# PEGASUS – Architecture for Mobility using in situ 802.11 deployments at high velocities

**Dan Kuklov**
University Of Maryland
Department of Computer Science
danilkuk@wam.umd.edu

## Abstract

This work presents a strategy and discusses an approach to enable WLAN based services for moving vehicles. Measurements and ongoing research have shown that WLAN connection for moving vehicles is feasible. The paper concentrates on strategies for preserving a continuous connectivity and seamless connection switching across multiple access points, without imposing any additional requirements on today's wireless routers, and without explicit management of third party hardware. Based on measurements and simulations, we have developed PEGASUS - architecture for WLAN connection switching at high velocities without impact on user level applications, and without additional management of the wireless access points. Our architecture utilizes an existing cellular network as a low bandwidth reliable control channel, and allows a moving client to connect to open WLANs on the client's path. For interleaved continuous wireless networks PEGASUS provides a client with an appearance of a stable continuous connection. We also discuss scenarios of intermittent connectivity, as well as various operational matters such as access point discovery, scalability issues, and other network related concerns.

## I. Introduction

Wireless access technologies are widely deployed in today's world, and they are a primary means in providing Internet connectivity to mobile users. The two most common approaches to wireless network access are cellular phone networks and Wireless LANs (WLANs). Each of the network types has its strengths and weaknesses. Cellular access is usually operated by a mobile phone provider; they are expensive to build and maintain and offer a lower bit access, however, they provide a ubiquitous permanent access, which does not require users or applications to adapt to the mobile environments.

Wireless LANs on the other hand, are often operated by individuals. Every wireless router is self-contained limited-range network. The biggest advantage of the wireless LANs is much higher bit rates in comparison with the cellular networks. The WLANs are relatively inexpensive to operate, and their number has grown significantly in the last 5 years. According to recent studies [5] the number of home-deployed wireless networks in the US is over 15 million and growing. 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. Nevertheless, due to the independent nature of every WLAN, moving from area covered by one access point to 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 due to WLAN switch. In addition, each WLAN usually operates with its own private subnet and NATs the internal network to the outside world. As a consequence, users have to adapt to this behavior, and application often need to be adjusted to handle breaks in connectivity.

The problems grow in magnitude and complexity when we talk about mobile users that are traveling by at higher velocities (i.e. by car). The average connection to a single WLAN network for such client is only 6 – 15 seconds. Also, because of the a limited range of a WLAN, and time spent for DHCP and other conventional connection setup procedures the precious connectivity time is wasted. Therefore, currently, rapidly moving users can only rely on cellular access, which is expensive and bandwidth limited.

