Joe likes to always be connected to the Internet, thus, he has his Linux laptop with three mechanisms to connect to the Internet: (1) through his laptop's wireless card that he has registered with several wireless providers, (2) an Ethernet cable in case he is lucky enough to find an open port, and (3) if there is no wired or wireless connection, he has a cell phone that has Internet connectivity.

When Mary, Lisa, and Joe meet, they often share their resources. Mary and Joe sometimes forget to bring a copy of the PowerPoint slides for a lecture. Mary keeps a large, diverse music library on her laptop. And while Mary and Lisa have wireless access to the Internet at school and at home, they do not have access to the Internet anywhere else.

When these three students want to share any of each other's resources, they must first create a wireless ad hoc network so their computers can communicate. Creating an ad hoc network requires that they choose a network SSID and find a way to choose IP addresses. In addition, they must configure each resource for sharing. For instance, in order for Mary to print using Lisa's printer, Lisa must share her printer and Mary has to install an appropriate driver. Only then will Mary be able to print. In order for Lisa to listen to music from Mary's iTunes library, Mary must run software to share her iTunes folder. Lisa can then either use iTunes for Windows or access a shared folder on Mary's laptop. In order for Lisa and Mary to access the Internet through Joe's laptop, Joe has to enable Internet sharing on his computer.

With minimal effort, these students appear to be able to share their resources without major difficulty. However, there are still several subtle but important problems. First, they are still using some amount of manual configuration. They are manually creating an ad hoc network and installing a printer driver. Even if these steps are acceptable in a structured environment where resources are relatively stable, these steps are cumbersome in dynamic environments where resources and network members appear and disappear fairly often. Second, while the three students trust each other, if they want to create an open network for others to share, there is no way for them to control the degree to which they share their resources. Either they grant full access to their ad hoc network, or they block access completely. A lack of monitoring and control mechanisms would both be a security risk and would be unfair to the devices that act as resource providers. Again, in structured networks, monitoring and control are not as critical due to the more capable devices and readily available power sources. Furthermore, while sharing Internet access inside a university or other location with public Internet access might not be of concern, if access is provided through a private ISP that charges for access, there are some interesting considerations.

We envision a scenario where users are able to share their resources without manual configuration. At the same time, they are able to monitor and control their resource usage.

2.2 Related Work

We have identified several problems that arise when sharing resources in mobile wireless networks. Some of these have been addressed and have promising solutions such as ZeroConf [13] and Bonjour [9] for both structured and unstructured networks. There are also solutions such as a power-aware web proxy [10] and file sharing service [11] that have focused on energy saving strategies. Furthermore, there is an attempt to address resource sharing as a "meta-protocol" for ad hoc networks. This section briefly describes these approaches and then contrasts them with Friend Relay architecture.

The IETF ZeroConf [13] working group leads an effort to achieve standards and techniques for the automatic creation of IP networks with neither manual configuration nor special servers (e.g., DNS or DHCP). These techniques are valid for both structured and unstructured networks. However, they do not provide solutions to preserve and control resources in constrained devices. Our approach addresses such problems by introducing monitoring and control of shared resources.

Bonjour is an open technology implemented by Apple for resource sharing in wired and wireless environments [9]. It has properties such as instant networking, dynamic service discovery, and zero-configuration. It uses standard technologies such TCP/IP and DNS to achieve its purpose. However, Bonjour does not have monitoring and control mechanisms for resource sharing in mobile wireless networks. Potentially, a device using a shared resource could drain a resource provider's CPU, memory, and battery power.

McKnight et al. propose a sharing protocol for wireless grids as a meta-protocol for ad hoc resource sharing [11]. They consider four elements for resource sharing: 1) resource description, 2) resource discovery, 3) a clearing mechanism, and 4) a coordination mechanism. This work does not yet have a working prototype, and similar to Bonjour, it does not provide monitoring and control of shared resources as key system services.

PAWP is the Power Aware Web Proxy for wireless LAN clients [10]. It provides for a savings of more than fifty percent of WLAN interface power consumption through web traffic scheduling. This technique provides a mechanism for creating intervals of high traffic and no traffic. Thus, WLAN interfaces can switch to a low power state during no-traffic periods. This work differs from our approach in the fact that we intend to deliver an end-to-end solution for resource sharing. Instead, PAWP is a specific-purpose solution that neither addresses zero configuration nor resource control. PAWP does not allow a user to set a limit on the bandwidth used by peer wireless LAN clients as Friend Relay does.

Ding et al. have proposed peer-to-peer file sharing over mobile ad-hoc networks [12]. Their work consists of evaluating five routing approaches: 1) broadcast over broadcast, 2) broadband, 3) DHT over broadcast, 4) DHT over DHT, and 5) DHT. They conclude that for these kinds of ad hoc environments, cross-layer protocols perform better than layered protocols. Our solution takes a layered approach that works over existing deployed protocols making it easily deployable. And again, our focus is on adding monitoring and control functionality.

3. FRIEND RELAY DESCRIPTION

We now describe Friend Relay's basic and extended requirements for a resource sharing framework for mobile wireless networks. While basic requirements are necessary to consider for this kind of environments, extended requirements are highly desirable but not necessarily critical.

Friend Relay is a resource sharing framework for mobile wireless devices that enable devices both to share their resources and to use other devices' shared resources. Figure 2 show a wireless network with devices running Friend Relay agents.

