A Wireless Public Access Infrastructure for Supporting Mobile Context-Aware IPv6 Applications

Adrian Friday[1], Maomao Wu[1], Stefan Schmid[1], Joe Finney[1], Keith Cheverst[1] and Nigel Davies[1,2]

[1]Distributed Multimedia Research Group,
Computing Department, Lancaster University,
Bailrigg, Lancaster,
LA1 4YR, UK

[2]Computer Science Department
Gould-Simpson Building
University of Arizona
Tucson, AZ

{adrian,maomao,sschmid,joe,kc,nigel}@comp.lancs.ac.uk

ABSTRACT

This paper presents a novel wireless access point architecture designed to support the development of next generation mobile context-aware applications over metropolitan scale areas. In addition, once deployed, this network will allow ordinary citizens secure, accountable and convenient access to the Internet from their local city and campus environments.
The proposed architecture is based on an approach utilising a modified Mobile IPv6 protocol stack that uses packet marking and network level packet filtering at the edge of the wired network to achieve this objective. The paper describes this architecture in detail and contrasts it with existing systems to highlight the key benefits of our approach.

Keywords

Public Access Point, Wireless Internet, Security, Authentication, Mobile IPv6

1. MOTIVATION

Recent years have witnessed the development of a number of prototype location based services, such as the GUIDE system [4,7], CyberGUIDE [13], or the HIPS tour guide [3], that aim to deliver information (such as reminders or tourist information) to mobile users as they roam throughout a geographical area. Indeed, access to the Internet and the provision of location based services is an area predicted to offer significant market impact in the near future.

At Lancaster we have dedicated ourselves over the past four years to investigating this application domain through the development and continued refinement of a context-aware tour guide system called GUIDE [5]. The development of the GUIDE system is the culmination of a plethora of smaller activities including: situated requirements gathering and analysis; deployment of a dedicated wireless access network; modelling and data capture concerning the local city and its associated attractions; and the iterative development and evaluation of the GUIDE application and supporting distributed system itself.

1st Workshop on Wireless Mobile Internet 7/01 Rome, Italy
© 2001 ACM ISBN 1-58113-423-1/01/07...$5.00

The current GUIDE system allows visitors to Lancaster to equip themselves with a dedicated guide unit from the local tourist information centre. So equipped, a tourist can develop a personal profile reflecting their interests, dietary requirements and preferences. This profile is then used by the system, together with dynamic information such as the user s current location or trail, to enhance their visit to the city with a wealth of individualised information and advice. A GUIDE user may find out about the city and its attractions, have personalised tours constructed that match their interests, be guided between attractions, re-orient themselves if they find they have become lost or use a range of simple interactive services (such as messaging, ticket booking, reservation services etc.).

More recently, in the GUIDE II project [10], we have begun to investigate the potential to evolve GUIDE farther by promoting the sense of community among users of the system (e.g. developing mechanisms whereby users can be made aware of the actions, views and recommendations of other users). For instance, a user might be made aware that another user of the system is sitting in one of the caf s that has been recommended to them on their itinerary and that they may be prepared to offer a personal view on that particular location [6]. Furthermore, recommender systems can be constructed that gather information about collective user behaviour which is then used in a suitably anonymised form to inform the user s decisions, e.g. *people with a similar profile to you prefer these caf s* . Such systems are analogous to the recommendation systems employed by on-line shopping systems such as Amazon.com, but offer the potential for interaction between users (clearly such a system must respect users wishes for privacy and anonymity).

Whereas the GUIDE system has deployed a bespoke client end-system, the other drive of GUIDE II is to open up the system to citizens of Lancaster such that they might use a range of new and existing applications on their own personal computing devices (laptops, PDAs, cell-phones etc.). As a side effect of offering

connectivity to citizens, we hope to encourage the active involvement of a wider community of users in our mobile systems research.

