Co-Link configuration : Using wireless diversity for more than just connectivity

Jean Tourrilhes and Venky Krishnan

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

This paper describes a novel approach to setting up peer to peer wireless links, such as IrDA, BlueTooth, and 802.11b. We explain how the lack of physical contact inherent in wireless creates new configuration challenges and ease of use issues, and we discuss the limitations of current wireless autoconfiguration techniques. We present Co-Link Configuration and Co-Link Activation, two techniques using wireless diversity to improve the performance and minimise the power consumption of autoconfiguration. We then describe two implementations of Co-Link, the simple IrDA Connection Point and the peer-to-peer Co-Link implemented within the Connection Diversity framework.

1 Introduction

In the computer industry, the expression "Plug-and-Play" has become mythical, a relentless quest in making things easier and transparent for the user. Gone are the traditional installation of drivers and configuration of I/O addresses and interrupts, except when Plug-and-Play fails.

Networking has always been an hard area for normal users : many things need to be configured, due to the distributed nature of the problem. Wireless networking makes this even more difficult.

This paper looks at the challenge of wireless autoconfiguration and proposes a solution, Co-Link Configuration, solving some configuration issues found in typical wireless networking scenarios.

2 Connection Diversity

This work is part of the Connection Diversity project at HP Labs, a project addressing multiple aspects of wireless networking.

2.1 Motivation & assumptions

Connection Diversity explores how mobile information devices can interact using the wide variety of wireless technologies existing today, with a special emphasis on peer-to-peer and ease of use. One goal is to bring the ease of use of wireless technology to the same level as removable storage (floppy, Compact Flash storage).

A principal underlying assumptions of Connection Diversity is wireless diversity : the availability of multiple wireless technologies with different characteristics in each information device. Connection Diversity also assumes that most wireless devices will perform user triggered transactions with either direct peers or the infrastructure [10].

2.2 On-demand TCP and P-Handoff

The work presented in this paper builds on previous work done with On-demand TCP [10] and P-Handoff [11].

On-demand TCP enables peer to peer TCP on a wide variety of wireless links [10]. TCP/IP connections are established and configured on-demand when applications need them, between two peer devices, without the need for infrastructure. We have implemented On-demand TCP over IrDA and 802.11.

P-Handoff enables transparent migration of peer to peer TCP connections between wireless links [11]. P-Handoff doesn't require any infrastructure and is fine grained, allowing flexible use of available links. A Policy Manager tries to optimally use those links for each connection based on range, speed and cost.

3 The main challenge : configuration

Configuration is a traditional barrier to ease of use. With wireless technology, configuration is usually even more complex than with traditional wired technologies [12].

3.1 The lack of physical contact

A physical contact between two entities establishes a context and a medium. With a wired link the physical action of plugging a cable in the device binds this device to another device or an infrastructure. With removable storage, inserting the media in a slot binds the contained data to this device.

The main benefit of wireless technology is that it removes all physical contact between the device and other devices or the infrastructure. The cost is extra configuration needed to create a virtual context replacing the physical contact.

The most important class of parameters is administrative configuration. Typically, a wireless identifier (such as the ESSID in 802.11 [4]) binds a wireless device to another device or an infrastructure to create a virtual network.

In addition, some wireless technologies may require physical layer configuration to interoperate properly [12]. These are some physical layer properties, such as modulation, frequency or timing information (needed to tune the wireless receiver and synchronise).

Lastly, security configuration may be needed [12]. Wireless links are perceived as less secure than a wired links, so most wireless links require the configuration of an authentication and encryption key to promote privacy.

Pervasive computing applications also need context or location configuration to bind to a space [1]. The space aggregates the local services and resources relevant to the location [2] and often determines the administrative

configuration of the wireless link. Many resource discovery protocols already exist, therefore this paper limits itself to the separable issue of "finding connectivity".