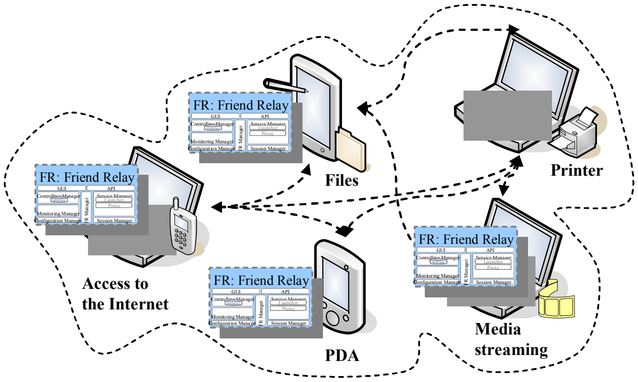

Figure 2 A mobile wireless network with Friend Relay.

In order to fulfill its goals, Friend Relay considers mobile device characteristics such as mobility, constrained computing power, and limited battery power. Mobility generates changing environments in which mobile devices that are brought together, create mobile networks that are stable for only short periods of time. In these networks, devices can join and leave at any point in time. Constrained computing capacity and limited battery power restrict the use of computing resources. With these considerations in mind, we now describe the basic requirements for resource sharing in mobile wireless devices. Thereafter, we offer a set of extended requirements that address desirable properties that relate to economics and legal issues.

3.1 Basic Requirements

The set of basic requirements includes features that every resource sharing infrastructure for mobile wireless networks should have. Such requirements include: automatic resource publishing, discovery and configuration, and system monitoring and control. There are two additional requirements that should be considered as part of all of these features. These are power awareness and security. We now justify these requirements.

- *Automatic resource publishing* allows mobile devices to advertise their resources to their neighbors through meaningful descriptions. Automatic broadcasting or multicasting of such descriptions is critical due to the dynamic nature of mobile wireless networks. A difference of structured networks is that they use directory servers to keep a list of resources. These servers are effective because the shared resource last for long periods (e.g., months or even years). In unstructured networks, resources are available from minutes to hours, making the use of any kind of directory server impractical. Publishing should advertise resource descriptions so to allow others to adequately recognize and classify them. However, these descriptions should not compromise the security of the mobile device owning the resource. Furthermore, resource description should not compromise the mobile device user's privacy.

- *Automatic resource discovery* allows mobile devices to discover the resource descriptions of their neighbors. Resource discovery should be automatic due to the dynamic nature of mobile wireless networks. Devices discover shared resources by listening to broadcast or multicast messages from neighbor devices instead of querying directory servers. Through this mechanism, mobile devices are able to maintain an updated pool of resources published by nearby mobile devices.

- *Automatic resource configuration* enables mobile devices to configure both local resources for sharing and remote resources for local usage without unnecessary intervention by users. A set of policies determines user preferences for how to handle resource configuration. Initially, the system has a default set of policies that is the result of a set of best practices. Users can modify these policies as desired. Automation is critical due to the frequency of resources appearing and disappearing, as well as their diversity.

- *Automatic resource monitoring* enables mobile devices to provide information about shared resources to both users and applications through Graphical User Interfaces (GUIs) and Application Programming Interfaces (APIs), respectively. Monitoring should provide users with a complete and clear snapshot of the resource sharing system. This requirement is very important due to the constantly changing environment. Furthermore, the system should alert users or applications of relevant events that might be of interest. These kinds of monitoring mechanisms will encourage users to share their resources and to use shared resources because of a better understanding of what resources are available.

- *Automatic resource control* enables mobile devices to manage resource sharing by means of policy enforcement. Resource control uses resource monitoring to collect information about the system in order to evaluate policy and to trigger enforcement control mechanisms, if necessary. Control is a critical requirement, since a system must do more than passively monitor. It must proactively implement the sharing policies as defined by the user. With limited computing capability and scare battery power, resource control is a critical tool to limit the usage of shared resources in order to preserve vital resource for the hosting mobile device.

- *Security awareness* is a vertical and horizontal requirement for ensuring a trusted environment for each phase of the sharing process. The system should set a clear set of policies for the phases of publishing, discovery, configuration, and usage. For example, usage requires more credentials than publishing. In addition, publishing has several flavors that include publishing for everyone or publishing only for a group sharing a set of characteristics, such as multicast group or a secret key. Similar considerations are valid for the discovery, configuration, and usage of resources. Mechanisms, such as encryption and passwords, should be used to create a trusted environment.

- *Power awareness* is a critical requirement that should be included in every part of the system. Publishing, discovery, configuration, monitoring, control, and security all should be designed and implemented with consideration for the limited CPU, memory, and battery power available. Since a heavily shared system could easily have its resources fully utilized and battery power quickly drained, the need needs to balance the ability of other users to use shared resources with the needs of the primary user. As a result, providing power saving mechanisms is critical to encourage users to share their resources. Monitoring and control are then used to enable users to limit use of their resources.

3.2 Extended Requirements

The set of extended requirements include features that are highly desirable but not necessarily critical for a resource sharing infrastructure. Such requirements include: context awareness, user centricity, economics awareness, and legal awareness. We now justify these requirements.

