System for 802.11 connectivity at high speed

Nikolaos Frangiadakis, Danila Kuklov, and
Nick Roussopoulos

Department of Computer Science, University of Maryland,
College Park, Maryland, USA.
{ntg,dkuklov,nick}@cs.umd.edu

Abstract. Measurements and ongoing research have shown that WLAN connection for moving vehicles is feasible. However none of the previous work suggests a solution addressing a complete array of the challenges in vehicular WLAN communications. To amend this we designed PEGASUS, a system that provides wireless connection roaming at high velocities transparent to user level applications, and does not impose additional requirements to existing infrastructures. PEGASUS offers simple deployment, security, and scalability. It remains efficient under intermittent connectivity conditions and supports heterogeneous network mediums for increased robustness.

1 Introduction

Wireless access technologies are widely deployed in today's world, and they are a primary means in providing Internet connectivity to mobile users. Connectivity can be improved from access to multiple mediums such as WiFi and cellular. In this paper we focus on utilizing 802.11 networks as the principal communication technology.

WiFi networks are usually operated by private parties where wireless routers are self-contained limited-range segments. They offer high bit rates in comparison to mediums with longer reach and they are relative inexpensive to operate. The number of 802.11 networks has grown significantly in the last 5 years. According to recent studies [5] the number of home-deployed wireless routers in the US exceeds 15 million and rising. Such statistics suggest that many of these networks may overlap and allow mobile users to remain in range of some WLAN for continuous periods of time. The challenges of using them from a moving vehicle emerge largely due to their independent nature and short range. Movement from area covered by one access point to an area covered by another access point often requires a user to acquire a new IP address, and reconstruct all of the connections that were broken because of the IP change. In addition, each WLAN usually operates with its own private subnet and NATs the

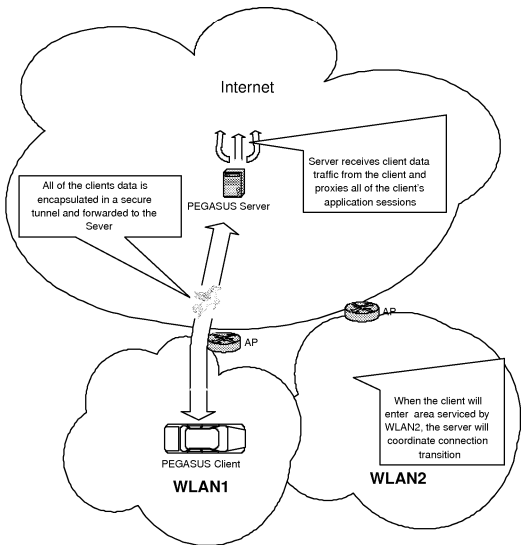

Fig. 1. PEGASUS - High Level Overview

internal network to the outside world. As a consequence, users have to adapt to this behavior, and many applications cannot handle breaks in connectivity.

The problems grow in magnitude and complexity when we talk about mobile users that travel at higher velocities (i.e. by car). The average connection to a single WLAN for such client is only 6 15 seconds. Moreover, due to time spent for DHCP and other conventional connection setup procedures the precious connectivity time is mostly wasted. Therefore, today, rapidly moving users cannot use WiFi and have to rely on other expensive and bandwidth-limited wireless access such as cellular.

PEGASUS:

To address the difficulties described above we have designed PEGASUS. PEGASUS is a system built to deal with rapid 802.11 access point connection switches; however it can use cellular or other long range mediums when WiFi is not available in the area. We focus on utilizing higher bandwidth connections and we abstract the underlying network management specifics. PEGASUS transparently switches between 802.11 access points, or even different mediums (WiFi or Cellular) presenting a constant IP and persistent connectivity appearance to users. To support transparency as clients move from one WLAN to another we use Pegasus

Server (PegSvc) to manage all connections. In addition, PegSvc hosts a connection database to aid clients as they move.

Our system's efficiency is based on re-using connection knowledge among many clients and pre-fetching connection candidates on the client path to minimize connection setup overheads. The overall PEGASUS architecture is designed to support very large networks with built-in security to protect both - system clients and network operators.

This research was inspired by ongoing work in the area of wireless connectivity for moving vehicles. Projects such as Drive-Thru Internet [11] have illustrated the feasibility of connecting to a WLAN on high speeds and effective use of its bandwidth. The CarTel [5] project illustrated an approach that maps numerous WLANs on the client route, and uses that information for future client connections. Still, although both of these projects offer a valuable insight in the vehicle WLAN connectivity, and they both share our view of reusing existing network protocols without requiring clients and applications to move to other transport layer approaches such as mobile IP; We strongly feel that PEGASUS offers a comprehensive solution to wireless connectivity through the existing infrastructure.