In the remainder of this paper we present an architecture that aims to address this application domain. More specifically, such as system must:

- Be simple and convenient for potential users (ease of installation, continued ease of use)
- Offer fine grained access control and accounting for the service provider
- Offer reasonable levels of security and authentication (such that users can trust the system and the system is not vulnerable to exploitation)
- Support unmodified use of legacy Internet applications
- Offer scalability (both in terms of number of users and extensibility of the system to cover the metropolitan area)

In the next section we discuss an architecture that we believe meets these objectives and provides a scalable and secure public access network. It is our intention that the architecture presented in this paper will act as a blueprint; providing recommendations for others wishing to deploy metropolitan area wireless public access networks, such as ours. Section 3 discusses the current implementation status and highlights the main architectural choices. In section 4 we present our concluding remarks.

2. PUBLIC ACCESS ARCHITECTURE

The public access architecture proposed in this paper draws on three individual research initiatives at Lancaster: the existing GUIDE wireless infrastructure, the LARA++ active router initiative [17] and the ongoing research into Mobile IPv6 [8,12] (which has resulted in the development of Mobile IPv6 stacks for both Linux and Windows 2000 Professional).

Significantly, our work with GUIDE has given us some experience of developing public access systems and will serve as a prototype environment for our new architecture. Once implemented and refined, we plan to integrate the wireless infrastructure with the recently announced Mobile IPv6 testbed collaboration between Cisco Systems, Microsoft Research and Orange [14].

Our work on programmable networks (the LARA++ active router architecture) provides us a flexible and dynamic deployment platform upon which to develop our network services. LARA++ is a programmable component based active router architecture that supports the dynamic instantiation of code in a secure runtime environment. We believe that by developing our public access system as a set of active router components we will be able to more rapidly deploy and evolve services within the network.

In the remainder of this section we consider the current GUIDE access network and how this should be evolved to meet the new GUIDE II objectives.

2.1 Existing GUIDE Infrastructure

The existing GUIDE wireless infrastructure is based on a network of 2Mbps IEEE 802.11 compliant cells (Lucent Technologies ORINOCO system) as shown in figure 1 below.

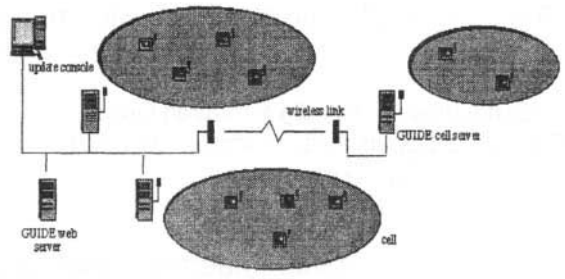

Figure 1 — The existing GUIDE wireless network infrastructure

Cells are linked back to the campus backbone (and hence each other and the Internet) via an arrangement of Symmetric DSL leased lines, microwave and point-to-point 802.11 links (in some cases, these links are provided by an existing educational wireless network established by the EDNET initiative which links local schools throughout the region to the Internet via our campus network).

Each cell is controlled by a multi-homed cell server (a standard low-specification PC, currently running Linux) supporting two network interfaces: one to the cell (an ORINOCO card with extended antenna); and the link back to the campus backbone. Cell servers are located in private areas of public or University owned buildings and are contained in physically secured cabinets.

In protocol terms, the cell architecture is currently based on IPv4 and does not support access to systems, users or applications outside of the GUIDE system. In more detail:

1. Each wireless cell is a class-A IPv4 subnetwork (GUIDE mobile stations have unique ten-dot class-A addresses). Cells are thus isolated from each other at the network layer. End-systems are individually addressable within a cell, but not across cells via IP.
2. Cell servers have standard campus class-B IPv4 addresses and are otherwise standard addressable Internet hosts. Campus backbone traffic and traffic within the wireless cell is not bridged by the cell server in the protocol stack (i.e. it does not function as either a router nor a link layer bridge).
3. Traffic to the Internet is bridged at the Application Layer by the GUIDE system. There is currently no scope for non-GUIDE applications to route IP traffic from the mobile systems to the Internet and GUIDE is thus able to have total control over access to the network.
4. Each cell server contains a proxy-server that uses HTTP over a multicast-UDP based on a cyclic schedule of

broadcast pages to disseminate information to users in the cell. Clients may request pages (e.g. from the Internet) using the GUIDE browser, which are subsequently incorporated into the broadcast schedule. The broadcast protocol is one of the key mechanisms for promoting the scalability of GUIDE, since by reference locality co-located clients are likely to require similar parts of the information base or underlying object model at specific locations. Moreover, the information base is dynamically scoped by the caching strategy of the proxy-server as it adapts to client requests.