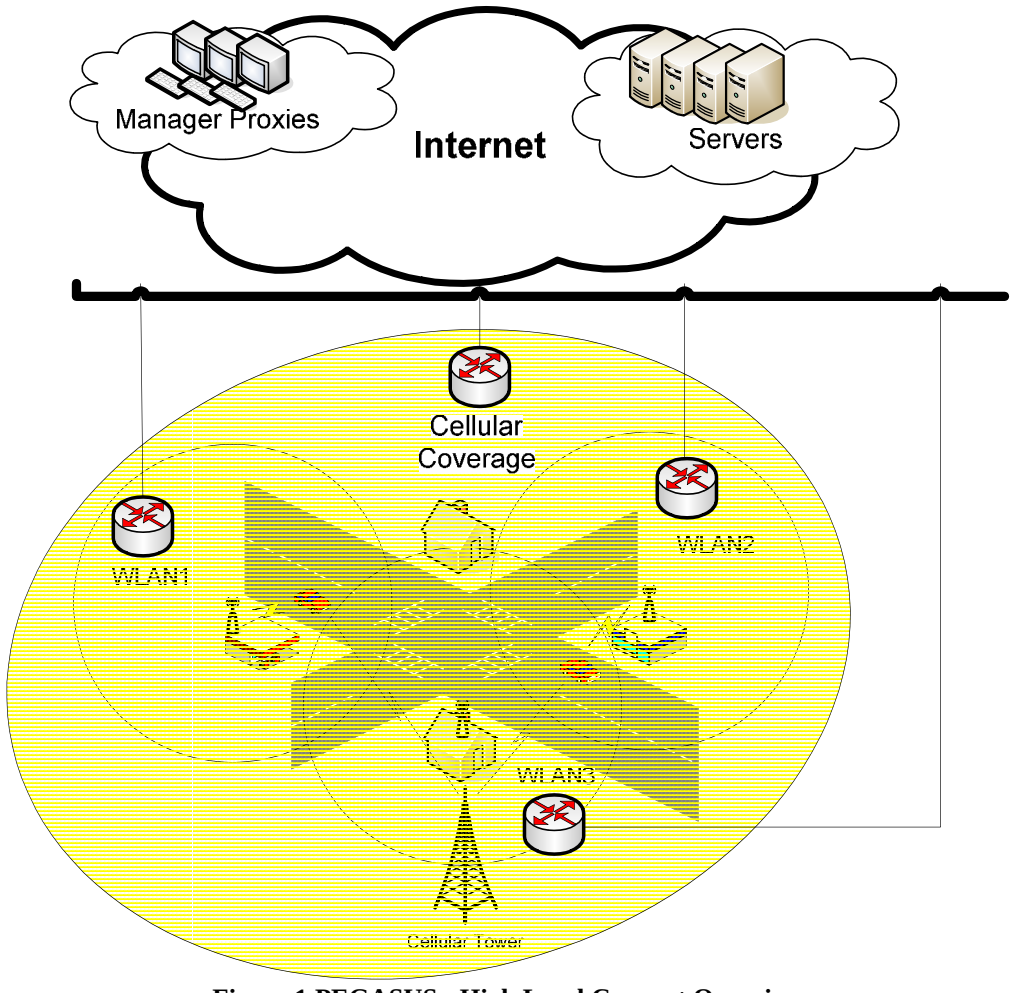


**Figure 1 PEGASUS - High Level Concept Overview**

To alleviate the challenges of using wireless networks in a moving vehicle we propose our own architecture – PEGASUS. PEGASUS is capable of abstracting the complexities of WLAN transitions without any impact to the client applications, optimize the connection acquisitions, and aid in the optimal network selection. Our research was inspired by ongoing work in the area of wireless connectivity for moving vehicles.

Projects such as Drive-Thru Internet [3] 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 a more comprehensive solution to tackle this problem.

The mentioned projects treat WLAN networks as separate domains and concentrate on solutions that deal with changing IP addresses and intermittent connectivity. In addition, both of them deal exclusively with the wireless network mediums. In contrast, PEGASUS concentrates on seamless WLAN network switching, and efficient selection of the optimal network in range. We feel that with our architecture we can maximize the "productive" connectivity periods, and minimize connection setup/teardown overheads. Furthermore, in order to achieve maximum efficiency we propose to use an existing cellular network as a control channel for coordination of the WLAN switching and access point discovery. The dependable control channel enables our mobile clients 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 a use case for PEGASUS. The mobile clients in the automobiles have connections to both - cellular and wireless networks. Client applications use the wireless connection for Internet, and all of the application sessions are routed through the manager proxy. As the vehicle leaves the area serviced by one WLAN, the manager will send next connection information via the cellular control channel, coordinating the client's switch to a new WLAN.

Now, every WLAN is independently managed, so we expect to deal with different ISPs, private address spaces and NATs. As depicted on Figure 1, to handle such heterogeneity we decided to use an intermediary (similar to FleetNet[7] and Drive-thru internet). Our intermediary (also referenced as manager or proxy) can be operated by a third party and acts as a multiplex point for all client Internet communications. The actual mobility management takes place in an application layer. PEGASUS clients use the cellular network and a control messaging protocol to coordinate WLAN switches. The manager 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 our proxy acts as fixed peer to the non-mobile connection endpoints, it buffers network packets, to smooth possible connectivity dead spots.

Since, PEGASUS deviates from the well-established end-to-end paradigm that most of the Internet Protocols are based on; this paper will justify the motivations for our approach, and explain our choices. The rest of this paper is structured as follows: Section II classifies our approach with respect to existing work. Section III describes PEGASUS, and explains the reasoning behind our approach. Section IV presents measurements and

results from the study with prototype implementations, and Section V concludes this work and presents future research directions.

## II. Related Work

The performance of TCP and UDP in wireless network scenarios from immobile clients has been relatively well-studied [6]. However, not many research efforts attempted to characterize WLAN performance from moving vehicles. The Drive-thru Internet project by Ott and Kutscher [2] 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 too 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 bin 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 [4], 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, and we also use our proxy to avoid wireless connection and DHCP discovery costs during WLAN connection acquisition.