The above mentioned projects treat WLAN networks as separate domains and concentrate on solutions that deal with changing IP addresses and intermittent connectivity local to the client. In addition, both of them deal exclusively with the wireless network mediums. These works recognized a need to minimize the connection setup period when the client roams. The client software caches DHCP leases and reuses them when the client is back in range of a previously known access point. In PEGASUS, however, to achieve efficiency we propose to reuse a DHCP cache globally. PEGASUS caches all of the DCHP connections from all of the clients in a global cache, and continuously reuses them. Since DHCP is bound to a client MAC address, so to achieve global DHCP reuse clients change their MAC address to a value handed-in by the PegSvc. Once a client moves on to the next connection, it changes the MAC address again. This concept of recycling acquired DHCP connections and using a different DHCP identity at each independent access point island is the core concept that allows PEGASUS to achieve its efficiency and scalability.

The DHCP cache on PegSvc is built dynamically by clients as they discover new access points and connect to them. Once a client creates a new connection it will be available to other clients which will use it when they pass thru that area. PegSvc builds up a global database of access point layout and DHCP connection and treats these as common infras-

tructure resources that are reused by many clients. Rapid switch between cached DHCP connections is achieved by reducing connection setup time. The larger the global cache, the higher the PEGASUS service effectiveness. To speed up the cache built up, Access Point owners can participate by creating connections to their access points using a PEGASUS utility. In this case APs can be secure with WEP and still, clients will be able to connect to them when in range. To the best of our knowledge PEGASUS is the first system to allow usage of secured access points for vehicular communication.

To coordinate client and manager applications PEGASUS employs a management protocol to maximize connection time utilization for useful data transfer and to switch to the next access point on the path before the connection deterioration. Figure 1 presents a high level overview of PEGASUS. The mobile client in the automobile is connected to WLAN1 network. Client applications use the wireless connection for Internet, and all of the application sessions are encapsulated in a tunnel and sent to the PEGASUS server. Before the vehicle leaves the area serviced WLAN1, the server will send next connection information, coordinating the client's switch to WLAN2.

We assume that every WLAN is independently managed, so we deal with different ISPs, private address spaces and NATs. As depicted on Figure 1, to handle such heterogeneity, client's traffic is tunneled to the PEGASUS server. The server can be operated by a third party and acts as a multiplex point for all client Internet communications. PegSvc attempts to predict the client movement through deployed WLANs and offers choices for the next access point connection. The switch from one AP to another will not sever the ongoing client application sessions. Moreover, since the server acts as fixed peer to the non-mobile connection endpoints, it buffers network packets to smooth possible connectivity dead spots. All of the tunneled traffic is encrypted to offer extra security for the client data.

In summary in PEGASUS strives to maintain a seamless, high throughput TCP connection during handovers. For efficiency we reuse a global DHCP connection cache among all clients and attempt to predict connection candidates on the client's path. The PEGASUS connection switch is transparent to client applications, and does not impose modifications neither to the infrastructure of deployed networks nor the Internet Protocol stack. We allow participation of secured access point and we offer client and network operator security with authentication and encryption ser-

vices. Finally, PEGASUS is not limited to WiFi and will use any wireless medium to sustain client connectivity.

The rest of this paper is structured as follows: Section 2 classifies our approach with respect to existing work. Section 3 describes PEGASUS, and explains the reasoning behind our approach. Section 4 presents measurements and results from the study with prototype implementations, and Section 5 concludes this work and presents future research directions.

2 Related Work

The performance of TCP and UDP in wireless network scenarios from immobile clients has been relatively well-studied [1]. However, not many research efforts attempted to characterize WLAN performance for moving vehicles. The Drive-thru Internet project by Ott and Kutscher [12] studied the behavior of network connections over 802.11b and 802.11g from a moving car. The study involved a number of measurements over both UDP and TCP, and the goal was to understand the impact of the car's velocity, transmission rate, bit-rate, and packet size on throughput and delay. Ott and Kutscher classified WLAN connection period as three stages: the "entry" stage, "production" stage, and "exit" stage. During the entry and exit stages, the vehicle is far from the Access Point and throughput is low. However, when the distance is 200 meters from the Access Point, the connection is considered to be in the "production" stage. It is in that stage when the significant volume of data can be transferred. Drive-thru project shares our position to use intermediate proxies to further improve connection performance. In their more recent work [13], they show that they can avoid TCP start up overheads by using proxies, and hiding short period of disconnection from the transport layer. In PEGASUS instead of concentrating on modification of the usual TCP behavior, we concentrate on providing a constant connectivity appearance to the client, without the need to deal with re-initialization of the broken TCP connections. We do this by avoiding DHCP discovery costs during WLAN connection acquisition for fast and efficient connection transitions, and by providing a layer on top of the physical network cards to offer a persistent IP address to client applications.