3.2 Typical wireless configuration

These configuration parameters are usually configured manually in the device, either by the end user or his network administrator, or are discovered over the wireless link.

An **802.11** link [4] requires that the ESSID, the mode and the encryption key are configured *before* the device can join the network. A typical 802.11 configuration :

```
wvlan0    IEEE 802.11-DS  ESSID:"Hewlett-Packard"
      Mode:Managed  Frequency:2.447GHz
      Access Point: 00:02:2D:0C:94:93
      Bit Rate:11Mb/s  Tx-Power=15 dBm  Sensitivity:1/3
      Retry limit:4   RTS thr:off   Fragment thr:off
      Encryption key:5736-6C0E-4365-C35D-4F1F-4BD9-0E70
      Power Management period:100ms
```

A **BlueTooth** device [3] associates with a peer (and not a network) and needs to know either its MAC address or a set of service attributes (SDP) uniquely identifying this peer. From this, it auto-discovers the frequency hopping pattern and clock offset necessary to synchronise with the peer.

All those parameters are specific to each link technology, therefore the manual configuration or wireless discovery is done independantly for each link layer of the device.

3.3 Mobile Wireless networking

Wireless links enable mobility, therefore a wireless device may want to interact with a wide variety of peers and networks, each having a distinct configuration.

The simplest way to manage this mobility is to require the user to manually enter the new wireless configuration each time a change occurs. Unfortunately, the user may not have the necessary information readily available, and will quickly find such operations tedious.

Most wireless systems allow users to store multiple configurations and to either select one or to try them all. This minimises user intervention, but is limited to a set of known configurations. Those systems usually don't know when to change configuration, the user must manually trigger it.

The third option is to use an autoconfiguration scheme based on wireless discovery and have the user or the system pick one of the discovered configurations.

3.4 Wireless discovery and its limitations

Wireless discovery uses a link level protocol on the wireless link to probe the link, discovering the configuration needed to connect to networks or peers available within range.

Wireless discovery may fail (wireless is unreliable) or take a long time (several minutes). For some technologies, the set of physical layer parameters may be so large that discovery is not reasonable. Automatic wireless discovery consumes additional battery power, so the user will often disable it.

Wireless discovery cannot autoconfigure the security parameters. The main desired property of an encryption key is that it can't be guessed or discovered. Some networks or devices also won't answer discovery probes for security reasons, making them impossible to discover.

In most cases, discovery of administrative configuration is ambiguous, and doesn't resolve to a single choice. There may be multiple peers or networks within range, so the device still has to somehow choose among them. The device may rely on the user to pick one, but the choices may be cryptic and require user intervention. The other solution is to use heuristics, prioritise already known networks (with user preferences) or pick the network with the highest signal strength, but those may not be the most appropriate choice.

The characteristics of discovery for some common wireless links are detailed in *Section 4.1* and in *Table 1*.

3.5 Usage model and configuration

The necessity of full autoconfiguration depends on the usage model : many successful wireless products, such as GSM phones and 802.11b cards. are sold today without it. Most wireless connectivity falls in one of three main types of usage, *the net*, *my* and *this*.

The first type of usage, *the net*, is accessing remote peers via the infrastructure, typically services and information in the Internet or the phone system. In this case, the device may pick *any* access point to the infrastructure. Public access points allow themselves to be discovered, and authorised users will configure private access points that they use often. In many cases, the device needs to know of only one network.

The second type of usage is derived from the Personal Area Network vision [3], where a group of personal devices that belong to a user act as a coordinated whole. We call it *my* because each device belongs to the user, and he has configured them to work together and remember each other.

In these two types of usage, there may be the need for manual configuration. However, this configuration is often tolerated because it's a long term binding ; the user expects to use his device in this configuration a large number of times and the long term value overcomes the initial inconvenience. In many cases this configuration is done at the factory (cordless phones), in the shop (cellular phones) or by a system administrator (wireless LANs).