5. Every cell has its own unique identifier that is used by clients to determine when a handover between cells has occurred. Handover is thus also an application layer issue in the current system.

While this architecture is clearly well suited to the original intentions of GUIDE, it presents a number of problems if we are to succeed in our objective of opening up the system for public access. For instance, clients are not individually addressable from within IP (and thus from within other applications in the Internet). In addition, all applications must be modified to interact using the GUIDE protocol suite (which is clearly an unacceptable restriction in the general case).

In order to support generalised public access within our network one possibility would be to turn each wireless cell into a specific subnetwork, enable routing at the cell servers and utilise Mobile-IP to address the roaming issues. However, such an approach would be inherently limited by the lack of available IPv4 addresses (although network address translation could be used to partially address the limitations of the IPv4 address range, such an approach would require modifications to mobile IP and add unnecessary additional complexity).

As a consequence, we have decided to base our public access strategy on the emerging Mobile IPv6 standard. In the following section we describe our architecture in more detail.

2.2 The Proposed Public Access Architecture

The public access architecture that we are deploying within the local city is illustrated in figure 2. We assume a network topology consisting of a wireless access network consisting of 11Mbps IEEE 802.11b compliant cells (each configured as an independent subnetwork) connected to a trusted core network. The core network is guarded from unprivileged access by access routers (the access routers effectively replace the cell servers of the GUIDE system). Access routers are in turn connected to a common gateway that links back to the campus backbone and from there to the Internet.

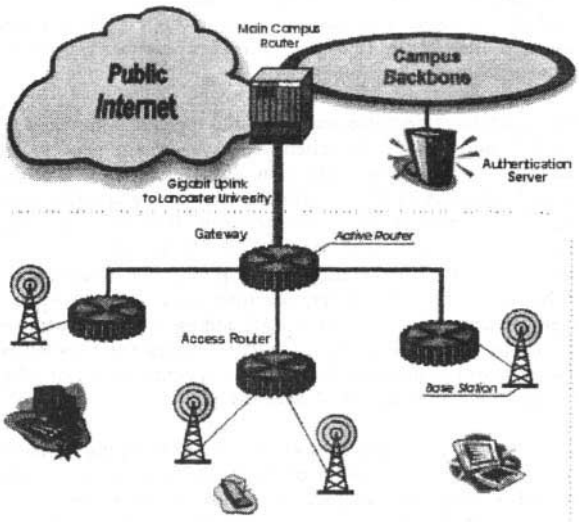

Figure 2 — The proposed public access architecture

Access to the Internet is controlled by two principles: packet marking and packet filtering[3]. IPv6 datagrams are tagged on the client end-system using a modified Mobile-IPv6 protocol stack. Access routers ensure that only packets containing a valid access token are allowed onto the trusted core network. In addition, access routers will only forward traffic originating from the core network (or the Internet) to clients that hold current valid credentials. Tokens are generated and disseminated throughout the system when clients interact with a central authentication server. The mechanisms behind this process are described in more detail in the following sections.

2.2.1 User Authorisation

In order to use our system the user will require our modified Mobile IPv6 protocol stack and the creation of a set of credentials. The user s credentials (username and password) are created and stored with the authentication server by the system administrator. We assume that at this time, the user is assisted in downloading, installing and configuring the protocol stack on their computer.

The user is assigned an access group (which may be a singleton, if the user has unique requirements). The access group permits differentiation of clients in terms of levels of service (QoS for example) and may place limitations on users access to the network (particular cells, specific times of day, durations or frequencies of use, or payload volume or data rate based limitations etc.).

Each time a user wishes to use the network (e.g. turns on their device in a cell), the protocol stack requires a valid token in order

[3] This mechanism is similar to that used in the Microsoft CHOICE system [2] and offers increased protection (e.g. against MAC address spoofing) over approaches that do not require modifications to the client software (e.g. Stanford s SPINACH [16]).

to be able to tag packets for transit via the access router[4]. At this juncture the user is prompted to enter their username and password interactively (this process occurs once per session and the credentials are otherwise cached).

