P-Handoff: A protocol for fine grained peer-to-peer vertical handoff

Jean Tourrilhes, Casey Carter[1]
Mobile Systems and Services Laboratory
HP Laboratories Palo Alto

HPL-2002-61
March 11th , 2002*

E-mail: jt@hpl.hp.com, ccarter@uiuc.edu

wireless,
IrDA,
802.11,
ad-hoc,
TCP/IP,
handoff

This paper describes a novel approach to using peer to peer wireless links, such as IrDA, BlueTooth and 802. 11b. We explain the concept of wireless diversity and its impact on developers and users. We also explore the limitations of classical vertical handoff protocols, such as the need of an infrastructure. We describe our Connection Diversity framework, upon which is built P-Handoff. We introduce P-Handoff, a new handoff protocol for peer to peer links with fine granularity, which complements vertical handoff and allows to optimise connectivity through multiple wireless links. P-Handoff doesn't require an infrastructure and uses link layer events. We describe some features of our implementation under Linux.

* Internal Accession Date Only Approved for External Publication

[1] University of Illinois at Urbana-Champaign, Urbana, IL 61801

P-Handoff :
A protocol for fine grained peer-to-peer vertical handoff

Jean Tourrilhes and Casey Carter

jt@hpl.hp.com
Hewlett Packard Laboratories
1501 Page Mill Road, Palo Alto, CA 94304, USA.

ccarter@uiuc.edu
University of Illinois at Urbana-Champaign
1304 W Springfield Ave, Urbana, IL 61801.

This paper describes a novel approach to using peer to peer wireless links, such as IrDA, BlueTooth and 802.11b. We explain the concept of wireless diversity and its impact on developers and users. We also explore the limitations of classical vertical handoff protocols, such as the need of an infrastructure. We describe our Connection Diversity framework, upon which is built P-Handoff. We introduce P-Handoff, a new handoff protocol for peer to peer links with fine granularity, which complements vertical handoff and allows to optimise connectivity through multiple wireless links. P-Handoff doesn't require an infrastructure and uses link layer events. We describe some features of our implementation under Linux.

1 Introduction

Wireless data technologies promises high benefits to the user, such as ubiquitous and mobile computing, but for those to be realised, both application developers and users need to make sense of it. For now, they are both overwhelmed and puzzled by the complexity and diversity of available implementations.

The goal of this paper is to explore the impact of this wireless diversity on users and developers. As part of the CoolTown project [10], we are especially interested in how we can harness this wireless diversity in appliances, how we can manage peer to peer wireless links and how we can make wireless connectivity transparent to applications (especially mobile and ubiquitous applications).

2 Wireless Diversity

Wireless connectivity is no longer a dream : various technologies are available at reasonable price. Most of these technologies are mature and reliable.

2.1 Main wireless technologies

Most wireless technology in use today fall in one of the following four main categories.

2.1.1 Infrared

Infrared, such as IrDA [3], offers a simple, directional, ad-hoc way to communicate to close range devices. It is well suited for short ad-hoc transactions directly between peers as well as to/from a network infrastructure [13].

On the other hand, because of the need to keep line of sight, it is tedious to use infrared to perform data exchanges exceeding one second.

2.1.2 Wide area cellular connectivity

Cellular networks [4] are improving their ability to carry data traffic ; the third generation cellular networks (3G) will increase the bandwidth available to those services, improving both throughput and latency. Despite this, cellular will still remain slow and expensive (billing charges) compared to alternate local wireless technologies.

However, cellular connectivity has a major advantage : the wide coverage available. Therefore, we can assume that cellular can always provide a connection to the infrastructure.

2.1.3 Wireless LANs

Wireless LAN [1] is a technology deployed locally, and offer isolated pockets of connectivity. It may be ad-hoc (single cell) or may be connected to the infrastructure.

Wireless LANs, such as 802.11b, offer higher speed, lower price and less power consumption than cellular connectivity, but have much more restricted coverage.