- *Context awareness* enables mobile devices to be attentive of their operational environment in terms of their position, users' activities, battery power level, and neighbors. This contextual knowledge empowers resource sharing systems to adapt their functionality as a response to changes in these parameters. For example, for a user who is in the middle of a presentation, the resource sharing system should be able to select and alert the user only about a small and select amount of shared resources that are relevant for her or him at that instant. An example of a relevant shared resource in this circumstance would be a shared resource that broadcasts the slides of the presentation.
- *User centricity* in a resource sharing system has the potential to offer an even better resource sharing experience to users. For example, for a user that is listening to music on a mobile device, the system can inform the user about other nearby users with related music. In summary, sharing systems are more likely to adapt better to users' resource needs if they are focused on the needs and wants of the user.
- *Economics awareness* enables resource sharing systems to consider economics in the resource sharing experience. We define "resource sharing economics" as the direct and indirect costs resulting from resource sharing. For example, for a user whose Internet provider bills him or her by counting the number of bytes transmitted or by amount of time online, the decision to share Internet access with other users could result in a more expensive bill. This scenario, and others similar to it that incur direct or indirect costs, will discourage users from sharing their resources unless they receive direct compensation. Therefore, the solution to overcome this problem is to include economics awareness in the resource sharing system. The solution should be able to provide the system with mechanisms to characterize, compute, and divide the cost of sharing resources. In addition, the system should provide users with billing and payment mechanisms to allow them to share these costs. Costs can be represented by money, points, or reputation. Finally, an economics-aware resource sharing system will create new opportunities for mobile users. For example, in the case of two users sharing their music libraries and interchanging songs, the resource sharing system could bill on behalf of the song owner. Furthermore, users sharing songs could receive a commission for every song they deliver, thus, becoming content "micro-distributors". These distributors could potentially become part of a new approach to delivering content both free and for pay.
- *Legal awareness* allows mobile devices to change their policies and behavior according to the local law. Law changes are more frequent for mobile users than for static users. In addition, resource sharing has several legal implications such as copyrights for digital books, author rights for music, and contractual obligations for Internet access. An ideal resource sharing system should provide mechanisms to enable users to adhere to relevant legal requirements.

4. FRIEND RELAY DESIGN AND IMPLEMENTATION

This section covers the design and implementation of our Friend Relay system. We present the architecture and the full protocol stack. Thereafter, we describe a use case in which Friend Relay is used to share Internet access.

4.1 Architecture

Friend Relay is an end-to-end application layer solution for resource sharing that depends on existing standards and protocols. Friend Relay provides, different from other solutions, mechanisms to monitor and control the use of, and access to, shared resources. In addition, Friend Relay provides, similar to other solutions, strategies for auto publishing, discovery, and configuration.

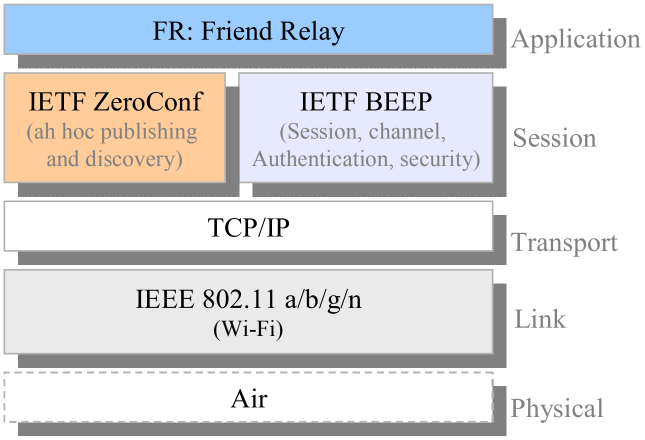

Figure 3 Full protocol stack for Friend Relay.

Figure 3 shows the full protocol stack for Friend Relay. This stack uses several well-known and widely deployed standards among its components. The rationale behind this decision is to facilitate the deployment of Friend Relay in existing mobile devices. The stack starts with Friend Relay at the top. Friend Relay includes mechanisms to monitor and control, in addition to auto publishing, discovery, and configuration functionality. For this second set of functions, Friend Relay depends on implementations of the IETF ZeroConf [13] and IETF Blocks Extensible Exchange Protocol (BEEP) [14] standards for full functionality. IETF ZeroConf provides ad hoc automatic publishing, discovery, and configuration of network elements, including shared resources. With IETF ZeroConf solutions, a mobile device running Friend Relay can advertise shared resources, discover other mobile devices' shared resources, and configure them depending on the type of resource. In addition, IETF BEEP provides session handling that includes authentication, security, multiple channels, and profiles. Both, IETF ZeroConf and IETF BEEP use TCP/IP as the transport and network protocols, respectively. Finally, Figure 3 shows IEEE 802.11 as the link layer protocol.

The Friend Relay architecture is a bundle of several components that includes both server and client functionality. This bundling allows any mobile device to share its own resources, and to use others' shared resources at the same time. Friend Relay neither explicitly supports nor is optimized for non-mobile wireless devices dedicated to a shared pool of resources. However, it can work under these circumstances, even in wired networks. Figure 4 shows the architecture of Friend Relay and its relationship with IETF ZeroConf and IETF BEEP. Friend Relay has the following components: Configuration Manager, Session Manager, Monitoring Manager, Controlling Manager, Service Manager, Friend Relay Manager, GUI, and API. In the next paragraphs, we explain the function of each component.