Another study that demonstrated the feasibility of using off-the-shelf 802.11b wireless connectivity from a moving car was performed by Gass et al [8]. The experiments where conduced in a controlled environment and they measured performance from a mobile client to a single access point in the California desert. The authors measured the connection quality

between the client and the AP, and they concluded that packet losses are low within 150 meters of the access point for a wide speed range (5-75 mph).

While the two studies above demonstrate the possibility of using a wireless network from a moving car, more projects were carried out to study IP communications on the road. The FleetNet [16] project investigates inter-vehicle communication in wireless add hoc network, for traffic-related control information using addressing geo-based routing. Similarly, a Hocman [7] project also addresses data sharing across vehicles. An important work to access an internet via already deployed and open wireless 802.11b/g is conducted by MIT CarTel project [5]. The MIT group preformed a study on the availability of the open urban WiFi networks, and they attempted to estimate the performance of using "in situ" access points. The experiment involved several cars that were driven in the Boston and Seattle metropolitan areas. Their results state an average connection time of 13 seconds to a single access point while driving, and their biggest challenge was to cut down connection setup times when the car exited one network and entered another one. In PEGASUS we concentrate not only on performance of a single client, but propose a complete system to support vehicular WiFi network connectivity. We look at all of the clients managed by a server as infrastructure with common resources and knowledge about access points. Furthermore, PEGASUS's global DCHP cache repository significantly improves connection switch efficiency and scalability.

Other numerous research activities worked on solutions to mitigate disruptive effects of handovers which cause intermittent connectivity in the mobile communication environment. Many of them suggest modifications in the transport protocol layer. I-TCP [2] is a split connection approach that introduces a transport layer intermediary for splitting a TCP connection between a fixed and a mobile host into two connections. The idea is to isolate the fixed host from communication anomalies of the mobile host. I-TCP explicitly breaks the end-to-end semantics of TCP, i.e. TCP connections are terminated at the intermediary. In case of a hand-over, a state transfer from one I-TCP to another has to occur. The Snoop protocol [3] provides a more transparent support, and relies on a dedicated agent that on the path between the mobile and fixed station that "snoops" on the TCP communication, and might buffer some TCP segments and offer some retransmission services. In case of a handover a state transfer is not necessary required. In our approach we choose not to modify the underlying TCP layer to enhance the TCP performance,

and rely on currently deployed infrastructures and protocols to use the "in situ" access points.

The projects described above limit themselves to using WiFi for all of the network communication in their proposals. Other systems like CAMA [13] and Mobile Router [12] explored using multiple wireless mediums. CAMA utilized cellular communications for control messaging purposes, while Mobile Router concentrates on allowing different client types (blue tooth, cellular, etc...) to connect to a common router on a commuter bus. The router will search for multiple available network types in the area for an outside connection as well. They however do not attempt to examine the mobility of the outbound connection issues, and concentrate on efficient ways to service the internal router network on the bus. In PEGASUS we allow every client to use multiple physical mediums if the client is capable of doing so, and our clients can switch between mediums in a transparent manner to the user applications.

3 Architecture

The main objective of PEGASUS is to provide a solution that will present client applications with an appearance of a consistent connection, optimize utilization of individual connection, and minimize the connection transfer overheads. In addition we want to be able to support a large client base, offer communication security, and create a system that is easy to deploy. In this section we present architecture for PEGASUS. First, we outline our assumptions about the underlying infrastructure available today. Next, we discuss the overall system architecture and introduce individual components and their responsibilities. Finally, we present the control protocol messaging interface and discuss the applicability of our approach.

3.1 Assumptions

- Availability of WLANs - with continuing deployment of wireless access point in the US households, and in accordance with reports from pervious research projects, we assume that our clients will travel in a more or less connected grid of WLAN connection spots, and they will be able to find an available WLAN network most of the time. Since PEGASUS operates on either open and secure APs with the consent of the access point owner, the assumption is realistic. The non-connectivity periods should be relatively brief, and PEGASUS can to fall back to non WiFi wireless network if need arises.

- Length of single WiFi connection - in 802.11b/g networks connectivity can vary from slow 100% of the connections, and access point range can span from 200m to 1000m or more. For 25% to 40% of the time as the client passes the "production zone" (area closer to the access point), the client will experience good connection quality. Once the client is ready to exit the "production zone," we would like to switch to the adjacent network for the next "production zone". Each access point connection can last from 5 seconds to almost a minute at various driving speeds [11]. To accommodate frequent switches (every 15 to 20 seconds) we need to minimize connection setup overheads and avoid DHCP discoveries which can take up to 7 seconds each.
- No change to the underlying "in-situ" infrastructure - each WLAN is operated by a different provider, thus we have to accommodate switching to different IP addresses and private NAT domains, as well as using different security credentials for each access point. For example, each wireless access point today may use its own channel, SSID, and WEP key in secure networks. To have a realistic solution PEGASUS needs to use "in situ" infrastructure and avoid imposing additional hardware or network protocol requirements. Therefore, using something like Mobile IP [16] or I-TCP is not possible.
- Utilization of multiple wireless mediums - mobile devices today, often have more then one type of a wireless interface. To deal with occasional intermittent connectivity of the mobile client when 802.11b/g wireless connection will not be accessible, PEGASUS will use other means to sustain connectivity.
- Network Security - - a complete solution needs means to protect user data, as well as ways to prevent network abuse and user illegal activities.