The third type of usage, *this*, is derived from the pervasive computing vision [1]. The user interacts in the real world with other people and appliances in spontaneous ways, and may want to complement the physical exchange with some data communication. He wants his personal device to interact with *this* other physical device (that he usually can see or touch).

Examples include exchanging business cards at a trade show, walking up to a printer in a public area and connecting to the specific local access point in a place you visit.

For this type of ad-hoc interaction, the duration of the binding between the two devices is only for the duration of the transaction, therefore the configuration can only be done at time of use. The user will be willing to do it only if the transaction has enough value and the cost of configuration is low enough. Without dependable autoconfiguration, it is likely that the user will seek alternative ways to avoid it, such as using removable storage [10].

4 Co-Link configuration and activation

Wireless discovery, when available, is a big improvement, but in most cases it falls short of our expectations of ease of use and dependability.

Co-Link configuration is a new mechanism based on wireless diversity, designed to deal with the problem of wireless autoconfiguration in spontaneous interactions. The main idea is to aggregate wireless discovery across all available wireless links instead of performing it only within the strict context of each individual link.

4.1 Wireless Diversity

Link layers have different characteristics. Various handoff protocols [11] already exploit the fact that they have different range, performance and cost. In addition, they also have different wireless discovery capabilities.

Table 1: Discovery characteristics of main wireless links

Discovery	*IrDA*	*BlueTooth*	*802.11b*
Type	Peers	Peers	Networks
Service attributes	Some	Rich	No
Speed	0.5s	20s	1s
Period	3s	minutes	minutes
Power (vs. Tx)	Low	High	Medium
Selectivity	2m - 30˚	10m	100m
Probability	Good	Average	Average

Some links may not have any discovery protocol, and not all discovery protocols are equivalent. Discovery protocols may find networks or peers, and may have various service attributes to identify the device functions.

The performance of the discovery protocol is mainly constrained by underlying link characteristics. Different links have different discovery speed, reliability and power consumption.

Power consumption affects how likely discovery will be enabled by the user. Discovery protocols often consume much more power than regular communications, since the device sends repetitive probes at full transmission power.

For transparent discovery, the period between discoveries is usually more important than discovery speed. Speed of discovery is constrained by the bit rate, the mean access delay to the medium, the number of channels/configurations that must be probed, and the listening behavior of peers. The period is a tradeoff between power consumption, traffic overhead and refresh delay, and can usually be tuned.

One of the most interesting aspects of this diversity is that different link layers have different scopes. The narrow beam of IrDA is very selective and can pin-point a specific device, while the large range of 802.11b often finds many networks.

The last criteria we consider is the probability that the discovery protocol will get an answer from the target network or peer. Some networks or peers won't answer for security reasons. Other may be already busy with some other activity that prevent them to do so ; a BlueTooth device needs to enter a specific mode (Inquiry [3]) and stop all communication in order to be able to answer discovery requests.

The main discovery protocols are described in *Table 1*. The picture is slightly more complex, because there is no need to do discovery on links already connected (such as a cellular connection), and the best link for autoconfiguration may not be the best link for communication.

4.2 Co-Link Configuration

Many devices already include multiple wireless interfaces to benefit from Wireless Diversity. Protocols such as P-Handoff [11] can reroute network traffic to the most appropriate link improving connectivity performance.

The goal of Co-Link configuration is to exploit the diversity of the wireless discovery capability of those same wireless links, and to always use the most "appropriate" link to perform autoconfiguration (based on client policies).

If a device needs to connect to a peer or network, it should use the wireless link with the best current discovery characteristics, by doing wireless discovery on this link. After the discovery and autoconfiguration is done, communication on only this wireless link is enabled.

Once this initial wireless link is established, it can be used to negotiate with one of the peers on this link the proper configuration for other wireless links. Wireless configurations are usually quite short (few bytes - see *Section 5.1*) ; it is often much more efficient to transmit them on the initial link than to perform wireless discovery on all the other links.