2.1.4 PANs (Personal Area Networks)

PAN [2] is a technology designed to bind together the distributed parts of a logical system, like attaching peripherals to a main unit (Wireless USB). In addition, it is possible to use this technology to also connect to peers and access points (in a similar fashion to Wireless LANs).

Table 1: Characteristics of main wireless technologies

	Infrared	*Cellular*	*WLAN*	*PAN*
Range	2m ; 30˚	Country	Building	10m
Speed	Mb/s	kb/s	Mb/s	Mb/s
Power	Low	High	Medium	Low

2.2 The need for multiple links

Most proponents of a particular wireless link technology see their technology as “the” solution to all connectivity problems. However, the laws of physics still apply, and no solution will cover all possible situations and configurations.

In particular, each wireless link technology may have widely different characteristics in term of coverage, performance and cost, and each technology offers a different compromise between these three characteristics (for example, shorter range allows higher performance).

For fixed appliances or single applications, this is not so much a problem, as the designer can often pick the link technology most suited to the expected usage conditions.

In contrast, mobile appliances may be used in a range of environments and connectivity conditions may vary widely

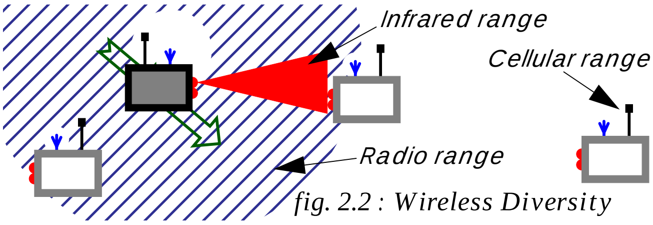


fig. 2.2 : Wireless Diversity

over the time. One special case is personal appliances : the user expects to use his personal device for various interactions with various other appliances, each using a specific wireless technology.

To address this wide range of needs and situations, it is likely that mobile appliances will be equipped with multiple link layer technologies (Wireless Diversity - see *fig. 2.2*). For example, they may have :

- an infrared link for ad-hoc directional interactions.
- a local radio link to have a fast and cheap connectivity directly with other appliances or with the infrastructure.
- a cellular connection for wider roaming.

The CoolTown experience [10] has already shown the practical benefits of Wireless Diversity (IrDA and 802.11).

2.3 The Babel of Wireless Technologies

Not only do wireless technologies have different operating characteristics, but most of them have been designed with very different APIs and user interfaces. This requires the user to manage each link differently and applications must be specialised to a particular link layer.

There is also a wide variety of possible topologies for the different wireless links. Some are point to point, others are broadcast. Some offer access to the infrastructure, or others offer only local connections. This impacts how the user and the application need to manage the link.

For example, IrDA and BlueTooth are usually local and point to point, whereas 802.11 is usually broadcast and connected to the infrastructure. The main API of IrDA is a dedicated socket interface (IrSock), for BlueTooth, it is a tagged serial pipe abstraction (RfComm), and 802.11 uses regular TCP/IP sockets. Peer selection on 802.11 uses DNS name or IP address, BlueTooth uses nickname or MAC address, and IrDA doesn’t require addressing.

2.4 Management of multiple links

Some devices are already including multiple wireless link technologies to benefit from wireless diversity. However, the management of those links is usually left to the user.

The current emphasis is on the link layer. The user picks a link layer, sets it up, picks an application for this link layer and performs a transaction on this link layer. But ultimately, the user does not care what link layer is used, he just wanted the data to be shipped to the intended recipient.

Therefore, our first goal is transparency. This means that the various link layers should be transparent to the user and also to the application, including device addressing and configuration. Users and application developers should not perceive the system as a collection of separate links but instead as integrated connectivity.

The second goal is that the choice of the link should be optimal for each application or network transaction at any time. This implies handling the dynamics common in most wireless environments (links may come or go), continuously offering the best service without disrupting communications, to really take advantage of the wireless diversity. Note that the definition of “best” depend both on the user and the application (see *section 4.5*).