3.2 Requirements

The above mentioned assumptions were compiled into the following requirements list for PEGASUS:

- Transparent connectivity appearance to client applications (i.e Web, email access, file transfer, etc. . .)
- Deployment on top of "in-situ" access points, without managing or changing the existing infrastructure, and support for any WLAN configurations
- Simple installation on clients and easy server deployment, without modifications to the existing operating systems and applications

- Support of the existing user equipment without requiring any custom or specialized hardware at the client or server devices
- Utilization of multiple network mediums available at the client
- Extensibility for future performance enhancements to further improve mobile connectivity
- Simple system deployment and dynamic growth of the system connection database
- Scale to support a growing number of clients
- Security for clients and network operators

3.3 System Architecture

In order to provide seamless connectivity in a mobile environment, and employ "in situ" network infrastructure, PEGASUS uses a service above the transport layer for connectivity management, and masks the physical connection transitions by offering a virtual network interface with a constant IP address to the client applications. The primary idea of PEGASUS is to split the end-to-end connection to conceal the client IP address changes from the applications on the mobile end and fixed host services. The two main components that achieve the connection splitting are the client module that resides at the mobile node and the manager proxy that is located in the network. The client and the manager nodes communicate with each other via a control message protocol and hide connection transitions from the application layer sessions. Additionally, to survive the loss of connectivity for brief periods of time and still achieve persistent connectivity view, the manager and the client modules maintain connection states and offer session traffic buffering.

Figure 2 depicts an overview of our architecture. The client is composed of the following elements:

- Transparency Layer provides the user applications with an appearance of a constant IP address. The layer creates a virtual interface and modifies client routing to send all of the outgoing traffic via that interface
- Connection State Management Management resides just below the transparency layer and its primary function is optimization to handle volatile connection conditions of the mobile network. This layer offers client side buffering, and it can track outgoing TCP connections to keep them alive during the moments of intermittent connectivity. To supplement the current functionality, this layer can be extended to

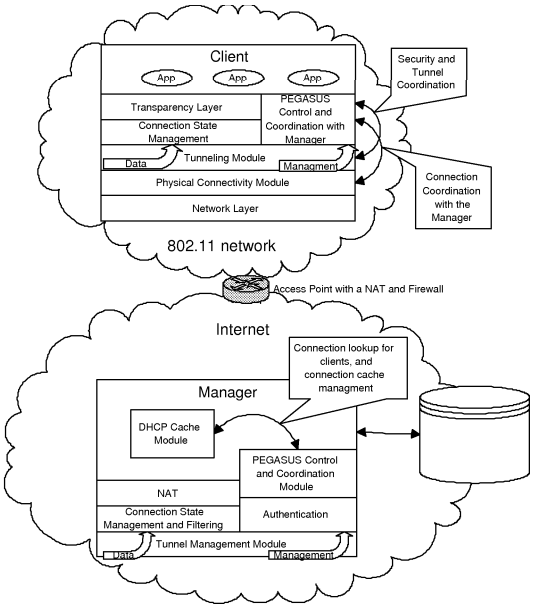

Fig. 2. System Architecture

notify applications that wish to have knowledge about actual current physical connectivity status.

- PEGASUS Control Module is responsible for communication with the manager to coordinate connection transfers and other manager supported services. This module tracks client movement, and keeps connection options received from the manager. In addition, this module offers an interface to authenticate with the server, and notify the server of new connections. All of the client-server control communication is handled by this module.
- Tunneling Module creates a UDP tunnel to the manager and forwards all of the traffic generated by the client applications and control module to the manger. This module supports open and encrypted tunnels. The tunnel parameters are negotiated during client authentication phase with the manager. Also, at this layer all of traffic received from the manager is classified as data and passed up to the Connection State Management module, or classified as management traffic and passed to the PEGASUS Control.
- Physical Connectivity Module is responsible for maintaining a physical connection at all times. This layer keeps track of the available connection mediums at the client, and monitors each medium for sig-

nal quality. The main focus in the current implementation is tracking of the 802.11 signal and detection of the signal deterioration. The module will ask PEGASUS control for a list of available connections which the manager has for client's location. The client scans for networks, and uses this list to select one with a good signal. In the cases, when the scan cannot match any of the found connections, the client will forward traffic over the secondary medium, and attempt to connect to 802.11 networks with DHCP. Once the connection is established, it is sent to the manager to add to the global cache. The purpose of using a connection from the manager list is to avoid DHCP discovery. The global cache contains connection information in the form of (MAC, IP, SSID, AuthInfo) tuples. When clients use these tuples they take the identity of an already configured entity in the WLAN. Once they move, the cached identity can be reused for other transit users. Such connection information recycling allows PEGASUS to avoid setup overheads, and this scheme guarantees that we will only use a limited number of resources protecting the wireless network owner and his access point from abuse.