The credentials (username and password) together with the MAC and IP addresses of the client end-system are sent to the authentication server for verification. The MAC address is part of the authentication to link a particular user with a specific set of end-systems, making using stolen credentials more complicated (the attacker would need to spoof the MAC address of the system). The IP address is used to track end-system location at the application level. Again, at any one time a user may only have a single IP address for each valid MAC address registered with the system. In Mobile IPv6 this limitation implies that a potential attacker must remain co-located with the system they are spoofing in order to minimise the risk of detection.

To avoid masquerading attacks based on snooping a user s credentials or impersonation of the authentication server, the payload is encrypted using the public key of the authentication server. Upon entering a cell the client is only allowed to communicate with the authentication server in order to obtain a valid token, all other traffic is dropped by the access router.

2.2.2 Token Generation

When a valid user successfully authenticates with the system a token is generated by the authentication server and returned to the client. The token is simply a pseudo-random integer that is unique to a given session. The token is used to tag packets belonging to the client for passage through the active router. To avoid other clients snooping the token, the authentication server encrypts the payload using the session key passed by the client (generated and transmitted with the user s credentials at authentication time). Optionally, the system may be configured to encrypt the entire client payload for enhanced security over the wireless hop to the access router[5].

To avoid brute force attacks on the client s token, all tokens have a fixed lifetime. Beyond this interval the client must re-authenticate using the user s cached credentials to obtain a new token. The re-authentication interval is chosen to reflect the size of the token (currently 10 minutes and 32 bits respectively).

2.2.3 Packet Tagging

When a packet passes through the modified Mobile-IPv6 stack on the user s end-system it is tagged with a new IPv6 first-hop extension header containing the token, a checksum and some housekeeping fields (including protocol version etc.). The token and checksum are both encrypted using the session key. The checksum is added to prevent a potential attacker from simply snooping the extension header and adding it to their own data[6].

Since the extension header must always have this level of encryption, a symmetric cipher is used to avoid the performance overhead of public key cryptosystems.

2.2.4 Packet Filtering

By default, packets must be filtered bi-directionally at each access router. Every access router maintains an access control list (ACL) indexed by MAC address that is used to validate packets to or from each client in the wireless cell. When each user successfully authenticates, a copy of the valid token and session key is distributed to the access router that forwarded the client s authentication request to update the ACL. The MAC and current IP address of the end-system is sent with the token and key to enable enforcement of the mapping of credentials to end-systems within the cell.

When a packet is received, the tag header is removed, decrypted using the session key and its contents validated against the ACL. Any packet that presents an invalid or expired tag, or contains no tag, is dropped. One exception to this rule is that when a host first transmits in a cell they are permitted to contact certain well-known IP addresses; this interval allows hosts the opportunity to contact the authentication server.

Each entry in the ACL is valid for a particular *expiration time* (the token refresh interval is chosen to be smaller than the expiration time). Beyond the expiration time a host must obtain a new valid token (which indirectly refreshes the ACL at the access router).

The maintenance of soft state at the access routers allows for end-systems to be limited to certain cells or time windows, or even be blacklisted from accessing the network. Other possibilities for extending this scheme are to allow for traffic accounting and monitoring or the introduction of billing. We plan initially to include elementary traffic shaping and monitoring code to stop malicious users from mounting denial of service or flood attacks on the authentication servers or access routers (e.g. monitoring and controlling the data rate of traffic between each client and the authentication server).

2.2.5 Failure Recovery

In the event that an access router misses an ACL update from the authentication server or crashes and reboots losing all state information, the access router will erroneously stop forwarding traffic. To support recovery in these situations we are planning to introduce one of two mechanisms; either i) upon discovering a client with an unrecognised tag, the access router contacts the client triggering immediate re-authentication, or ii) the access router forwards the tag from the client to authentication server and receives an ACL update if the credentials are valid (or a negative acknowledgement if it is to refuse the client access to the network).