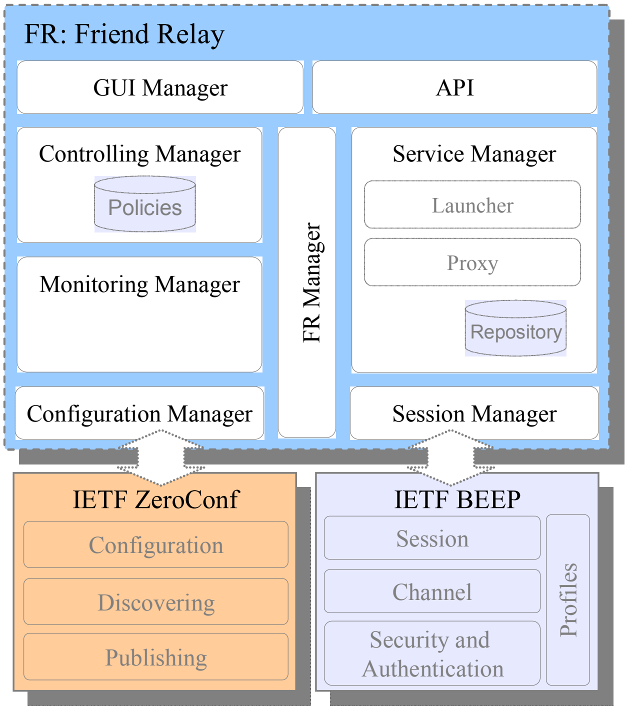

Figure 4 Friend Relay Architecture.

- The *Friend Relay Manager* (FRMan) is responsible for initializing and coordinating the functions of all the other components within Friend Relay. FRMan connects to each of the other components. The exchange of messages among components uses FRMan as the communication intermediary.

- The *Configuration Manager* (ConfMan) performs automatic publishing, discovery, and configuration with assistance of an IETF ZeroConf compliant implementation, such as Bonjour [9]. In Friend Relay, *automatic publishing* consists of advertising services by using their profile attributes (e.g., name and description) to build a description and their policies (e.g., open/private and text/encrypted) to establish a scope. In contrast, *automatic discovery* consists of scanning for neighbors' shared resources and notifying the Resource Manager about these findings. Finally, *automatic configuration* configures either a local resource for sharing or a remote shared resource for local usage. In addition to these three tasks, ConfMan installs drivers or software for shared resources to work properly on the local device. If shared resource policies require confirmation to perform installations, ConfMan will prompt the user for authorization. In addition, ConfMan will verify the authenticity of drivers or software by checking their certificates.

- The *Session Manager* (SessMan) is responsible for authentication, security, and session handling with assistance from an IETF BEEP compliant component, such as BEEPCore [27] or PermaBEEP [26]. The Resource Manager would request SessMan to create a session between two mobile devices to complete the configuration process. This session is used to provide a secure channel to exchange configuration parameters, as well as drivers and software that are needed by the shared resource to work properly in the mobile device.

- The *Service Manager* (ServMan) is responsible for handling shared resources and candidate shareable resources. One task for ServMan is to keep a list of local shared resources, potential local shareable resources, and neighbors' shared resources. Another task for ServMan, depending on resources profiles and policies, is to prompt users for permission or automatically do the following tasks: a) advertise resources by sending messages to the Configuration Manager; b) ask the Configuration Manager to configure potential local shareable resources and set them as shared; c) configure neighbors' shared resources for local usage through the Configuration Manager and Session Manager. ServMan also notifies the Monitoring and Control managers when resources join or leave. In addition, ServMan is able to launch services (e.g., a NAT translator) in order to share resources (e.g., Internet access). Furthermore, ServMan can run proxies for shared resources that are not prepared to operate in networks. Resource proxies provide resource emulation, giving the impression that they are in fact local resources when they are not.

- The *Monitoring Manager* (MonMan) collects data about shared resources to send to the Control Manager and to the GUI Manager. MonMan relies on adapters, which are specific for each kind of resource, to collect the desired data from shared resources. If neighbors advertise new kinds of resources, the Resource Manager will ask the Configuration Manager to request monitoring adapters from these neighbors. In the absence of specialized adapters, MonMan uses general purpose adapters that are able to collect generic data (e.g., connection duration). MonMan can also send requests to a neighbor's MonMan to collect local data on its behalf. In addition, MonMan collects system-wide data such as the number of connections, networks, users, as well as traffic for each user.

- The *Control Manager* (ConMan) checks policies against the data collected by the Monitoring Manager. This data can lead to enforcement changes in the functionality of shared resources. ConMan uses control adapters, which are specific for each kind of resource, to perform these functional changes. ConMan, similar to the Monitor Manager, uses general purpose control adapters to control shared resources in the absence of resource-specific control adapters. Control enforcement includes limiting network bandwidth and limiting the number of concurrent users. By controlling the usage of shared resources, Friend Relay can reduce energy consumption as well as reserve computing power for usage by the resource provider.

4.2 Case Study

This section describes a use case for sharing a resource using Friend Relay as the sharing architecture. The resource for this exercise is an Internet sharing service that allows one device with Internet access to share its connectivity with other devices. This section first covers the design and implementation of a NAT box that redirects Internet traffic from one mobile device to another. The NAT box is a key component of the "Internet Access Resource". Second, this section describes the use of Friend Relay to share the Internet Access Resource.