3 Classical Vertical Handoff (V-Handoff)

Our solution to improve transparency in case of multiple wireless technologies, P-Handoff, is based on the large body of work done in the area of wireless handoff.

3.1 Horizontal and vertical handoffs

Wireless links have only a limited range, so when a node physically moves within a cellular network, it needs to change its point of attachment to the infrastructure. This process of migrating from one Access Point to another is called handoff, handover or roaming.

The first instances of vertical handoff [8] were found in overlay networks, where different cellular architecture are mutually overlaid in the same area offering different type of coverage (from local high speed to wide area lower speed). A standard handoff (horizontal) is simply migrating within the same cellular architecture, and a vertical handoff is moving from a cell in one cellular architecture to a cell in a different architecture (for example GSM and 802.11).

By extension, we will call vertical handoff any handoff from one wireless technology to another wireless technology, even if those are not cellular. Of course, a V-Handoff requires the node to have more than one wireless network interface.

The main difference between various V-Handoff protocols are how the system determines which wireless links are available and how it redirects network traffic on the selected wireless link (especially the downlink traffic).

3.2 Mobile IP based

The initial V-Handoff proposal [8] was entirely based on Mobile IP. Mobile IP [5] is a generic technique to perform handoff of IP traffic between IP subnets using straightforward routing techniques and IPIP encapsulation, and is well suited to do vertical handoff.

Mobile IP advertisements are used to discover which links are available ; Mobile IP routing ensures that both incoming and outgoing IP traffic uses the chosen interface.

Relying on MobileIP advertisements for detection of link state change is slow. A common optimisation is to directly use link layer information : most wireless interfaces can be queried for the current state of the link layer.

Standard Mobile IP has also its share of problems, like triangular routing and firewall traversal, but bi-tunnelling and routing optimisation solve most of these.

3.3 TCP/DNS based

Recently, some work has been done to do V-Handoff at the TCP protocol level [9]. This eliminates some problems of the Mobile IP approach (triangular routing) but introduces new ones (see *section 3.4*).

The IP address of the node is no longer static but is an IP address valid on the link being used (i.e. part of that subnet) and subject to change. The DNS name of the node is its globally unique address ; the node uses DNS updates to update its DNS record with its current IP address.

Any peer that wants to connect to that node has to query the DNS for the current IP address. Once a TCP connection has been established, handoff is done using specific TCP options to migrate traffic from the old TCP connection to a new TCP connection (TCP migration).

3.4 The need for TCP/IP for V-Handoff

There are various points of the networking stack where we could perform V-Handoff. It can be done at the link layer, above the link layer, at the IP layer or at the TCP layer.

Horizontal handoff are often done within the link layer. However, the link layers of the wireless technologies used in vertical handoff use different protocols and implementations (see *section 2.3*), so the handoff can't be done at this layer.

Some wireless links are not TCP/IP native and have their own specific protocol stack. For those links, we need to do the handoff above the protocol stack. Doing it directly above the protocol stack (like between IrSock and RfComm) is difficult, because important state and data is embedded in the protocol stack that can't be transferred to the new link. Also, the APIs are different, requiring the application to manage this transfer explicitly, so this is not transparent for the application.

The easiest way to do handoff between different link layer protocol stacks is to use a transparency layer above those stacks. The most common transparency layer existing today is TCP/IP (it provides transparency and reliability). Moreover, most network applications are based on TCP/IP. This is why most solutions for vertical handoff are based on TCP/IP.

Doing handoff at the IP level (see *section 3.2*) is natural because IP is stateless, connectionless, and common for most links, and it mostly amounts to changing a few routes in routing tables. Through the handoff the application keep using the same TCP/IP socket, providing transparency.

Doing handoff at the TCP level (see *section 3.3*) still achieves transparency, but is more complex because some TCP state has to be migrated from one connection to the other (like the current packet queue and window). All fixed peers on the Internet also need to be modified to support TCP migration. And supporting applications that pass IP address in the payload of the packets or cache IP address in the application between requests is problematic (like in NAT).