2.2.6 Handover Support

In a network such as ours, we are concerned to minimise the

[4] We assume that this process is an augmentation of the IPv6 auto-configuration protocol that is instantiated when the client enters a new cell.

[5] This offers an alternative to the IEEE 802.11 wired equivalent privacy (WEP) protocol, which has been shown to be vulnerable to attack [1].

[6] Although appreciably weaker than a 1-way hash function such as MD-5 or SHA-1, the checksum seems to offer a good compromise for size against resistance to attack.

impact of our access strategy on hosts as handover occurs between cells. In particularly, as a client end-system hands over to a new cell it must immediately re-authenticate; we aim to reduce the impact of packets dropped by the new access router during the authentication period. Such losses would introduce unwanted jitter in continuous media applications and would be interpreted incorrectly as congestion by TCP.

In order to address this issue we propose two enhancements to our architecture. Firstly, we intend to introduce a brief *reprieve time* as a client enters a cell. During the reprieve time an existing authenticated client from another cell is allowed to continue using the network while the authentication takes place (i.e. all the client s credentials *except* the IP address are validated at the access router). However, if the authentication request or ACL update messages are not forthcoming, the client s traffic is blocked as normal. Note that the reprieve time interval may allow a node to extend the lifetime of its token slightly since sufficient time must be given to the client to successfully authenticate upon entering the new cell. To avoid malicious users from exploiting this feature, for example by repeatedly changing MAC address, the access router should monitor MAC to IP address mappings. Note that correspondent frequent changes of IP address are self-limiting as they typically break the end-to-end semantics of most network applications.

The second improvement, which works in tandem with the reprieve time, is to introduce the active dissemination of ACL updates to neighbouring cells of a given mobile. In more detail, when a client authenticates and the ACL of the active router is updated to reflect the validity of the client, the update message is sent instead to a group of access routers that include the immediate neighbouring cells. When the client roams, a valid token and session key pair have already been distributed to the access router and the clients traffic can be legitimately forwarded until the next refresh interval. In practice, we still plan to force the client to re-authenticate upon entering the new cell. The group of routers to disseminate the token to can be adjusted at runtime based on observation of the mobile s roaming behaviour.

2.2.7 Securing the Core Network

Our public access architecture has been designed based on the principle that the core network is difficult to attack (the access routers are physically protected by locked cabinets in areas of buildings not opened to the general public). However, our system is vulnerable if a potential attacker can gain access to this network. To increase the security of the core without utilising heavyweight authentication or encryption schemes, we plan to use the gateway (see figure 2) as a firewall. This gateway will only accept traffic originating from the access routers whose MAC addresses are stored in its access control list[7]. Furthermore, the gateway is able to monitor for the absence of public access accounting and authentication traffic that should be associated with user datagrams originating from a known access routers MAC address (further complicating spoofing attempts).

We would recommend a stronger approach (e.g. distributing keys to access routers and encrypting traffic using IPSec between the core network components) if our scheme were to be deployed over an untrusted network (such as the Internet). This security could be trivially added to the system using a similar mechanism as proposed for the client end-systems, i.e. forcing the access routers to authenticate with the authentication server and distributing session keys to encrypt all forwarded traffic. Token based filtering could be used by the firewall to protect the remainder of the network from unauthenticated traffic. A summary of the security features of our architecture is presented in table 1.

Potential Attack	Protection Mechanism
No Credentials	All traffic except authentication traffic dropped by the Access Router (no token present or host not in ACL).
Stolen Credentials	Modified IPv6 Stack passes unique random token, MAC and IP addresses with data.
Invalid or Expired Credentials/ Cell based Access Control	The token and session key have a limited lifetime and must be refreshed periodically.
Stolen IP Address	The MAC Address of client is passed with the token to the Access Router for validation.
MAC Address Spoofing/ Rapid cycling of MAC addresses	A 1:1 MAC to IP address ratio is maintained and monitored at the Authentication Server.
Masquerading attacks on authentication	Authentication traffic is encrypted with the public key of the Authentication Server.
Stealing a client s token	The token is never passed in cleartext over the network.
Brute force attacks on client s token or session key	Token and session key are only valid for one re-authentication interval.
Attacker steals encrypted tag	Encrypted checksum within the token is extremely unlikely to match the attacker s payload.
Masquerading as Access Router	All MAC addresses are registered with the firewall router and physically protected.