For example, if we use IrDA to perform discovery, we can download an 802.11 configuration over this IrDA link. Once the other links are configured, any appropriate vertical handoff protocol [11] can be used to migrate communication from the initial link, chosen for its discovery performance, to the link most appropriate for communication.

The device usually wants to know from the peer or the network the full list of possible links, because it's impossible to know in advance which links will match and be within range. The configuration of these extra links may be used in subsequent handoffs.

The first advantage of this technique is that it enables the device to configure wireless links where wireless discovery is not possible. The second advantage is that a link with a narrower scope can disambiguate the discovery and enable autoconfiguration without random choice or user intervention. This technique should enable faster, more secure and more power efficient autoconfiguration.

In some cases it may be difficult to know a-priori which link is the best for wireless discovery, so the device may do discovery in parallel or in sequence over a set of possible wireless links and correlate the results.

4.3 Co-Link Activation

The same Co-Link technique can also be used for power management or to reduce connection latency, through Co-Link Activation.

Wireless interfaces consume a significant amount of power, even when only waiting for incoming connections, so a device may wish to switch off interfaces not currently in use.

Unfortunately, when an interface is switched off, it ignores incoming connection requests (it is blind).

During the Co-Link negotiation procedure to select link configurations, the two devices can exchange the state of their links and negotiate activation of a subset of those links. Once the selected links are configured and activated, they may be used for communications and shut off when no longer needed.

The strategy for a device running on batteries with multiple links would be to keep only a subset of them active, most likely those with good power and discovery characteristics. Higher power links would be enabled only when necessary, through an on demand scheme like Co-Link Activation, or when the node want to use them.

4.4 Co-Link Activation vs. Scheduled Wake-up

People doing research on sensor networks and ad-hoc routing have explored schemes similar to Co-Link Activation and concluded that scheduled wake-up (waking up at regular intervals on the same link) is usually more efficient than adding a low power link to trigger wake-up [6]. Scheduled wake-up has lower hardware cost and better power saving.

However, we are dealing with a totally different usage scenario, which leads to different conclusions. The devices we consider already have multiple links, and those links already do wireless discovery. Our idea is reduce power by exploiting this existing redundancy.

Scheduled wake up has very poor performance in terms of initial discovery latency. This doesn't matter when the system is establishing long term bindings between devices. However, in our scenarios, devices don't have prior knowledge of each other, won't keep a long term relationship and the user expects a quick response. Therefore, the performance of the initial discovery is a very important factor ; this is where Co-Link Activation performs better.

If the two devices decide to keep a long term relationship, then the system should change its strategy and use scheduled wake ups (and probably using the most power efficient or longest range link).

4.5 IrDA as a configuration link

IrDA has achieved some success as a communication wireless link, but its characteristics (directionality) limit it to very short ad-hoc transactions (the "point and shoot" model). This means that IrDA is very good for pushing or getting data (at 16 Mb/s, most common data objects will transfer fast enough), but is not good for client/server requests and browsing, where the link needs to be maintained over a long period.

Fortunately, the discovery characteristics of IrDA make it ideal for use as a configuration link, used by the device to perform discovery and activate other links. The scope of IrDA is very narrow, allowing the user to physically point to device (a very natural user interface). Discovery over IrDA is low power, reliable and reasonably fast (less than 3 seconds). In addition, IrDA offers a simple form of physical security.

In Connection Diversity, our main use of IrDA is not primarily to communicate data, but to do the initial handshake with devices we want to access, in order to enable communication using a radio link (BlueTooth or 802.11). In other words, IrDA enables a "point and discover" or "point and connect" usage model.

The directionality of IrDA allows us to remove all user interface from the device itself : no button to click nor list to parse, this is replaced with a simple gesture. This is especially important on portable devices which have limited and inconvenient user interfaces, and fits the expected usage model (see *Section 3.5*).