The complementary part of the clients in PEGASUS system is the manager proxy that multiplexes all of the client connections and stores them in a global DHCP cache to be recycled. The manager proxy server consists of the following elements:

- DHCP Cache Module stores all of the known connections created by PEGASUS clients. The cache expires old and stale information and updates the renewed connections. Along with the DHCP data the module keeps the connection locations. The location information is used to respond to client requests with access points on the client's path. The cache is dynamically populated as client nodes discover new access points, and by access point owners that want to participate in PEGASUS and create connections to their network. Every connection in the database can have set of filter rules to restrict client internet access. We do not promote such restrictions, but it allows access point owners to have more control over their network.
- PEGASUS Control Module responds to all incoming requests from the clients. The Control Module authenticates clients to use PEGASUS, and it responds to client connection requests with entries from DHCP Cache. In addition, the Control Module tracks client movement and connection usage, and updates client NAT entries to correctly route traffic when connection switches.

- NAT is a network address translation scheme used for connection splitting in the system. As the client moves from one connection to another, PEGASUS hides client mobility by NATing all of the client's connections. The client end points in the NAT are constantly updated to route client data to the correct connection.
- Connection State Management Module is the manager equivalent of the client Connection State Management Module. This piece is not finished; the full implementation would need to keep track of various protocols above IP to keep alive application sessions on the fixed host end, when the mobile nodes experience intermittent connectivity. The main purpose of Connection State Management module pair is to improve connection robustness in volatile mobile environments. The client side deals with the mobile end, and the server side handles the fixed host session end.
- Tunnel Management Module unpacks data and management traffic from the client tunnels. The data traffic is forwarded to Connection State Management and NAT, while management traffic is forwarded to the PEGASUS Control. The tunnel security parameters are negotiated during client authentication.

3.4 Control Protocol Messaging

Our architecture requires client and manager proxy to maintain a persistent relationship for managing wireless connection transfers during client movement from an area serviced by one access point to the area serviced by the next access point. To achieve this we have developed a control protocol for PEGASUS clients and managers

Now, we present a quick overview messages that we support:

- authenticate - client sends this message when it connects to the PEGASUS to authenticate itself with the manager, and to negotiate tunnel encryption settings.
- connection_list_request - client sends this message to request a list of connections in its proximity. Manager will reply with a list of connections within 500 meters from the client. Furthermore, the client attempts to pre-fetch extra information to avoid delays when the connection deteriorates and a switch is desired.
- connection_in_use - client uses this message to notify server that uses this connection. The server can sometimes NACK, if the connection is already used by another client. When the connection is new, the client embeds the connection information in the message and server will add

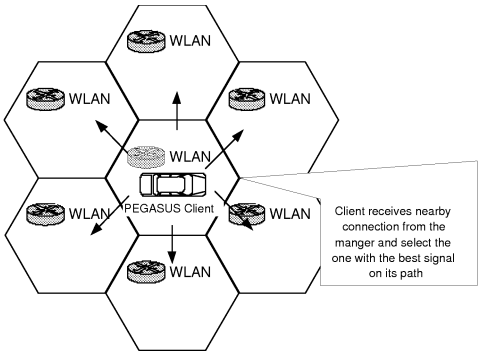

Fig. 3. Client Connection Switch Options

it to the cache. Also, if the server detects that the client moved on (by noticing a change in the client's tunnel end point), it will mark the connection for client's use without explicit "connection_in_use" message.

- connection_add - client uses this message to add a new connection to DHCP cache without actually using the connection for communication.
- ack/nack - used by the manager to allow/disallow client connection use.

This is the list of messages that PEGASUS currently supports; in the future, the protocol can be extended to support additional services and requirements.

To demonstrate control protocol usage Figure 7 depicts a simple use case. When a client needs a connection, it sends a "connection_request" to the manager, and receives a response with a list of connections in the proximity. With this information the mobile node can select a connection with the best signal, and it can predict the next one or two connections along its movement path. Once the client decides on the next connection, it sends a "connection_in_use" message, which the manager, can "ack" or "nack" depending on availability of that connection. In most scenarios, the manager will acknowledge the connection, and update the UDP tunnel and NAT mappings to route to a new client address. When client approaches the edge of the connectivity area, it will send another "connection_request" and transition to the next connection.

In cases when the client does not receive a connection from the manager it will try to find an open network and connect. Once connected, it will notify the manger about the new connection with "connection_in_use".