Table 1 — Summary of security features in the GUIDE II access network architecture

3. SYSTEM IMPLEMENTATION

Implementation of the components of our public access architecture is currently underway. In this section we briefly discuss our current implementation status and outline our preliminary choices for the various cryptographic components that comprise the system.

3.1 Client Protocol Stack

We have chosen to use IPv6 as the starting point for a client protocol stack. IPv6 offers us a number of advantages including increased address space, simple extension header parsing and powerful integrated services such as mobility support, anycast and IPSec[8]. Anycast in particular will allow us to replicate the authentication server behind an anycast address to increase availability and redundancy. In addition, we have considerable experience with IPv6 and have implemented Mobile IPv6 for Linux and Windows 2000 during previous projects [8,12].

The user authentication protocol is based on a lightweight request/ response protocol (e.g. a UDP or ICMPv6 datagram). The request is secured using IPSec public key encryption based on RSA [11] public key mechanism. The public key of authentication server is pre-configured into each protocol stack at installation time

[7] The approach is similar to the remote configurable VLAN switch of CMU s Netbar system [15] and the intelligent hub of UC Berkeley s public access system [20].

[8] Since designing our protocol the requirement to support IPSec in Mobile IPv6 has been dropped as part of the standardisation process. We may have to introduce our own end-to-end encryption to address this issue in the future.

(avoiding the need for a key distribution infrastructure[9]). This approach has two main benefits; firstly, the user authentication protocol exchange is protected from snooping, and secondly, impersonation of the authentication server would require knowledge of the corresponding private key. We currently plan to encrypt the response from the authentication server to the client using triple-DES [18].

As highlighted earlier, the protocol stack also carries out the packet marking, including the most recent access token into every packet as an extension header. To prevent MAC address spoofing and replay attacks, we encrypt the access token using the shared session key along with a packet checksum. We plan to use the high performance block cipher TEA (Tiny Encryption Algorithm [21][10]) to minimise the latency due to encryption of the packet tags. The 32-bit checksum can be taken from dynamic protocol fields of the IPv6 header: source and destination addresses, flow id and payload length.

3.2 Authentication Server

The authentication server runs as an application level program on a well-known host and port. The authentication program manages user accounts and is responsible for processing incoming UDP authorisation request datagrams. On success, the user authentication application sends the encrypted response datagram back to the client and triggers the dissemination of the ACL update message to the access router(s). The tracking of user location, motion prediction and formation of the router dissemination groups are the responsibility of a separate application (to reduce load on the authentication program).

The authentication server requires a standard IPv6 protocol stack with support for IPSec (in order to support the secure authentication mechanism). We plan to use UDP datagrams for the authentication dialogue instead of TCP as the amount of data exchanged is very small and does not warrant the overhead of establishing separate TCP sessions. Reliability is to be achieved through a simple client driven retransmission strategy.

3.3 Access Router and Gateway

As stated in section 2, we plan to base our access routers and the gateway firewall on the LARA++ active router architecture [17]. We have chosen to implement our access routers upon a substrate of active routers to facilitate rapid deployment of new services (such as updates to the packet filtering components, QoS support, billing or in-circuit diagnostics). In addition, our public access network will offer an interesting application domain and test-bed environment for future active router research.

In the initial configuration, the packet filtering and access control list management will be implemented as dynamic components instantiated into the access routers. The gateway router will include a number of active components that attempt to secure the access network from malicious external nodes. These components will try to detect denial-of-service attacks (for example, ping floods) by external nodes based on packet analysis (i.e. packet types, source addresses, data rate etc.).

4. RELATED WORK

In the design of our architecture we have drawn on the experience of earlier public access point research. More specifically, the MAC source address filtering of our gateway router is similar to the remote configurable VLAN switch of the CMU NetBar system [15] and the public access system of UC Berkeley [20]. The packet tagging and filtering concept is similar to the PANS mechanism employed by Microsoft s CHOICE system [2]. Unlike Stanford s SPINACH [16], Berkeley and CHOICE we do not rely on DHCP. More importantly, we do not require any non-standard modifications to the protocols or services on the network once the wireless hop has been negotiated. Table 2 presents a summary of how our architecture compares to the SPINACH and CHOICE systems.