Internet sharing, shown in Figure 5, is a key component of the Internet Access Resource. It implements both Network Address Translation (NAT) and Network Address and Port Translation (NATP). For ICMP packets, it performs Source and Destination NAT (SNAT and DNAT), and for UDP (SNAPT) and TCP (DNAPT). The main goal behind Friend Relay NAT is to implement a general purpose NAT device with support for NAT-friendly applications [7]. Address and port translation mechanisms are based on the IETF RFCs for ICMP [5], UDP [6], and TCP [8] operation within a NAT environment.

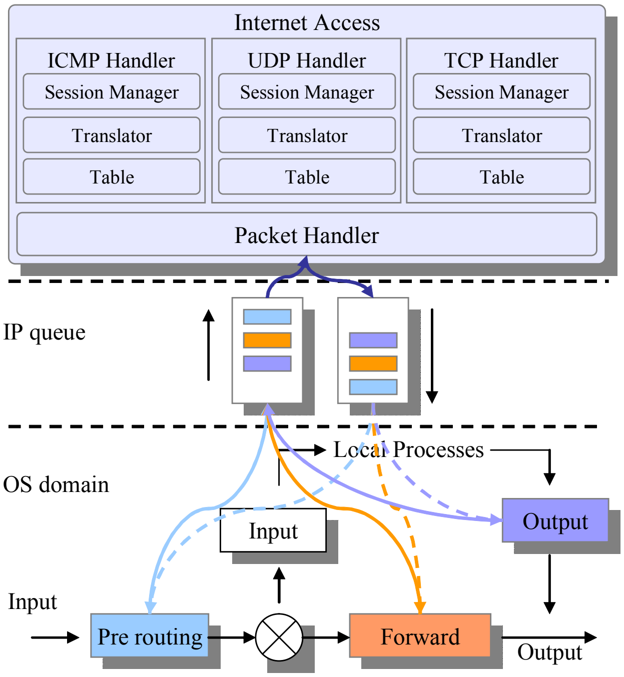

Figure 5 Internet sharing architecture.

Internet sharing does not need Friend Relay to work. However, it would provide only basic packet redirection and address translation. Internet sharing alone neither offers monitoring and control, nor does it provide automatic publishing, discovery, and configuration. The case is different if Friend Relay shares Internet connectivity as a resource (Figure 6). In this case, mobile devices will be able to advertise, discover, configure, monitor, and control Internet sharing. For example, mobile devices sharing Internet connectivity can now set limits on the bandwidth for mobiles devices using this shared resource.

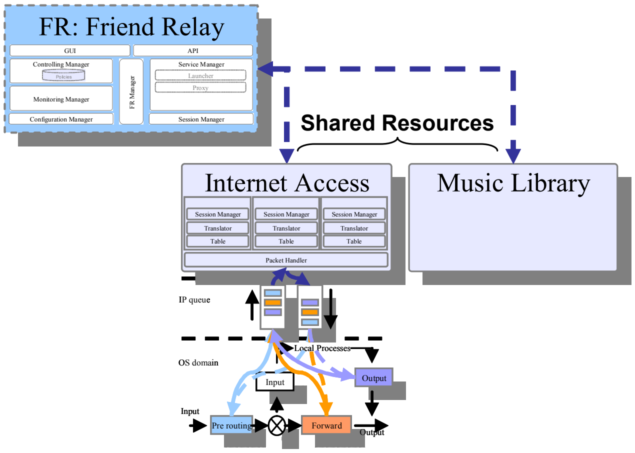

Figure 6 Friend Relay relationship with shared resources.

5. EVALUATION

The goal of our evaluation is to show the benefits of having monitoring and control functionality when sharing resources in mobile wireless devices. In particular, our focus is on quantifying Friend Relay's ability to accurately control bandwidth usage.

In order to assess the benefits of Friend Relay, we measured the bandwidth usage of the Internet sharing application with and without Friend Relay. The traffic load was generated by various downloads of 600 MB files over a period of 15 minutes. The experiment ran several times on two laptops, one acting as the resource provider (the server), and the other, the resource consumer (the client). In the first set of experiment, the client generates traffic but the server only sends acknowledgements.

Figures 7 and 8 show the bandwidth usage of the client with and without Friend Relay. As expected, when no control is imposed, Figure 7, the client uses as much of the server bandwidth as it can, possibly exhausting the server's resources. In contrast, by using Friend Relay to control the bandwidth usage, Figure 8, the client's bandwidth stays within the desired limit of 800Kbps.

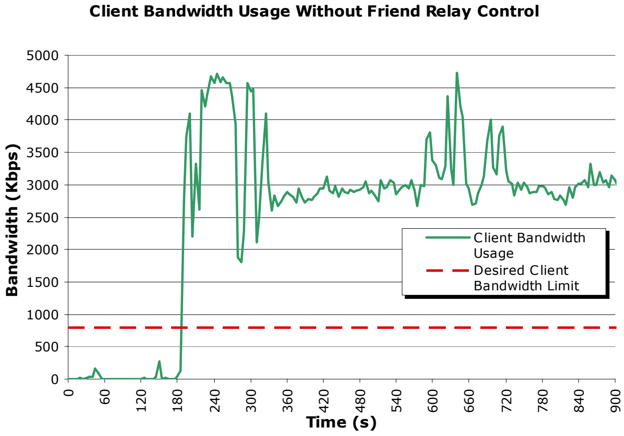

Figure 7 Client bandwidth usage without bandwidth control.