3.5 Applicability

The described above components comprise our approach. The client transparency layer achieves application connection transparency. The physical connection layer attempts to provide network connectivity at all times. The Tunneling layers at the client and server deal with private networks, NAT and firewalls at 802.11 access points. The tunnels are simple to establish and allow client traffic to remain transparent to the internal WLAN settings. Also, the tunneling provides the connection splitting mechanism between the rapidly moving client, and the fixed endpoints. Since all of the client connections are NATed at the manager - client mobility is hidden. The efficiency of the connection switching comes from the global DHCP cache, and pre-fetching of the connections on the client's path. Finally, PEGASUS connection management layers provide mechanisms to deal with brief periods of intermittent connectivity, and the system provides security with authentication and tunnel encryption.

The last system requirements that we stated was ease of deployment and scalability. At the client the required modification is a single executable module to abstract the physical connection. The manager proxies also run a module that inspects incoming traffic and runs NAT. PEGASUS proxies can be scaled by increasing a number of server machines and splitting the connection database among them by geographic regions. The proxies do not require any centralized communication or synchronization aside from client authentication services. As the number of clients in the system increase, one can install more managers and keep partitioning connection database on the basis of connection location. The overall infrastructure is very light and does not impose any additional rules on the deployed networks, and we hope current technology trends continue to introduce more mobile devices with capabilities to connect to multiple wireless mediums, making them potential client devices in PEGASUS. [12]

4 Measurements

PEGASUS is implemented on top of Ubuntu Linux distribution. To implement various routing and networking functionality on client and server

we took "Click Modular Router" project [1] and extended it. In addition on the client, we have incorporated Wireless Extensions for Linux [17]. Thus PEGASUS will work with any 802.11 card supported by Linux.

To measure PEGASUS performance we have simulated a mobile environment in our lab. PEGASUS server is a Pentium III with Ubuntu Linux deployed in public domain. For "in situ" WLANs, we installed Linksys 54g access points that are available in any store configured with default factory settings. Every access point runs a firewall, NAT, and DHCP for its private network. Several of the APs are secure with WEP and we imported their connection information into PEGASUS DHCP cache manually. For the open access points the DHCP cache on the server is populated dynamically by clients that connect to every WLAN and send the DHCP connections to server. The client used for measurements is a regular laptop running Linux with ipw3945 Intel wireless card which is a standard for Dell laptops. To simulate movement, the client switches its wireless connection from one access point to the next and the connectivity period to each WLAN depends on the simulated driving velocity.

To benchmark performance we use TTCP tests and a web browsing session. The TTCP application runs unaware of the ongoing physical connection transitions and measures end-to-end TCP bandwidth. In order to emulate different application behaviors we test with several TTCP configurations; we simulate large continuous data transfers, and multiple smaller data transfers. For the web browsing sessions we record response times and the number of times a web page comes back with status "400 Page Not Found".

To evaluate PEGASUS we developed several scenarios. First, we run our test suite without PEGASUS. The client uses a direct WiFi connection with no proxy involved to record a baseline performance metrics. Then, we run the tests with PegSvc proxing but without any connection transitions, the client remains connected to the same access point for the duration of the tests. Finally, we simulate several driving scenarios for velocities from 20 - 100 km/h (12 - 65 mph). First, we measure client performance when the client always has to request a fresh DHCP from every access point during connection transition. This is the worst case scenario for PEGASUS, since the client does not use global DHCP cache. Second, we measure performance when the client acquires DHCP from access points 50% of the time while the other 50% of the time it uses a connection from PEGASUS DHCP cache. Finally, we evaluate performance of PEGASUS when the client uses a connection from PEGASUS

DHCP cache for every transition, and does not need to do any DHCP requests.

For the simulations, we have assumed that a WLAN range is 250 meters in diameter, and our DHCP renewal process takes 3 seconds. Previous studies reported WLAN ranges of up to 500 meters in radius, and a conventional Linux DHCP usually can take up to 7 seconds.