4.6 Other link combinations

For non directional discovery, BlueTooth is a good candidate, because it has a rich discovery protocol and is found in many devices. A very low power link such as 802.15.4 [5] could also be used if it becomes widespread.

Ultrasound is usually too slow to be used for communication, but could be used as a trigger for activation.

The various 802.11 links are good candidates to be used as communication links and enabled only on demand. An 802.11a link may also be enabled based on 802.11b discovery, because 802.11b is very common.

4.7 Limitations

We don't believe that one solution fits all, and are well aware of the limitations of Co-Link. We believe that Co-link is only part of the array of techniques necessary to make wireless systems more user friendly.

This technique requires a certain amount of coordination, if devices choose to not enable any common links they won't discover each other. It is also limited by the wireless technology, a short range wireless link can't be used outside its range and therefore can't enable a long range link that would be within its own range.

5 Implementation : IrDA Configuration Point

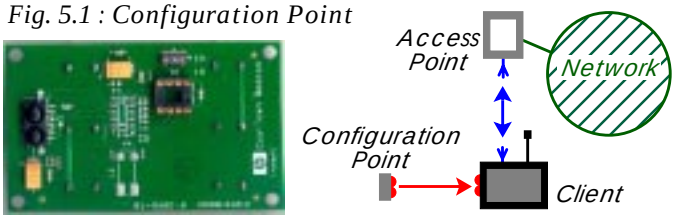
We started with a simple implementation of the Co-Link concept. The idea was to simplify 802.11 roaming. If a user goes in a cafe, hotel or airport, he needs to get connected to the local 802.11 network in a transparent and secure manner in order to access local services.

5.1 Configuration point hardware

The configuration point is a standalone device located physically in one place, it broadcasts via short range radio or infrared the configuration information for the local 802.11 network (ESSID, encryption keys...).

As part of the CoolTown project [1], we have implemented a hardware infrared beacon (see *Fig. 5.1*). It is cheap (BOM is around $1), simple, small (3 cm by 3 cm) and the battery lasts for years. It uses the lightweight IrDA Ultra

Fig. 5.1 : Configuration Point



protocol [9] to broadcast a contextual URL. The use of the Ultra protocol significantly reduces complexity and latency.

We simply modified these beacons to broadcast a 802.11 configuration. A few example configurations are :

```
<WLAN Protocol="IEEE 802.11-DS" ESSID="Coffee-Shop"
 Mode="Managed" Encryption-key="012AF56789"
 Encryption-index="1"/>
<WLAN Protocol="IEEE 802.11-DS" ESSID="My-Network"
 Mode="Ad-Hoc" Frequency="2.46GHz" Encryption-key="off"/>
```

The definition of these configurations is based on a subset of the Wireless Extension API [12], which enables it to support a wide variety of wireless LAN standards.

5.2 Client software support

Minimal client support is needed. The application to receive IrDA beacon receiver was modified to apply the received 802.11 configuration on the wireless interface and activate it. This was implemented for both Linux and WinCE.

The user only needs to bring a device in close proximity (IrDA range) of the configuration point to have it configured to access the local 802.11 network. The owner of the network may control who access his network by controlling physical access to the Connection Point.

5.3 HotSpot versus public access points

Public 802.11 access points already exist today, and users don't have trouble using them. Those access points are set to be discoverable, so the user can use 802.11 discovery. The security is usually handled by a captive HTTP portal [7] with a prearranged subscription model.

Although this type of scenario was less in need of autoconfiguration (see *Section 3.5*), the Connection Point can enable new usage models.

For example, the Connection Point can be used to distribute relatively securely tokens giving limited time access through the portal (anonymous pay per use, or as a complimentary service).

The Connection Point can also disambiguate overlapping access points or HotSpots (from two neighboring retailers). HotSpot are access points that also offer localised services to users relevant to the space in which they are [7], therefore the user wants to connect to the proper HotSpot and not just any of them.