3.5 Limitations of V-Handoff

The first limitation of V-Handoff is that it treats all connections in the same way. Only one wireless interface is used at a time (the best one) and all TCP/IP connections are migrated from one interface to another simultaneously.

The definition of "best" rests ultimately with the end user and specific application (see *section 4.5*), and in most cases it may be appropriate to have a finer grained approach and use different strategies for different connections.

The second limitation of V-Handoff protocols is that they require a common network infrastructure. Both the mobile node and peers need to be able to reach the Mobile IP Home Agent or DNS server. All wireless interfaces used must therefore be part of the same MobileIP or DNS infrastructure.

These two limitations mean that vertical handoff can't integrate ad-hoc technologies such as IrDA, BlueTooth and ad-hoc-802.11b, because each ad-hoc network contains only a few nodes that may not be part of the infrastructure (direct peer to peer). This means that we are losing an important part of the wireless diversity, because those peer to peer connections are usually the most efficient (they can often shortcut the slow and expensive infrastructure).

4 Connection Diversity

The P-Handoff protocol (see *section 5*) is a part of our Connection Diversity framework and interacts tightly with other components of this framework and depends on them.

4.1 The Connection Diversity framework

The Connection Diversity framework aims to manage a device connectivity to its immediate peers (one hop). It mostly reuses existing components of the network stack and existing techniques, combining them in a new way. The TCP/IP stack and Link layers are unchanged, but their API need to be extended for additional events and control. On the other hand, applications built upon them are unmodified.

The main new piece of functionality is the Connection Manager, with its Policy Manager, which is in charge of managing the link layers (see *fig. 4.1*). The P-Handoff protocol is mostly implemented in the Connection Manager.

4.2 The IP adaptation layer

Using TCP/IP on all wireless links goes a long way toward providing transparency, because all the applications and users see a common interface. However, setting up TCP/IP on some wireless links can be quite challenging and may require a complex IP adaptation layer.

There is two classes of link layers that we need to deal with, the first class is broadcast connectionless (802.11), and the second class offers point to point connections (IrDA, BlueTooth). The first class is usually TCP/IP native and simpler, the second class usually is problematic and needs a way to map IP traffic on the link layer. Solving these problems is outside the scope of this paper, and designing efficient IP adaptation layer is already being dealt elsewhere [11].

The first task of the IP adaptation layer is to provide the delivery of IP packets over the link layer to the intended destination. Then, P-Handoff needs the IP adaptation layer to be extended to provide facilities to do IP discovery (see *section 5.3*) and to monitor the link (see *section 5.4*).

4.3 On-demand TCP

Most often, it's impossible to just direct IP packets on a wireless link without initially either configuring it or establishing a link level connection on it. This is particularly true on point to point connected links (IrDA, BlueTooth).

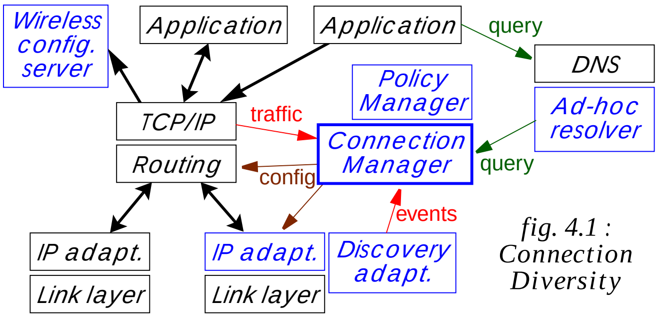


fig. 4.1 : Connection Diversity

The link layer, together with TCP/IP, should provide autoconfiguration, both at the link and the TCP/IP level, so that the user is not involved in setting up connections manually, which would defeat out goal of transparency.

We use the On-Demand TCP protocol [11] to set up and tear down link layer connections and minimise the usage of link layer resources. By default, all link layer connections to a peer are disconnected, and IP to this peer is redirected to the Connection Manager through a special pseudo interface. When the Connection Manager receives an outgoing packet, it selects the best link associated with the destination IP address and setup it up. On connected links, it uses the IP adaptation layer to create link layer connections, on broadcast links it uses a simple UDP handshake to probe the link.