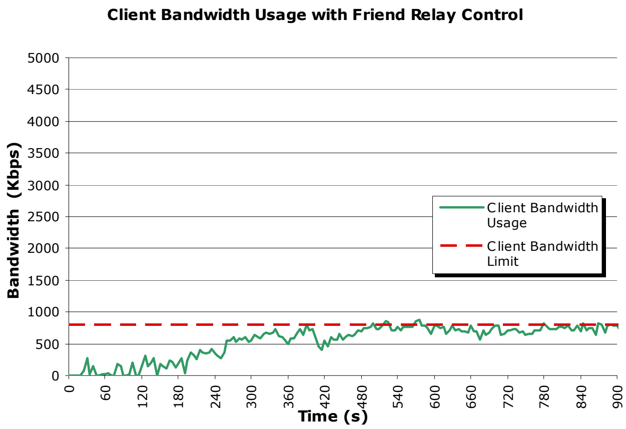

Figure 8 Client bandwidth usage with bandwidth control.

In the second set of experiments, both the server and the client generate Internet traffic. Figure 9 shows that during the server's idle period, the client consumes as much bandwidth as it can. As soon as the server starts to generate traffic, the client's bandwidth usage decreases. The client, however, still tries to send as much traffic as the channel will allow. Figure 10 shows the benefit of adding bandwidth control. In this experiment, even when the server is idle, the client's bandwidth utilization is limited to 800 Kbps, thus, not allowing the client to over-utilize the server's resources.

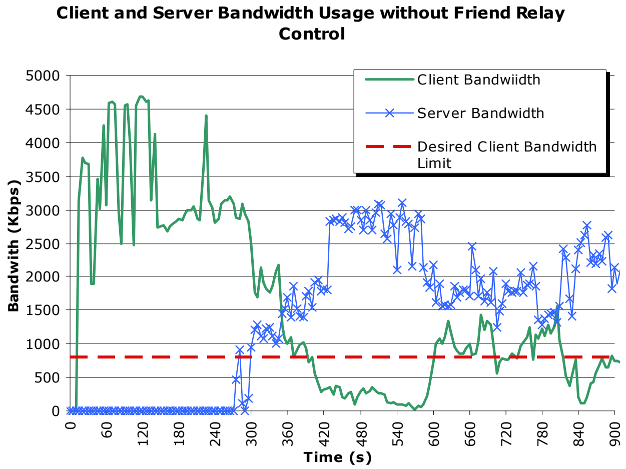

Figure 9 Server and client bandwidth usage without control.

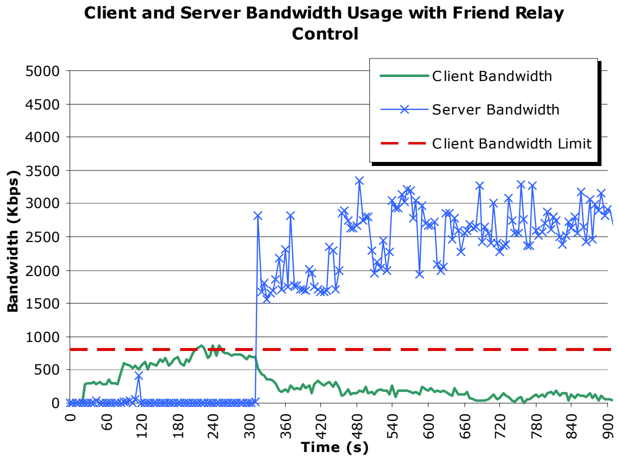

Figure 10 Server and client bandwidth usage with bandwidth control on the client.

Figures 11 and 12 show the bandwidth usage of two clients generating the same traffic load as the previous experiments with and without Friend Relay control, respectively. The server does not generate any traffic in this case. Similar to previous results, the two clients consume all possible bandwidth from the server, as shown in Figure 11. As shown in Figure 12, when Friend Relay's control mechanism is activated, the two clients consume only up to the maximum allowed bandwidth.

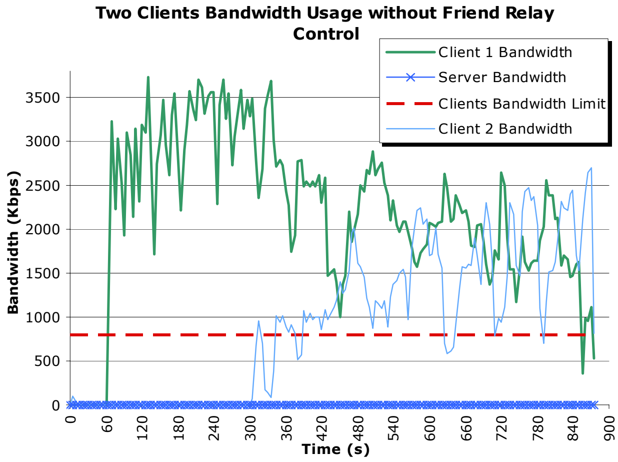

Figure 11 The bandwidth usage of multiples clients without control.