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 and 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 [7] project investigates inter-vehicle communication in wireless add hoc network, for traffic-related control information using addressing geo-based routing. Similarly, a Hocman [9] 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 CarTel group shares our vision to use "in situ" open access points deployed in the residential areas to connect to an internet from a moving vehicle. The MIT group preformed an important study on the wide availability of the open urban Wi-Fi networks, and they attempted to estimate the performance of using "in situ" networks. The CarTel experiment involved several cars that were driven in the Boston and Seattle metropolitan areas. The group recorded their connectivity and data upload results. In contrast to that effort, in our work we concentrate not only on network performance measurements, but propose PEGASUS - a light weight architecture

to abstract the client applications from the roaming nature of the connectivity and to improve overall vehicular client experience.

On that note, although 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 [10] 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 [11] 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.

I-TCP and Snoop both attempt to present optimizations for handling short-term communication problems during connection handovers. Our approach differs from the mentioned techniques because we do not attempt to enhance the TCP performance or modify the underlying TCP implementation. In PEGASUS, we strive to maintain a seamless, high-throughput TCP connection during handovers between two base stations, by relying on availability of our control channel and the ability of our manager proxy to predict an optimal connection switch with minimal handover overhead costs. Our switch is without connection disturbance to the client applications, and does not impose modifications to the infrastructure of deployed networks or protocols.

# III. Architecture

Based on the insights from related works, we have developed a following architecture for PEGASUS presented in this section. 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 briefly discuss the applicability of our approach.

## *A. Requirements*

Our main objective is to provide a solution that will present client applications with an appearance of a consistent connection, optimize utilization of individual connection "production" zones, and minimize the connection transfer overheads. Due to the continuing deployment of wireless access point in the US households, and in accordance with reports from pervious research projects, we decided to 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. In case the connection is not available right away, the non-connectivity period should be relatively brief. Also, because our communication with the manager proxy will be conducted over the cellular

network, we will assume that each client has a cellular interface to send control protocol messages to the manager.

Now, the 802.11b/g connectivity can vary from slow to almost optimal conditions, and each network can span from 500m to 1000m or more, equivalent to 5 seconds to almost a minute periods of connectivity at various driving speeds[3]. For 25% to 40% of the period the client will the in the optimal or "production" zone. Once the client is ready to exit the "production" zone, we would like to switch to the adjacent network for the next "production" zone.

These assumptions indicate that for each individual connection, we need to minimize the connection overhead and avoid DHCP discoveries. Moreover, we cannot assume that every WLAN is operated by the same 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 and SSID. Finally, since our goal is to use "in situ" access points, our manager will maintain a WLAN map, thus, PEGASUS infrastructure needs to have built-in support for dynamic discovery of new access points.

All of these restrictions make the deployment of a solution that may impose specific hardware or network protocol requirements on the available wireless access points - impractical. Therefore, using something like Mobile IP [16] or I-TCP is not possible. In addition, any proposed architecture needs to deal with occasional intermittent connectivity of the mobile client when 802.11b/g wireless connection will not be accessible. To handle such connectivity dead spots some infrastructure built-in buffering capabilities are desired.

To summarize the above mentioned assumptions/requirements our architecture should support the following:

- the architecture should provide a persistent connectivity view to existing client application programs (i.e Web, email access, file transfer, etc…)
- the architecture should not be dependant on a specific WLAN configurations, and it must be applicable to different authentication technologies
- no changes to the existing operating systems and applications should be required; and the final solution must not require usage of specialized mobile devices but must support existing user equipment (laptops, inbuilt computer in cars, etc…)
- the architecture should allow inclusion of performance enhancing proxy elements to further improve mobile connectivity
- the architecture must provide means to dynamically discover and report new open access points, for expanding and updating the available connectivity map for the mobile clients

## *B. System Architecture*

To provide seamless connectivity in the very mobile environment, and employ the "in situ" network infrastructure, our approach 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 enhance the concept of connection splitting for the purpose of concealing the constant client IP address changes from the mobile client applications 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 over a reliable (cellular) network interface, to coordinate and mask the connection splitting from the application layer sessions. Additionally, to survive the loss of connectivity for brief periods of time and still achieve the persistent connectivity view, manager and the client modules maintain connection states and offer session traffic buffering.