4.4 Ad-Hoc name resolver

Since most applications and user interfaces deal with names and not directly with IP addresses, the system needs to perform name resolution. On links which are connected to the infrastructure, we can use regular DNS services, but on ad-hoc links, those may not be accessible.

Our solution is to use an ad-hoc name resolver [11]. The Ad-Hoc resolver is specific to each link layer and uses the Discovery Adaptation layer and a lightweight protocol to resolve the names into IP addresses. In most case, the IP discovery (see *section 5.3*) and name resolution protocols are combined to minimise overhead.

Most of those Ad-Hoc resolvers are able to resolve both fully qualified DNS names and link names. Link names have the scope of a specific link layer, are based on the link layer nickname of the peer and use the "dot" notation *<nickname>.<link>* (for example *bougret.irda*). The additional suffix "*.adhoc*" aggregates the various local link name-spaces. Some link resolvers can also use special nicknames or service attributes (such as *printer.any.bt*).

4.5 Policy Manager

All handoff protocols are supposed to redirect IP traffic to the best link layer, but we have not yet defined what "best" means and how link layers are evaluated.

Comparing link layers is not an easy problem, particularly since optimisation of connectivity may be done for a variety of goals, such as maximum throughput, minimum latency, lower power consumption, lower billing charge (cost), or lower disruption of connectivity. I can also depend on the connectivity history, the user preferences, the requirements of the application and other QoS factors.

For this reason, we have decided to keep this optimisation in a separate module, the Policy Manager, allowing multiple policies over the same framework. Our current policy is the most simple ; each link is assigned a fixed priority and we always pick the available link with the highest priority.

5 The P-Handoff protocol

The goal of P-Handoff is to extend the classical concept of V-Handoff to fully exploit wireless diversity and to deal with a wider variety of wireless links and configurations available.

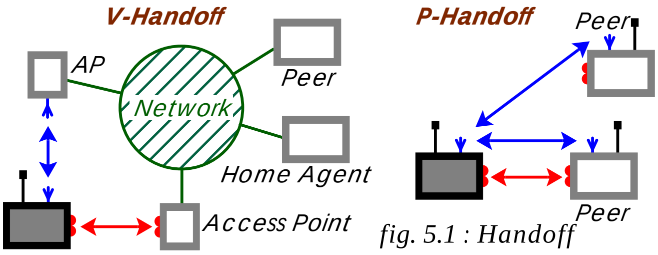


fig. 5.1 : Handoff

The main challenge of course is to deal with ad-hoc wireless links (direct peer to peer).

5.1 Main characteristics

P-Handoff is complementary to Mobile IP based V-Handoff and it is based the same assumptions (even though it doesn't use Mobile IP). P-Handoff deals mostly with handover between ad-hoc links (without infrastructure), that are only one hop away (peer to peer direct communication).

All user communications are IP based, traffic is rerouted using IP routing. Each node has a Global IP Address (the equivalent of the Home Address in Mobile IP). P-Handoff doesn't route traffic via the home network, but uses the Global IP Address mainly to uniquely identify a node.

The granularity used by P-Handoff is the IP destination address. For each Global IP Address, the set of links that can be used to reach this address is computed, and the best link selected. Each Global IP Address may be routed on a different link and, therefore, all links may be used simultaneously. Having a per-connection granularity would be expensive with minimal added benefit, so was not considered.

5.2 State machines and events

The state machines implemented in our prototype are quite complex, due to the link layer abstraction, error conditions and the interaction with On-Demand TCP. However, the concept behind P-Handoff can be described through two events and two simple state machines.

If a node we are connected to is discovered on a different link layer (*discovered* event), the protocol evaluates this link, and if the new link is better than the currently selected link, reroutes traffic for this node to the new link (see *fig. 5.3*).