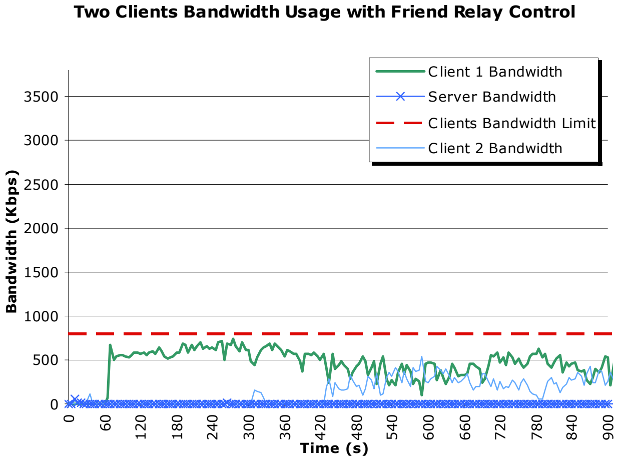

Figure 12 Bandwidth usage of multiple clients with control.

Figures 13 and 14 again consider the case when the server also transmits data. Figure 13 shows how the two clients and the server fight for bandwidth when there is no bandwidth control. In contrast, in Figure 14, the two clients are constrained to a limited amount of bandwidth, but the server is allowed to use the rest.

All figures support the idea that using the Friend Relay system when sharing resources provides effective bandwidth control. By limiting the resource usage of clients, the server saves CPU cycles as well as battery power. As a result, extensive resource sharing in mobile ad hoc wireless networks becomes feasible and attractive to users. Therefore, monitoring and control mechanisms are fundamental mechanisms for resource sharing even before any incentive scheme is added. These two functions provide users the means to observe and control their resources in such a way that they feel in control of their mobile devices and resources.

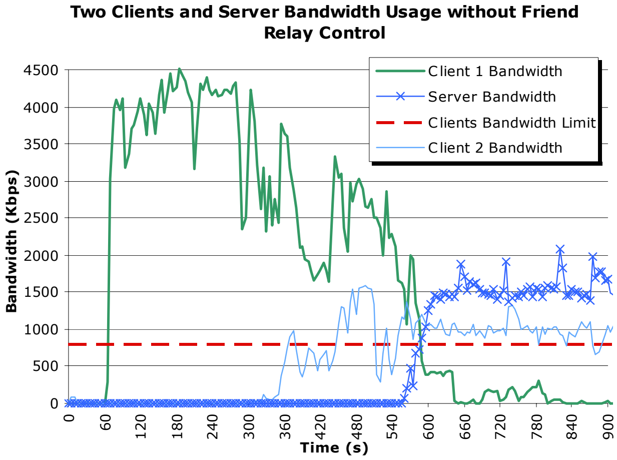

Figure 13 Bandwidth usage of multiples clients and server without control.

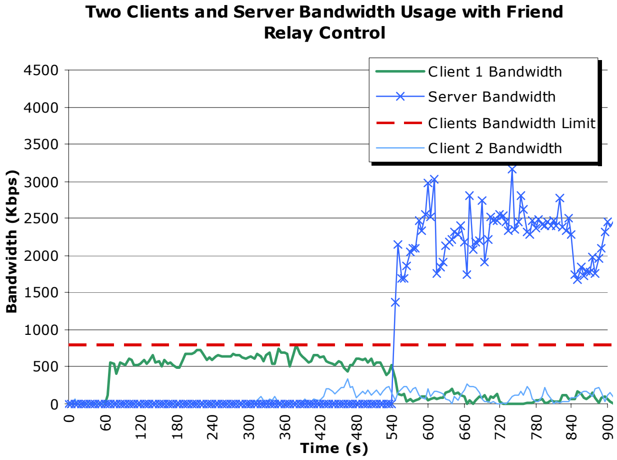

Figure 14 Bandwith usage of muliple clients and server with control.

6. IMPLICATIONS

Friend Relay, as with other resource sharing systems, raises several interesting points for consideration. First, Friend Relay affects the relationship between economics and users sharing their access to the Internet. Users will typically have primary providers, such as ISPs, wireless internet providers, and universities. Some of these providers either charge fees directly for access (e.g., tMobile), or charge indirectly (e.g., universities). The first type of providers uses different criteria to charge for their service. These criteria may include counting the amount of time or traffic that users generate. Users who are being charged on a per-use basis may want a specific, tangible way of recouping the costs of resource sharing. On the other hand, users who share access to the Internet based on time may be infringing and undermining providers' business models when sharing their connection. In this case, providers can either change their business model or charge additional fees to users who are sharing their Internet access. For the later case, providers need to instrument their network with tools to detect when users are sharing their access. Such tools might use techniques such as traffic analysis to detect secondary users. In fact, determining when users are sharing their access to the Internet based on their traffic is an interesting research challenge we leave for future work.

Second, different than sharing Internet access, are scenarios in which, for example, an iTunes's library is shared. iTunes charges users for each downloaded song. Users sharing their iTunes library might be circumventing the iTunes service. Removing the direct, user-to-user sharing option from iTunes could be one way to preserve Apple's business model. This solution, however, does not fix the problem. User can still copy songs using other means such as sharing file system folders. Instead, an alternative may include a model where the primary owner of a song is charged if he or she decides to share the song. In this model, primary song owners may charge other users to listen to these songs or to copy them, thus, in their turn, they pay the extra fees to the distributors. Furthermore, primary song owners may receive some percentage for re-sale of these songs. The ability for users to re-sell songs creates an interesting economic system and several additional research challenges.