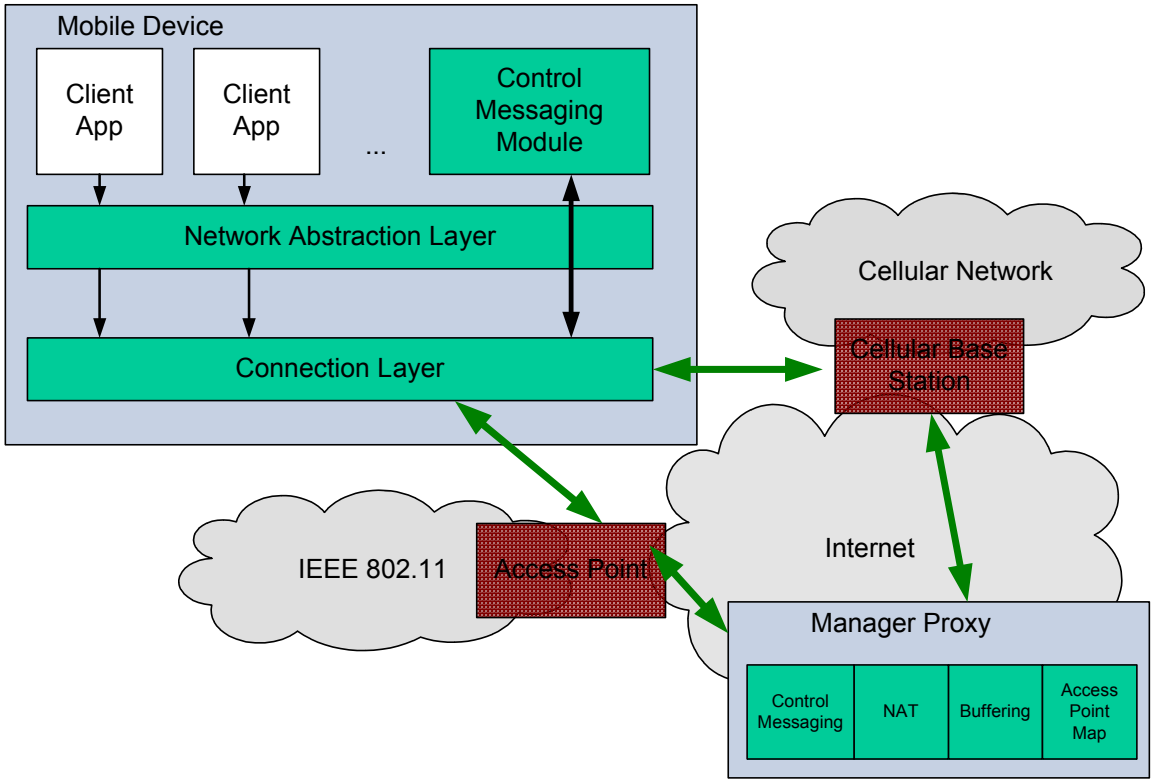


**Figure 2 PEGASUS System Architecture**

Figure 2 depicts an overview of our architecture composed of the following elements:

- **client control messaging module**, which is responsible for coordination of the connection transfers, and modification of the link layer connections during transitions. This layer can be extended to provide additional client – manager services to deal with anticipated connectivity losses, and to deliver information to the client applications that are aware of the mobile connectivity nature of the client.
- **client network abstraction module** provides a stable connection interface to the client applications. The stable connection interface is a virtual interface on the client, which is used by applications that require Internet. The network

abstraction layer can be augmented to carry out data buffering and request batching to take advantage of periods with the best connection rates, and mitigate the effects of possible intermittent connectivity.

- **the client connection layer** is responsible for a physical connection to the access point, and traffic forwarding to the manager proxy. For every connection transition, the control messaging module receives a list of possible connection candidates. Then the client connection layer selects the best choice, and switches to the new access point. To avoid DHCP discovery, the manager sends tuples of (MAC, IP, SSID, AuthInfo) to the client, allowing the connection layer to use the tuple data to take identity of an already known and configured entity in the WLAN. This means that at various times, distinct mobile clients will appear to an access point as the same node. Such scheme allows PEGASUS to avoid WLAN connection setup overheads by using pre-allocated tuples. It also guarantees that we will only use a limited number of resources protecting the wireless network owner and his access point from abuse by PEGASUS.

  Note, that in order to deal with the client's changing IP address and to handle various NAT configurations, the client connection layer will encapsulate and forward all of the Internet traffic within a UDP tunnel to the manager proxy. The proxy will perform the session connection splitting between the clients and fixed endpoints. Also, the UDP tunnel can be encrypted to provide further security and anonymity of the mobile client traffic within the wireless networks.
- **the manager proxy** is the counter-part of the mobile client module in the fixed network and conceals the mobile node volatile IP address and temporary unavailability from the corresponding (fixed) application peers. The manager contains a map of access points in each geographical area, and for every access point it retains a list of pre-configured connection tuples. The tuples can be populated by clients that participate in the discovery of open WLANs (the control messaging protocol section explains the discovery process in detail).