6 Implementation : Peer to peer networking

Our second implementation was designed to address some of the limitations of the Connection Point. First, we wanted to address peer to peer scenario, which is the usage model most in need of Co-Link. Second, we wanted a generic solution, which apply to cases more general than just 802.11 configuration over IrDA. Third, we wanted to add "management" of the configuration, so we don't reconfigure an interface that is already in use.

6.1 Peer to peer : Connection Diversity

We have already established a solid framework for peer to peer connectivity [10], the main issue was mostly integrating the Co-Link concept into Connection Diversity.

Connection Diversity has been implemented for Linux, the central part of it being the Connection Manager, a daemon managing the various wireless interfaces of the system and connections to peers that support Connection Diversity.

The Connection Manager already implements Transparent TCP/IP and P-Handoff, the addition of Co-Link makes Connection Diversity a compelling solution for peer to peer connectivity.

Connection Diversity gives us two features essential to the implementation of Co-Link : efficient TCP/IP connectivity on all links and active management of all connections.

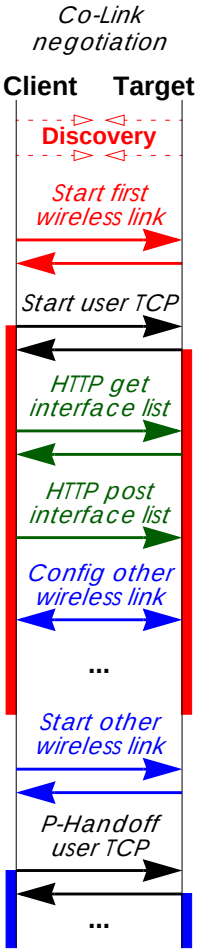


Fig. 6.2

6.2 Negotiation manager

When the Connection Manager connects to a peer (the target), it selects the link for the initial connection amongst those on which the peer has been discovered. This depends on which wireless links are doing discovery.

It then initiates the Co-Link negotiation. The negotiation between the two peers is entirely based on HTTP, so can be done on whatever link happens to be connected at that time.

The initiator of the connection requests from the target a list of interfaces and configurations, using an HTTP GET. It parses this list and tries to apply some of those configurations to its own interfaces. If setting these configuration fails, or if it realises that the target interface is off, the initiator push its own list of interfaces (modified after the configuration attempt) to the target, via an HTTP POST (see *fig. 6.2*).

Negotiation is done this way because quite often the target may be serving multiple clients or plugged in, and in those cases only the first part of the negotiation is needed. It needs to be asymmetric to avoid peer to just swap their configurations.

The interface configuration list looks like this :

```
<?xml version="1.0" ?>
<IP DNSname="mobile.hp.com" IPv4addr="213.130.63.232"/>
<InterfaceList>
  <Interface>
    <Type>IrNET</Type>
    <State>Free</State>
  </Interface>
  <Interface>
    <Type>WLAN</Type>
    <State>Off</State>
    <Config>
      <WLAN Protocol="IEEE 802.11-DS" ESSID="My-Network"
Mode="Ad-Hoc" Frequency="2.46GHz" Encryption-key="off"/>
    </Config>
  </Interface>
</InterfaceList>
```

This list is generated on-the-fly by the Connection Manager. Some configurations may be kept private. The interface state may be *Free*, *Off*, *Used* or *Busy.*

6.3 Configuration managers

A configuration manager needs to be implemented for each link layer type. It manages multiple configurations on each interface and switching between them on demand. Only one configuration can usually be active on each interface.

The configuration manager keeps track of which configurations are needed. The current policy is that if the configuration is currently used by a connection on this interface, or is acting as a backup for a connection on another interface, it is marked as needed and is active on the interface.

One of these configurations is special, the default configuration of the interface. This configuration may have the interface switched on or off. When no other configuration is needed, the default configuration is applied to the interface. Depending on the default configuration, this may switch the it off for power saving, or leave it doing wireless discovery.