Again, unlike SPINACH, we have chosen not to use the popular Kerberos [19] authentication service (although our approach is based on a number of similar concepts) since we need to provide rapid handover performance and subtly different semantics. More specifically, authenticating the client and accessing the service (the network) is achieved in one protocol exchange. Moreover, the authentication server (which effectively contains the ticket granting service of Kerberos) distributes valid token information to both the client and the access router group pertaining to that client. These features of our authentication protocol are designed to speed up handover performance significantly.

We believe that our architecture will provide a high level of security and authentication (at least as good as competing systems). In addition, our architecture will provide fine grained access control, accounting and monitoring, enabling us to implement a wide range of access policies and police for potential abuses of the system. The use of IPv6 and the choice of lightweight mechanisms and high performance cryptographic protocols should allow the system to scale to a moderate number of users without significant performance degradation. Moreover, IPv6 offers us a sufficient address space to allow us to establish each cell as an independent subnetwork; this further enhances efficient and scaleable routing.

	SPINACH	CHOICE	GUIDE II
Special Hardware	No	No	No
Address Allocation	DHCP	DHCP	Mobile IPv6
Authentication	WebLogin, web interface to Kerberos	MS passport	Authentication server
Authorisation	SPINACH router	PANS server	Authentication server
Verification	SPINACH router, modified Linux kernel	PANS server, PANS miniport driver	Access router, active router, active component
Special client software	No	PANS driver and user module	Modified mobile IPv6 stack
Security	(MAC, IP)	(key_id, key, token)	(MAC,IP,key,token)
Levels of security	No	Yes	Yes
Different access policies	No	Yes	Yes
Mobility support	No	Yes (beaconing scheme)	Yes (mobile IPv6)
Location information tracking	No	No	Yes
Handoff support	No	No	Yes
Public key encryption	SSL	SSL	IPSec

Table 2 - A comparison of the GUIDE II public access system against SPINACH and CHOICE

[9] Currently absent from the Mobile IPv6 IPSec specification.

[10] There have been no known successful cryptanalyses of TEA, it is purported to be at least as secure as the well-known IDEA cipher and is cheap to compute.

Finally, our system will allow unmodified IPv6 applications to run over the public access infrastructure. Work is in progress on developing an IPv4 within IPv6 mapping component that enables unmodified legacy IPv4 applications to run over IPv6 networks (and hence over our public access network) [9].

5. CONCLUSIONS AND FUTURE WORK

In this paper we have introduced a potential blueprint for developing a public wireless access network that need to scale to cover metropolitan areas. Furthermore, within the context of GUIDE II, this network will enable the continued experimentation with context-aware applications and provide rich opportunities for context-sharing and community enhancing applications.

In addition, we believe our network architecture to offer a number of important distinguishing features from other related approaches, namely:

1. The use of soft-state based authorisation at the edge of the wired network. This offers a high level of security, as secret credentials are re-issued at a configurable interval (reducing the risk of brute force or spoofing attacks).

2. Configurable levels of security over the wireless link. Users/ administrators may choose between encryption of the access credentials (spoofing protection) and full IPSec encryption (providing privacy).

3. Support for continuous media or streaming applications while roaming. The short reprieve time and predictive token distribution algorithms minimise latency due to authentication during the handover process.

4. Scalability; due to the use of network layer routing afforded to us by the large address space of IPv6 and defining each cell as its own network.

In addition, our approach is novel for a number of technical reasons including the public use of Mobile IPv6 to support public access and the use of active router technology (LARA++) to speed the development and continued refinement of the system.

ACKNOWLEDGEMENTS

This work has been carried out as part of the EPSRC funded GUIDE II project (GR/M82394) with considerable cooperation from Lancaster City Council. The project also has received support from Cisco Systems, Microsoft, Orange, HP Labs (Bristol) and Lucent Technologies.