  Upon receiving a connection transfer request from the client, the manager replies with a list of possible connection tuples to APs near client's location. Then, the client will choose the best alternative from the list. Also, in order to conceal the client's mobility, the manager NATs all of the Internet traffic originated from the client and dynamically updates the NAT entries when client's IP changes. The NAT enables PEGASUS to persist all of the client's TCP and UDP sessions during the connection transitions, while the UDP tunneling of all of the data traffic between client and manager, avoids complications with the NAT configurations at the wireless access points. Keep in mind, that, since the client and the manager proxy send their control messages and connection state updates via reliable interface, the manager is always able to make an intelligent choice regarding the buffering of the client connection data, and will not attempt to forward any packets to a stale UDP tunnel.

The described above components comprise our approach. PEGASUS is successful in providing the client applications with a view of a consistent connection to the Internet, and we are able to perform very fast wireless connection transfers avoiding the usual DHCP and TCP start up overhead costs.

## *C. 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. The control protocol messages help mobile clients to avoid unexpected connection losses and preemptively fetch the connection transition variants.

PEGASUS proposes to use client's cellular connection as the reliable channel, because the control messages send minimal amount of data (therefore we don't need a high bandwidth connection), and the cellular infrastructure is already available and supports our reliability needs. The dependable nature of the control communications ensures that mobile clients always have the latest access point maps, and will be able to make intelligent choices to handle unexpected conditions of the mobile environment.

Now, we present a quick overview of the possible control functions to support architecture goals stated in the beginning of this section:

- Our infrastructure needs dynamic discovery and mapping of the available open wireless access points. PEGASUS should allow additions of the new APs and removal of the no longer existing routers. Also, in order to perform low-overhead transitions, PEGASUS caches several distinct DHCP connection tuples per access point that are later shared by clients passing through the wireless network.
- The client should be able to request a list of available connections near the its location. Moreover, the manager can anticipate the client's movement pattern and provide additional connection tuples along the client's path. Such forecasting permits fast connection transitions (since no additional communication is required when the client is actually prepared to switch). Also, the client is able anticipate periods when the wireless communication might not be available.
- The PEGASUS client needs to notify the manager of connection transfers, in order to ensure a one-to-one client-wireless connection mapping at any given time.
- Finally, the manager can use the control protocol messages to authenticate the client, and to negotiate any security/encryption parameters for the client-manager UDP tunnel.

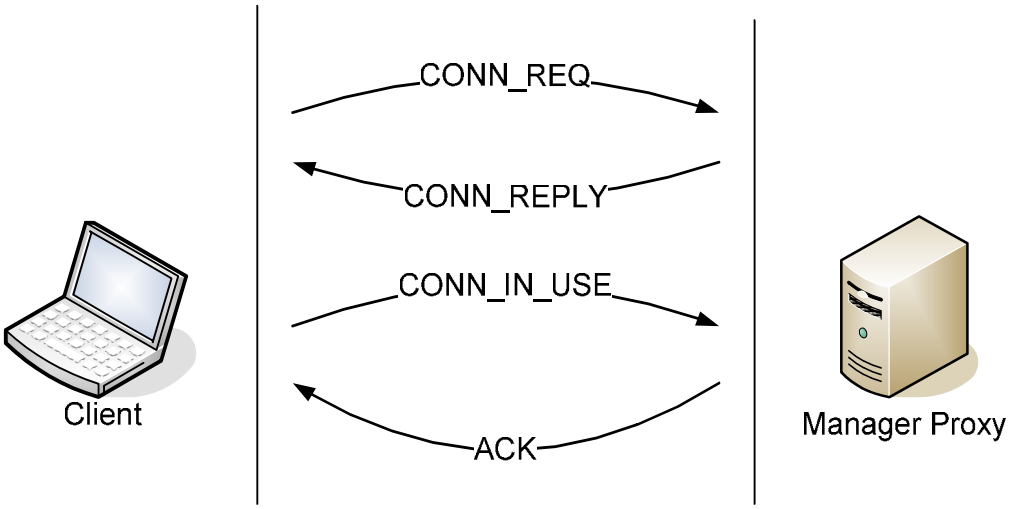


**Figure 3 Client - Proxy Control Message Flow**

Figure 3 demonstrates a sample client/manager control message flow. For our initial prototype, we have not implemented all of the possible control messages, and concentrated on the pieces involved in an actual connection switching mechanism. The manager Acess Point map is built by passive nodes that discover WLANs in their range. They attempt several DHCP requests with different credentials, and send created connection tuples to the manager.