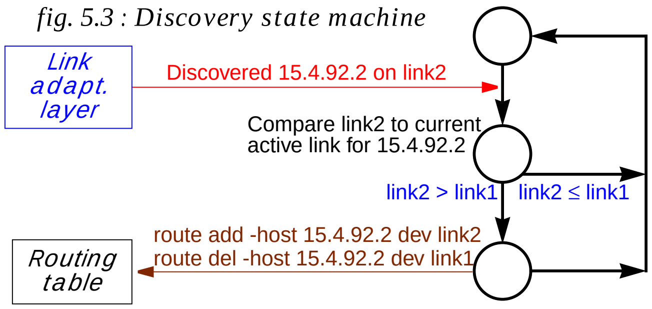
Similarly, if the current connectivity to a node is broken on its active link (*unreachable* event), the protocol reroutes the traffic on the next best available link (see *fig. 5.4*).

5.3 IP discovery and matching

In V-Handoff, the protocol only needs to know the presence of an infrastructure to decide if it can use a link. With P-Handoff, we go down to the granularity of peer, because each link can connect to a limited subset of peers.

P-Handoff uses a discovery protocol to collect the Global IP address of each peer available through each ad-hoc interface. If peers discovered through two different interfaces have the same Global IP Address, we assume it's the same node and that it can be reached via two different paths.

Most point-to-point connected links offer a native discovery protocol. In those cases, our Global IP discovery protocol is based upon it, through the Discovery Adaptation Layer [10]. This solution avoids unnecessary TCP/IP fig. 5.3 : Discovery state machine


Link adapt. layer
Discovered 15.4.92.2 on link2
Compare link2 to current active link for 15.4.92.2
link2 > link1
link2 ≤ link1
route add -host 15.4.92.2 dev link2
route del -host 15.4.92.2 dev link1
Routing table

connections and is efficient (the link layer protocol has already done most of the work).

For connectionless broadcast links offering no native discovery protocol, a simple UDP multicast request-response protocol is currently used, but Neighbor Discovery for IPv6 [6] could also be used.

5.4 Connection monitoring

One of the main performance constraints of any handoff mechanism is quickly detecting when connectivity fails and adjusting for it. The user and application often won't mind if the traffic is not optimally routed, but connectivity needs to be maintained.

V-Handoff may use the link layer state to monitor connectivity, however most ad-hoc links never change state, only individual nodes on this link may become unreachable or come back into range. The IP discovery protocol doesn't give positive event on expiry and is usually too slow.

For most point-to-point connected links, an event can be generated to indicate failure of a specific link layer connection. For broadcast connectionless links, we have been experimenting with an event generated each time a packet is lost due to excessive link retry. In each case, the IP Adaptation Layer matches this event to the relevant Global IP address.

These events are not 100% reliable and not always available, so we additionally monitor the IP traffic itself and probe the link when idle to generate an "unreachable" event.

5.5 Tunneling and Routing

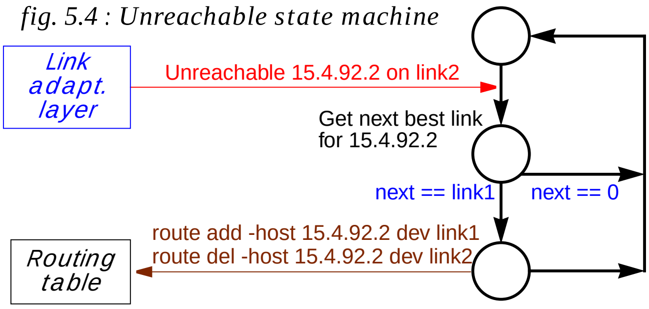
Like in Mobile IP, the destination address of all network traffic is the Global IP Address, so that traffic is independent of the current interface being used and can be migrated.

Directing the IP traffic on the selected interface is trivially done using a host route in the IP routing table (an IP route which applies only to a single IP address). The routing table contain an host route for each Global IP Address actively managed by the Connection Manager and directed on one of the ad-hoc links. No additional routing is needed because the system is strictly peer to peer (one hop).