We have mentioned in several places how users can share fees or resell content. However, we do not imagine users collecting cash or charging users each time a resource is used. Instead, we envision business models and software tools that allow users to automatically handle the economic interactions resulting from their sharing experiences. Furthermore, while some interactions may require monetary transactions, other interaction may be doable by sharing other kinds of resources, for example, e-currency or reputation points. These interactions may require mobile devices to remember other mobile devices. This scenario also raises several research challenges for future work.

One of the most important aspects to consider is the kind of incentive mechanism [15] [16] [17] that could be used to both support and encourage resource sharing. The correct approach would need to address the problems of misuse or abuse (e.g., users that only consume resources and do not share) in resource sharing [18].

Finally, context awareness, user centricity, economics awareness, and legal awareness are important considerations that should be addressed in the context of resource sharing for wireless mobile devices. How to provide these services while maintaining an intuitive and easy-to-use interface is quite a challenge.

7. CONCLUSIONS

In this paper, we have introduced the Friend Relay architecture, a resource-sharing infrastructure for mobile wireless devices. Friend Relay provides automatic publishing, discovery, and configuration of resources, as well as resource usage monitoring and control as chief contributions. Our goal has been to introduce mechanisms to monitor and control the utilization of shared resources. We have developed a prototype of Friend Relay. In addition, we have developed a prototype of an Internet sharing resource. We have evaluated the benefits of Friend Relay monitoring and control functionality for this service. The evaluation results, obtained by sharing Internet access with and without Friend Relay monitoring and control functionality, show that mobile devices either sharing or using resources can benefit from the control offered by Friend Relay. Through better control of the limited resources on a mobile device, we hope to encourage resource sharing and the realization of new network services.

8. REFERENCES

[1] Egevang, K., and Francis, P. The IP Network Address Translator (NAT), Internet Engineering Task Force (IETF) Request for Comments (RFC) 1631 (1994)

[2] Lee, C., and Helal, S. Protocols for Service Discovery in Dynamic and Mobile Networks. International Journal of Computer Research, 11 (2002) 1-12

[3] Ma, L. Develop P2P Applications with Device Discovery Technologies. IBM Developer Works (2005)

[4] Zhu, F., Mutka, M., and Ni, L. Service Discovery in Pervasive Computing Environments. IEEE Pervasive Computing 4 (2005) 81-90

[5] Srisuresh, P., and Sivakumar, S. NAT Behavioral Requirements for ICMP protocol. Internet Engineering Task Force (IETF) Internet Draft: draft-ietf-behave-nat-icmp-00.txt (work in progress) (2006)

[6] Audet, F., and Jennings, C. NAT Behavioral Requirements for Unicast UDP. Internet Engineering Task Force (IETF) Internet Draft: draft-ietf-behave-nat-udp-07.txt. (work in progress) (2006)

[7] Senie, D. Network Address Translator (NAT)-Friendly: Application Design Guidelines. Internet Engineering Task Force (IETF) Request for Comments (RFC) 3235 (2002)

[8] Guha, S., Biswas, K., Ford, B., Francis, P., Sivakumar, S., and Srisuresh, P. NAT Behavioral Requirements for Unicast TCP. Internet Engineering Task Force (IETF) Internet Draft: draft-ietf-behave-tcp-01.txt (work in progress) (2006)

[9] Bonjour. Apple, Inc. (2005)

[10] Rosu, M., Olsen, M., Narayanaswami, C., and Luo, L. PAWP: A Power Aware Web Proxy for Wireless LAN Clients. Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications, English Lake District, UK (2004) 206-215

[11] McKnight, L.W., and Howison, J. Towards a Sharing Protocol for Wireless Grids. Proceedings of the International Conference on Computer Communication and Control Technologies, Orlando, FL (2003) 24-31

[12] Ding, G., and Bhargava, B. Peer-to-peer File-sharing over Mobile Ad hoc Networks. Proceedings of the IEEE Annual Conference on Pervasive Computing and Communications Workshops, Orlando, FL (2004) 104-108

[13] Cheshire, S., Aboba, B., and Guttman, E. Dynamic Configuration of IPv4 Link-Local Addresses. Internet Engineering Task Force (IETF) Request for Comments (RFC) 3927 (2005)

[14] Rose, M. The Blocks Extensible Exchange Protocol Core. Internet Engineering Task Force (IETF) Request for Comments (RFC) 3080 (2001)

[15] Anceaume, E., Gradinariu, M., and Ravoaja, A. Incentives for P2P Fair Resource Sharing. Proceedings of the IEEE International Conference on Peer-to-Peer Computing, Konstanz, Germany (2005) 253-260

[16] Ngan, T.-W. J., Wallach, D. S., and Druschel, P. Enforcing Fair Sharing of Peer-to-peer Resources. Proceedings of the International Workshop on Peer-to-Peer Systems, Berkeley, CA (2003)

[17] Cohen, B. Incentives build robustness in BitTorrent. Proceedings of the Workshop on the Economics of Peer-to-Peer Systems, Berkeley, CA (2003)

[18] Adar E., and Huberman, B. A. Free Riding on Gnutella. First Monday (2000)