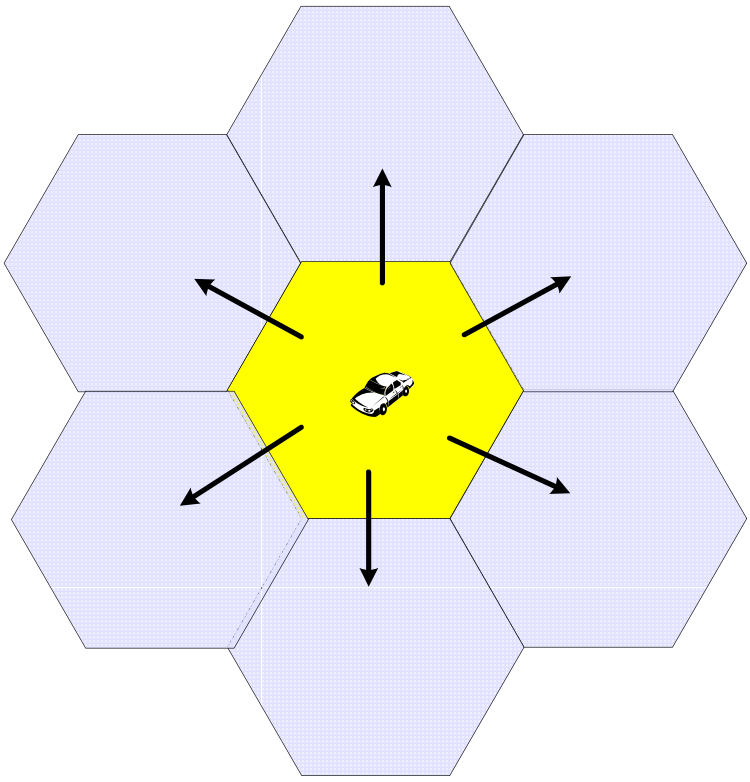

**Figure 4 Client Connection Switch Options**

Once a client needs to connect to the Internet, it sends a connection request to the manager with location coordinates, and receives a response with a list of connections in the proximity (Figure 4). Therefore, the mobile node can select a connection with the best signal, and it already has information about 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. As the client approaches the edge of the connectivity area, it will send another "connection request" and transition to the next connection.

The next step is to expand our control messages to combine the discovery process with the connection switching, to support both functions within a single client-manager session.

### *D. Applicability*

In summary, it should be clear the PEGASUS provides the connection splitting mechanism between the rapidly moving client, and the fixed endpoints. The control messaging interface offers a fast connection transfers, and dynamic access point discovery; and the protocol can be extended to further improve and optimize overall connection performance.

The required modification to the existing clients is a single executable module that will abstract the physical 802.11b/g connection, and use the cellular connection for control protocol messaging. The manager proxies can be independently managed entities, which do not need to cooperate. One possible coordination service, that in our vision could be beneficial to the overall infrastructure, is a shared global map of the discovered access points with the connection tuples. The proxies will need to coordinate the use of the connections to ensure that any given connection is allocated to a single mobile client at given time. Nevertheless, 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]

PEGASUS allows arbitrary deployment of proxies by volunteer participants, and offers a lot of room for system optimizations as it will become more widely used.

## IV. Measurements

For our prototype we have not implemented, all of the aspects of PEGASUS. Currently, the initial version supports the connection switching portion of the control protocol, and we disabled NAT at the access points to avoid unnecessary complications with the first draft. The discovery process occurs as a separate client session that reports discovered access points and connection tuples. Each tuple is comprised of client's wireless type, mac address and IP, along with the Access Point's IP, ESSID, mac and wireless hardware address.

For the development platform we picked an Ubuntu Linux distribution for both – the client and the server, and we use "Click Modular Router" project [1] to implement the routing and network specific modules. The client, is configured to listen for control messages on its reliable interface (Ethernet for our Lab setup), and in our experiments, the client changes its connection between two preconfigured Linksys 54G wireless access points. We wanted our system to be as generic as possible, so we opted to incorporate the Wireless Extensions for Linux [17] into Click. The PEGASUS client runs in user space which allows use of any 802.11 card supported by Linux. In our Lab setup we use the ipw3945 Intel wireless card which ships as standard 802.11g option with Dell laptops. The client applications use a virtual interface for outbound connections, and the packets are intercepted and forwarded to the manager. "Click" encapsulates all of the application session packets and sends them as UDP traffic to the manager proxy.