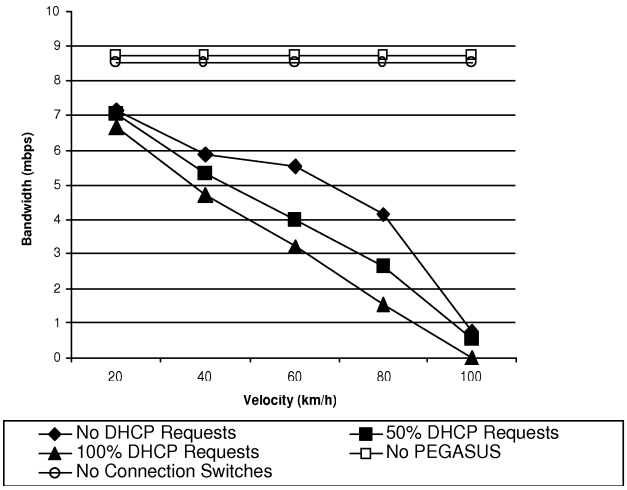

Fig. 4. Client TCP performance for continuous transfers

The results for continuous TCP transfers illustrated that the connections splitting is not very heavy in overhead. In our lab, clients we able to achieve bandwidth of 8.7 mbs when they did not use PegSvc to proxy their connections, and we measured bandwidth of 8.5 mbs when client connections when through PegSvc. For driving simulations, the chart on figures 4 and 6 demonstrates the benefits of DHCP cache faster connection transitions. At low velocities the transitions are rare, and the difference in effective bandwidth is not very noticeable, but at higher velocities simulations with 50% DHCP and no DHCP clearly use the access points more efficiently. At 100 km/h tests that required DHCP 100% of the time never completed. The curve for scenarios without DHCP shows a gradual bandwidth decrease with a large dip, for velocities from 80 to 100 km/h. The connectivity period to individual to access point goes from 12 to less than 9 seconds for these cases, and since PEGASUS needs to

scan the network when the signal worsens, our scan time starts to be a larger overhead factor. However, the scan can be optimized with a more efficient implementation. Overall, PEGASUS performs very nicely, supporting bandwidth close to 750 kbs even at 100 km/h. At slower velocities, which are more common for urban driving, the bandwidth is a lot higher.

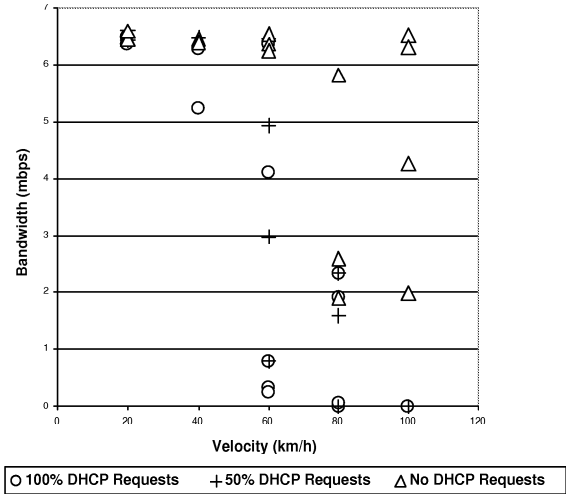

Fig. 5. Client TCP performance for short transfers

In experiments with shorter TCP transfers, we again show the connection stability that can be achieved with PEGASUS in mobile environment. In Figure 5 transfer rates for the shorter segments vary because the transfers are more susceptible to connection transitions. Some of the segments do not experience transitions at all. Nevertheless, PegSvc with no DHCP clearly illustrates the most stable behavior where all of the data is eventually delivered.

To compare the web browsing simulations we ran them in a continuous loop for a fixed period of time for every connectivity environment. In Figure 7 the mobility increases, our results show growing response times, and our client is able to issues less requests in the test time window. However, using PegSvc and DHCP cache obviously provides a significantly slower degradation rate. With PegSvc at 100 km/h we could issue half of the requests compared to the standing still client, and our response time

Speed	Bandwidth (Mbps)		
	No DHCP	50%DHCP	100%DHCP
20	7.1377	7.0204	6.6516
40	5.8826	5.3323	4.7181
60	5.5419	4.0092	3.1951
80	4.1477	2.6153	1.5322
100	0.7623	0.5428	0

Fig. 6. Client TCP performance for continuous transfers - Table

Experiment	Number of Requests	Average Response Time
No Pegasus	992	79.913
No Switching	808	157.623
No DHCP req (20 km/h)	853	124.34
50% DHCP req (20 km/h)	823	169.788
100% DHCP req (20 km/h)	697	178.34
No DHCP req (40 km/h)	783	141.589
50% DHCP req (40 km/h)	709	173.452
100% DHCP req (40 km/h)	660	192.772
No DHCP req (60 km/h)	851	146.537
50% DHCP req (60 km/h)	487	236.342
100% DHCP req (60 km/h)	435	313.641
No DHCP req (80 km/h)	564	246.221
50% DHCP req (80 km/h)	340	383.855
100% DHCP req (80 km/h)	216	478.855
No DHCP req (100 km/h)	442	267.852
50% DHCP req (100 km/h)	238	336.175
100% DHCP req (100 km/h)	171	410.579

Fig. 7. Client Web browsing performance

grew by a factor 2 as well. Without PegSvc, the number of requests went down by a factor of 6, and response time grew 300%.