The negotiation manager submits to each configuration manager incomming configurations that matches its link type (IrNET, WLAN). The configuration manager tries to find an interface for which the current configuration is not needed (for example, which has the default configuration unused). If it finds one, it just applies the new configuration to it (which may switch the interface on), otherwise it return a failure to the negotiation manager.

6.4 Current status

The negotiation manager has been implemented in the Connection Manager, and a complete configuration manager for 802.11. 802.11 configuration is done through the Wireless Extension API [12]. The IrNET configuration manager is dummy, as IrNET doesn't require any configuration [10].

Co-Link Configuration and Activation have been tested with various standard TCP/IP applications in Linux. The IrDA link is used for negotiating 802.11 setup. We can also use 802.11 to negotiate 802.11 setup, but this will be useful only when 802.11a drivers are be available for Linux.

Co-Link is integrated with Transparent TCP/IP and P-Handoff, so the connection can be automatically re-routed on the newly activated interface and any loss of the initial link is totally transparent to the user. The Co-Link negotiation happens after the initial TCP/IP link is established, in parallel with the user transaction, so is not noticed by the user (and is fast anyway on most link layers).

One of the problems with the current 802.11 hardware is that it's impossible to bypass 802.11 scanning, therefore we don't get the full benefit of this approach (lower latency). Hopefully, hardware will become more flexible in the future.

Our current policies are crude and need to be enhanced for real deployments. We only manage peer to peer Co-Link, and we still need to integrate infrastructure discovery support.

The current implementation does not include support for BlueTooth. We are considering adding this.

7 Conclusion

Wireless usage is limited by the complexity of wireless technology. Current wireless discovery and autoconfiguration techniques don't fully address the peer to peer scenario. Users need simpler and more intuitive means to discover networks and devices they can connect to.

The proposed Co-Link technique takes advantage of wireless diversity, it uses the wireless links with the best discovery characteristics to configure and enable other wireless links that may have better communication characteristics. This usually can achieve more accurate, faster and lower power discovery than traditional techniques.

We have implemented a simple IrDA Connection Point to demonstrate the concept and that can be used with HotSpots.

We have also implemented a more generic peer-to-peer Co-Link within our Connection Diversity framework, with complete configuration negotiation and management, based on HTTP and seamless integration with peer-to-peer vertical handoffs.

8 References

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

[2] D. Caswell and P. Debaty. *Creating web representations for places*. Proc. of HUC'2000, Bristol, England, 2000.

[3] J. Haartsen, M. Naghshineh, J. Inouye, O. J. Joeressen and W. Allen. *BlueTooth: Vision, Goals, and Architecture*. ACM Mobile Computing and Communications review, Vol. 2, No. 4, (October 1998).

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

[5] J. A. Gutierrez et al. *IEEE 802.15.4: A Developing Standard for Low-Power, Low-Cost Wireless Personal Area Networks*. IEEE Network, vol 15, no.5.

[6] M. Nosovic and T.D. Todd. *Scheduled Rendezvous and RFID Wakeup in Embedded Wireless Networks*. Proc. of ICC 2002.

[7] P. Reddy, V. Krishnan, K. Zhang and D. Das. *Authentication and Authorization of mobile clients in public data networks*. HP report HPL-2002-213.

[8] D. Das, G. Manjunath, V. Krishnan, P. Reddy. *HotSpot! - a service delivery environment for Nomadic Users System Architecture*. HP report HPL-2002-134.

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

[10] Jean Tourrilhes, Luiz Magalhaes & Casey Carter. *On-Demand TCP : Transparent peer to peer TCP/IP over IrDA*. Proc. of ICC 2002.

[11] Jean Tourrilhes & Casey Carter. *P-Handoff : A framework for fine grained ad-hoc vertical handoff*. Proc. of PIMRC 2002.

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