At the manager, we are running a modified "Click" NAT, and iptables configuration. The manager distinguishes the control traffic from the reliable interface, and passes them to the control messaging module. The rest of the traffic is extracted from the UDP tunnels, and the translated to use the manager's IP, as the stable communication address for the fixed endpoints. When the control module is notified of a client connection transition, it updates the NAT translation tables, to forward the client's traffic to the new IP address.

Our experiments revealed some optimistic findings. To test the effect of our connection transition process we set up two wireless access points and made our client request a connection switch every *n* seconds, and took an average reading on the rate of "wget" 100+ megabyte file download. The results are presented in figure 5

| **Switch Frequency** | **Dowload rate** |
|---|---|
| Never | 1.2 mb/s |
| 60 sec | 1.15 mb/s |
| 30 sec | 1.08 mb/s |
| 20 sec | 1.05 mb/s |
| 10 sec | 0.925 mb/s |
| 5 sec | 0.527 mb/s |
| 3 sec | 0.143 mb/s |

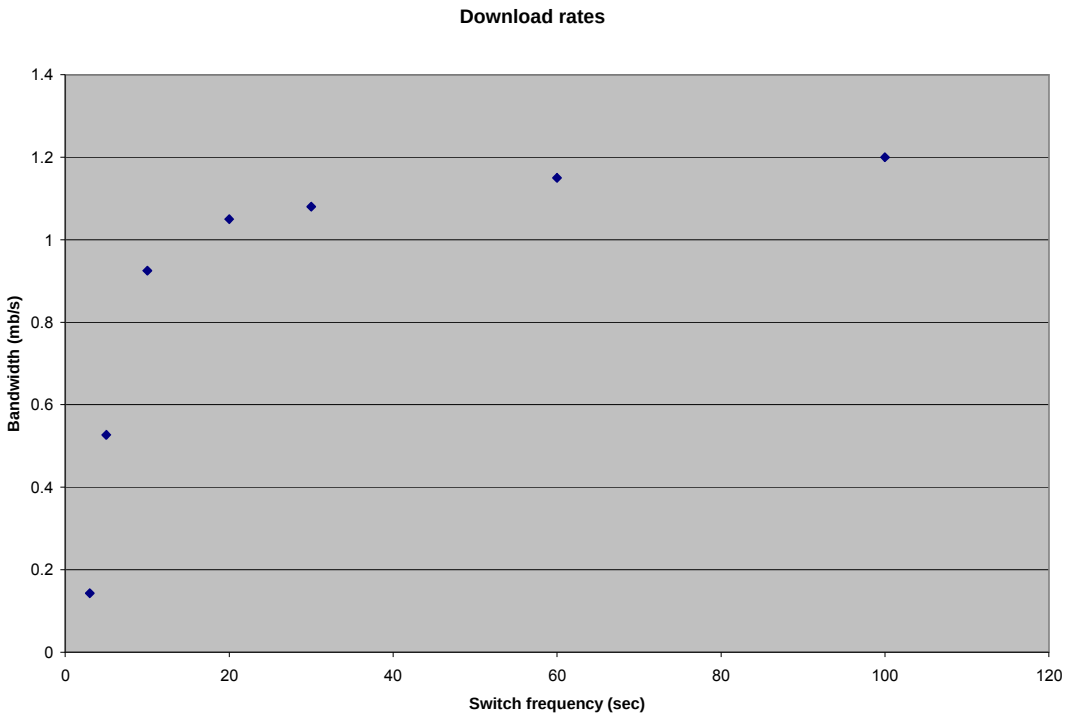


**Figure 5 Pegasus download rate vs. connection transition frequency**

The figure shows that without connection transitions, PEGASUS client is able to get an average download rate of 1.2 mb/s. As we introduce connection switches, the throughput decreases, but the connection still offers good transfer speeds. Keep in mind, that these results have been collected with our initial version, which has not been optimized, and does not employ any buffering. The results show that with connection transfers every, 60, 30, 20, 10 seconds the transfer rate goes down from 1.2 to 1.15, 1.08, 1.05, and.925, respectively. The main reason for drop in rates is increased interference between access points in our lab setup, and a number of TCP packets that are lost and retransmitted every connection transition (the TCP losses are due to a lack of buffering in the current prototype). Notice that with connection transitions every 10 seconds, we are still able to show only a 20% connection speed reduction from an optimal rate of 1.2 mb/s. At a velocity of 60 mph a car passes ~270 meters in 10 seconds, and this distance is bellow the connectivity ranges reported by the Drive-Thru Internet group. Moreover, with connection switches happening every 5 seconds, we were still able to achieve an average download speed of over 500 kb/s. Finally, our system was able to handle the connection switches every 3 seconds, and produce transfer rates faster than anything available to clients in moving vehicles today.