On the other hand, the various network interfaces of a node may have different IP addresses (Local IP Addresses), depending on the result of the autoconfiguration process. Those Local IP Addresses are often not the Global IP address and may change over time due to mobility.

Mobile IP uses simple IPIP tunneling to get around this problem [5]. However, as we address only single hop connectivity, we can do even simpler. If the IP adaptation layer uses PPP, the Local IP Address (the end of the PPP tunnel) can be the Global IP Address [11]. For other cases, we just set the host route using the current Local IP address as a gateway (this is similar in spirit to using Proxy ARP [7]).

fig. 5.4 : Unreachable state machine


Link adapt. layer
Unreachable 15.4.92.2 on link2
Get next best link for 15.4.92.2
next == link1
next == 0
route add -host 15.4.92.2 dev link1
route del -host 15.4.92.2 dev link2
Routing table

5.6 Synchronising forward and return path

Our system is peer to peer, and each node runs the same Connection Manager. However, each Connection Manager may have different policies and receive different events at different times, so they may make different routing decisions.

This means that between two nodes, the forward and return path of the traffic may use different links, because each Connection Manager may route its outgoing traffic differently. Forcing both forward and return path to use the same link frees up resources on the other link, allowing its use by other connections or shut it down entirely.

To synchronise those paths, we plan to introduce roles in the Policy Manager. The node initiating the connection would be the master and would make all the handoff decisions. The target of the connection would be the slave, and it would just mimic the decisions of the master. The slave only needs to listen for connection requests from this peer on the various links to know what the master is doing (either the link layer event on connected links or the simple UDP handshake on broadcast links).

5.7 Infrastructured links and V-Handoff

P-Handoff only manages ad-hoc links (direct peer to peer), and not links connected to the infrastructure.

Most often, through the infrastructure, the node can reach the whole Internet, so IP discovery on the infrastructure would return a potentially large number of addresses (a subset of the Internet). To avoid this, we just assume that any link connected to the infrastructure can be used to reach any peer also connected to the infrastructure, via the default route.

P-Handoff can be used alone, with a single infrastructured link (default route), or combined with V-Handoff to offer full handover across all wireless links.

V-Handoff manages the default route, migrating the it to the best infrastructured link by querying the state of those links (connected or not). Any peer that is not discovered across an ad-hoc link is not managed by P-Handoff, and it will use the default route set by V-Handoff. If the connection to a peer is broken and P-Handoff can't find any alternative ad-hoc link, it will redirect its traffic on the default route. In other words, P-Handoff shortcuts V-Handoff when possible.

P-Handoff could probably also be integrated with Ad-Hoc Routing which may be use on some of the ad-hoc links, but this requires further investigation.

6 The Implementation

We have implemented P-Handoff as part of an experimental Connection Diversity framework. The implementation was done under Linux and has been demonstrated with real, unmodified applications.

6.1 The Linux implementation

The Connection Diversity framework is implemented as several modules using various parts of the Linux OS. The Connection Manager is implemented as a daemon process. The Link Layer Adaptation module is both in the kernel networking stack and in the daemon. The Ad-Hoc resolver is both in a C Library module and in the daemon.

The current implementation only manages IrDA and ad-hoc 802.11b connectivity, which are representative of two extremes of wireless topologies and APIs.

6.2 The IrDA subsystem

Most of the IrDA subsystem has already been described in our previous paper [11] and is related to our implementation of the IrNET protocol. IrNET provide a very efficient way to carry TCP/IP traffic over IrDA in point to multipoint connections. The IrNET control channel allow the Connection Manager both to control precisely the connection setup and to receive various events related those connections.

One problem of IrDA is that the IrLAP connection is very persistent and will timeout only after 12 seconds elapses without any response. Of course, most of the time we will have already rerouted IP traffic on another link well before that (thanks to the “Blocked link” event), but it also means that we can’t reuse the link in the meantime.