5 Conclusion

PEGASUS is a system to enable wireless connectivity for fast moving vehicles. It provides clients with a constant IP address to preserve application session connectivity, while transparently changes physical connections to sustain steady connectivity. Efficient connection switching is achieved by storing a global DHCP connection cache service PegSvc on PEGASUS server and predicting connection candidates on the client's path. A salient feature of the PEGASUS system is that it does not impose modifications to the infrastructure of deployed networks or protocols,

thus making its deployment attainable. Using in-situ infrastructure and inexpensive dynamic population of the cache helps bootstrapping the service at very low cost and PEGASUS is built to scale with the increasing number of clients.

For WiFi connectivity PEGASUS can use either open access points or WEP enabled access points with the owner's consent for secured connections. Since all the communication is piped through an encrypted tunnel, PEGASUS offers clients and network operators security with authentication and encryption services.

We have implemented the system for deriving performance parameters which were plugged into an extensive simulation. Our experiments showed solid transfer rates and continuous connectivity for high velocity client simulations. The DHCP cache proved to sustain client connection transitions when the conventional connection renewal schemes degraded beyond workable conditions. We were able to achieve usable and stable network with speeds of up to 100 km/h.

Currently we are experimenting with PEGASUS buffering and connection state management mechanisms to improve handling intermittent connectivity. In addition we are incorporating multiple network medium usage into future experiments.

References

[1] Daniel Aguayo, John Bicket, Sanjit Biswas, Glenn Judd, and Robert Morris. Link-level measurements from an 802.11b mesh network. In SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications, pages 121–132, New York, NY, USA, 2004. ACM Press.

[2] Ajay Bakre and B. R. Badrinath. I-tcp: indirect tcp for mobile hosts. In Proceedings - International Conference on Distributed Computing Systems, page 136, Vancouver, Can, 1995. IEEE, Piscataway, NJ, USA. Compilation and indexing terms, Copyright 2006 Elsevier Inc.

[3] Hari Balakrishnan, Srinivasan Seshan, Elan Amir, and Randy H. Katz. Improving tcp/ip performance over wireless networks. Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM, pages 2–11, 1995.

[4] Bharat Bhargava, Xiaoxin Wu, Yi Lu, and Weichao Wang. Integrating heterogeneous wireless technologies: a cellular aided mobile ad hoc network (cama). Mob. Netw. Appl., 9(4):393–408, 2004.

[5] Vladimir Bychkovsky, Bret Hull, Allen Miu, Hari Balakrishnan, and Samuel Madden. A measurement study of vehicular internet access using in situ wi-fi networks. In Gerla et al. [9], pages 50–61.

[6] R. Chakravorty, P. Vidales, L. Patanapongpibul, K. Subramanian, I. Pratt, and J. Crowcroft. Performance issues with vertical handovers – experiences from gprs cellular and wlan hot-spots integration, 2004.

[7] M. Esbjrnsson, O. Juhlin, and M. stergren. The hocman prototype: Fast motor bikers and ad hoc networks, 2002.

[8] Richard Gass, James Scott, and Christophe Diot. Measurements of in-motion 802.11 networking. In WMCSA '06: Proceedings of the Seventh IEEE Workshop on Mobile Computing Systems & Applications, pages 69–74, Washington, DC, USA, 2006. IEEE Computer Society.

[9] Mario Gerla, Chiara Petrioli, and Ramachandran Ramjee, editors. Proceedings of the 12th Annual International Conference on Mobile Computing and Networking, MOBICOM 2006, Los Angeles, CA, USA, September 23-29, 2006. ACM, 2006.

[10] Robert Morris, Eddie Kohler, John Jannotti, and M. Frans Kaashoek. The click modular router. In SOSP '99: Proceedings of the seventeenth ACM symposium on Operating systems principles, pages 217–231, New York, NY, USA, 1999. ACM Press.

[11] J. Ott and D. Kutscher. The "drive-thru" architecture: Wlan-based internet access on the road. In Vehicular Technology Conference, 2004. VTC 2004-Spring. 2004 IEEE 59th, pages 2615–2622, New York, NY, USA, 2004. ACM Press.

[12] J. Ott and D. Kutscher. Drive-thru internet: Ieee 802.11b for "automobile" users. In INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies, volume 1, 2004.

[13] Jorg Ott and Dirk Kutscher. A disconnection-tolerant transport for drive-thru internet environments. In INFOCOM 2005. Twenty-fourth AnnualJoint Conference of the IEEE Computer and Communications Societies, pages 1849–1862.

[14] Pablo Rodriguez, Rajiv Chakravorty, Julian Chesterfield, Ian Pratt, and Suman Banerjee. Mar: a commuter router infrastructure for the mobile internet. In MobiSys '04: Proceedings of the 2nd international conference on Mobile systems, applications, and services, pages 217–230, New York, NY, USA, 2004. ACM Press.

[15] Jean Tourrilhes. Wireless extensions for linux. Wireless Tools for Linux.

[16] URL. Homepage of fleetnet project. http://www.fleetnet.de/.