In the current implementation every connection switch requires ~ 1 – 2 second window to fully restore all of the application sessions. These times are significantly faster, than a DHCP request (6 – 7 seconds), and even when DHCP request is not required, application connections rebuilding in other projects take more than 2 seconds. In the end, the experiment, proved our 2 second limitation - the file transfer broke with connection

switches every 2 seconds. However, we think we can improve on this result in the next version.

## V. Conclusions

PEGASUS is an architecture to enable 802.11 connectivity for a fast moving vehicles. In PEGASUS, with a help of a manager proxy, and reliable control channel to a manager proxy, we showed a way to keep a persistent connection for mobile client applications. In our approach, when a client moves in an area with interleaved WLAN deployments, we demonstrated a method to efficiently transit from one WLAN to another without a connection disruption to the client applications. In addition, PEGASUS does not require client applications to change, nor it relies on any changes to today's network infrastructure. With this approach, we hope that we will be able to allow the 802.11 connectivity to mobile clients using "in situ" wireless networks. Finally, our preliminary experiments displayed very optimistic transfer rates for PEGASUS clients even in environments with frequent connection transitions. The next step in our project is to test PEGASUS in a real world environment, and the optimistic outcomes from the initial trials increase our hopes in introducing future network connectivity improvements for vehicular mobile clients with PEGASUS.

## References


[1] E. Kohler, R. Morris, B. Chen, J. Jannotti, and F. Kaashoek, *The click modular router*, ACM Transactions on Computer Systems, August 2000

[2] J. Ott and D. Kutscher, *Drive-thru Internet: IEEE 802.11b for Automobile Users*, In INFOCOM, 2004

[3] J. Ott and D Kutscher, *The "Drive-Thru" Architecture: WLAN-based Internet Access on the Road; accepted for publication*; In VTS, May 2004

[4] J. Ott and D. Kutscher, *A Disconnection Tolerant Transport for Drive-thru Internet Environments*, In INFOCOM, 2005

[5] Vladimir Bychkovsky, Bret Hull, Alen Miu, Hari Balakrishnan, Samuel Madden, *A Measurement Study of Vehicular Internet Access Using "In Situ" Wi-Fi Networks*, MobiCom, September 2006

[6] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, *Link-level measurements from an 802.11b mesh network*, in Proc. ACM SIGCOMM, August 2004

[7] *Homepage of FleetNet*, http://www.fleetnet.de/, 2003

[8] R. Gass, J. Scott, and C. Diot, *Measurements of In-Motion 802.11 Networking*, In Proc. WMCSA, April 2006

[9] Mattias Esbjornsson, Oskar Juhlin, and Mattias Ostergren, *The Hockman Prototype – Fast Motor Bikes and Ad-hoc Networking*, In Proc. MUM, 2002

[10] Ajay Bakre and B. R. Badrinath, *I-TCP: Indirect TCP for Mobile Hosts*, Department of Computer Science, Rutgers University, October 1994

[11] Hari Balakrishan, Srinivasan Seshan, Elan Amir, and Randy H. Katz, Improving TCP/IP Performance over Wireless Networks, in Proc. MobiCom, November 1995

[12] P. Rodriguez, R.Chakravorty, J. Chesterfield, Ian Pratt, and S. Banerjee, *Mar: A commuter router infrastructure for the mobile internet*, in MOBISYS, June 2004

[13] Bharat Bhargava, Xiaoxin Wu, Yi Lu, and Weichao Wang, *Integrating Heterogeneous Wireless Technologies: A Cellular Aided Mobile Ad hoc Network (CAMA)*, Purdue University, 2004
[14] Alok Nandan, Shirshanka Das, Giovanni Pau, Mario Gerla and M.Y Sanadidi, UCLA, 2004
[15] R. Chakravory, P. Vidales, L. Patanapongpibul, K. Subramanian, I. Pratt and J. Crowcroft, Performance Issues with Vertical Handover – *Experiences from GPRS Cellular and WLAN hot-spots Integration*, In Proc. PerCom, March 2004
[16] *Mobile IP*, http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120t/120t1/mobileip.htm
[17] *Wireless Extensions for Linux*, http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Linux.Wireless.Extensions.html
[18]*Wireless Tools for Linux*, http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html