REFERENCES

[1] Arbaugh, W., N. Shankar, and Y.C.J. Wan, Your 802.11 Wireless Network has No Clothes , in *Technical Report*, Department of Computer Science, University of Maryland, College Park, Maryland 20742.

[2] Bahl, P., A. Balachandran and S. Venkatachary, The CHOICE Network — Broadband Wireless Internet Access in Public Places , MSR-TR-2000-21, February 2000.

[3] Broadbent, J., and Marti, P., Location Aware Mobile Interactive Guides: usability issues , in *Proceedings of the Fourth International Conference on Hypermedia and Interactivity in Museums (ICHIM97)*, Paris, September 1997.

[4] Cheverst, K., Davies, N., Mitchell, K., and Friday, A., The Role of Connectivity in Supporting Context-Sensitive Applications , in *Lecture Notes in Computer Science No. 1707*, Springer-Verlag, (1999), 193-207.

[5] Cheverst, K., et. al., Developing a Context-aware Electronic Tourist Guide: Some Issues and Experiences , in *Proceedings of CHI 00*, Netherlands, pp 17-24, April 2000.

[6] Cheverst, K., G. Smith, K. Mitchell, A. Friday and N. Davies. "The Role of Shared Context in Supporting Cooperation between City Visitors". To appear in COMPUTERS & GRAPHICS, Vol. 25, No. 4, 2001.

[7] Davies, N., K. Cheverst, K. Mitchell, A. Friday, Caches in the Air: Disseminating Information in the Guide System , in *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '99)*, New Orleans, Louisiana, U.S., 25-26 February 1999.

[8] Finney, J. and A. Scott, Implementing Mobile IPv6 for Multimedia , Proceedings of 1st GEMISIS symposium on Multimedia Network Technology, Salford, UK, May 1998.

[9] Finney J and G. O Shea, Mobile 4-in-6: A Novel Mechanism for IPv4/v6 Transitioning , Submitted to IDMS 2001.

[10] Guide II: Services for Citizens , Research Project, Lancaster University, EPSRC Grant GR/M82394, 2000.

[11] Kaliski, B. and J. Staddon, PKCS #1: RSA Cryptography Specifications , IETF Internet RFC 2437, October 1998.

[12] LandMARC, Research Project, Lancaster University, available via the Internet at http://www. landmarc.net/, October 1999.

[13] Long, S., Kooper, R., Abowd, G.D., and Atkeson C.G., Rapid Prototyping of Mobile Context-Aware Applications: The Cyberguide Case Study , in *Proceedings of 2nd ACM International Conference on Mobile Computing (Rye NY, 1996)*, ACM Press.

[14] Mobile IPv6 Testbed, Collaboration with Cisco, Microsoft and Orange, Lancaster University, February 2001, http://www.mobileipv6.net/testbed

[15] Napjus, E.A. NetBar - Carnegie Mellon s Solution to Authenticated Access for Mobile Machines , CMU White Paper, http://www.net.cmu.edu/docs/arch/netbar.html

[16] Poger, E. and M. Baker, Secure Public Internet Access Handler (SPINACH) , Proceedings of the USENIX Symposium on Internet Technologies and Systems, 1997.

[17] Schmid, S., J. Finney, A. Scott, D. Shepherd, Component-based Active Network Architecture , *submitted to 6th IEEE Symposium on Computers and Communications (ISCC 01)*, Hammamet, Tunisia 3-5 July, 2001,

[18] Schneier, B. Applied Cryptography , Second Edition, John Wiley & Sons, New York, NY, 1995, ISBN 0-471-12845-7.

[19] Steiner, G., B.C. Neuman and J.I. Schiller, Kerberos: An Authentication Service for Open Network Systems , In Proceedings Winter 1988 Usenix Conference, February, 1988, http://web.mit.edu/kerberos/www/papers.html

[20] Wasley, D.L. Authenticating Aperiodic Connections to the Campus Network , June 1996, http://www.ucop.edu/irc/wp/wpReports/wpr005/wpr005_Wasley.html

[21] Wheeler, D. and R. Needham, TEA: a Tiny Encryption Algorithm , in *Technical Report*, Computer Laboratory Cambridge University, England.