6.3 The Wireless LAN subsystem (ad-hoc 802.11b)

The WLAN subsystem mostly manages 802.11b ad-hoc links and will be described in a subsequent paper. 802.11b is IP native broadcast medium, so no connection setup is really needed (we have a simple handshake). IP traffic is simply directed on the link using a host route. A simple periodic multicast protocol enable IP discovery and name resolution.

To detect connection failures, a specific Wireless Event is generated by the driver when MAC retries are exceeded for a specific outgoing packet. This event carries the destination MAC address of the packet and is part of the standard Wireless Extensions for Linux [12].

6.4 P-Handoff Performance

The overall performance of the protocol can not be evaluated until we define performance metrics and implement a policy manager tailored to that goal.

The handoff performance itself is mostly governed by the characteristics of the individual link layers and latency of the events triggering handoff, leading to some effort optimising those events (especially the *unreachable* event).

The typical TCP/IP handoff latency between IrNET and 802.11 is 1.4 s (time between the last IP packet transmitted on IrNET and first IP packet transmitted on 802.11), and the typical TCP/IP handoff latency between 802.11 and IrNET is 0.9 s. This is acceptable for most Internet applications.

The detailed values in *table 2* are typical of our implementation. If the OS is busy (paging from disk) or if the medium is busy (interference), those values may be higher.

Table 2: P-Handoff performance

Link layer latencies	*IrNET*	*802.11*
Discovery period	3 s	10 s
Connection setup	~0.8 s	~0.3 s
Unreachable event	1 s	~0.1 s
Connection closed event	12 s	10 s

7 Conclusions

Wireless diversity is presently a source of confusion for the user, but has many opportunities to dramatically improve the versatility of connectivity to peers and services. Some handoff protocol is needed to pick the best available link for any connectivity and redirect it based on user roaming.

The P-Handoff protocol complements classical vertical handoff, redirecting traffic to the best ad-hoc link on a peer by peer basis. P-Handoff uses simple IP routing techniques and integrates well in our Connection Diversity framework. In addition, P-Handoff doesn’t require any infrastructure support and uses link layer discovery and unreachable events to drive the protocol behaviour.

P-Handoff has been implemented in the Linux OS over IrDA and 802.11 link layers as part of our Connection Diversity framework. It has been demonstrated with real, unmodified TCP/IP applications. Its performance is good and is only constrained by the link layer implementation.

8 References

[1] *IEEE 802.11 : Wireless LAN medium access control (MAC) and physical layer (PHY) specifications*. IEEE.

[2] J. Haartsen, M. Naghshineh, J. Inouye, O. J. Joeressen and W. Allen. *BlueTooth: Vision, Goals, and Architecture*. ACM MCC Vol. 2, No. 4, 1998.

[3] Patrick J. Megowan, David W. Susak and Charles D. Knutson. *IrDA Infrared Communications : An Overview*. www.irda.org.

[4] M. Mouly, M.-B. Pautet. *The GSM System for Mobile Communication*. Published by the authors, 1992.

[5] C. Perkins. *IP Mobility Support*. RFC 2002.

[6] T. Narten, E. Nordmark, W. Simpson. *Neighbor Discovery for IP Version 6 (IPv6)*. RFC 2461.

[7] S. Carl-Mitchell, J. S. Quarterman. *Using ARP to Implement Transparent Subnet Gateways*. RFC 1027.

[8] Mark Stemm and Randy H. Katz. *Vertical Handoffs in Wireless Overlay Networks*. ACM MONET 3(4), 1998.

[9] Alex C. Snoeren and Hari Balakrishnan. *An End-to-End Approach to Host Mobility*. Proc. of MobiCom 2000.

[10] CoolTown team. *People, Places, Things: Web Presence for the Real World*. www.cooltown.com

[11] Jean Tourrilhes. *On-Demand TCP : Transparent peer to peer TCP/IP over IrDA*. Proc. of ICC 2002.

[12] Jean Tourrilhes. *Linux Wireless LAN Howto*. http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux.

[13] Jean Tourrilhes. *e-Squirt for Linux-IrDA*. http://www.hpl.hp.com/personal/Jean_Tourrilhes/IrDA/