Smart Edge Server – Beyond a Wireless Access Point

G. Manjunath, T. Simunic, V. Krishnan,

J. Tourrilhes, D. Das, V. Srinivasmurthy, A. McReynolds

Hewlett Packard Labs

1501 Page Mill Rd.

Palo Alto, CA 94304

{geetham, ddas,venuks}@india.hp.com {tajana, venky, jt, allanm}@hpl.hp.com

ABSTRACT

Wireless access at cafes, airports, homes and businesses have proliferated all over the globe with several different Wireless Internet Service Providers. Similarly, digital media has created a paradigm shift in media processing resulting in a complete change in media usage models, revamped existing businesses and has introduced new industry players. We believe there is a tremendous opportunity for application and system services at the intersection of the above two domains for exploiting the wireless connectivity to provide ease of use in handling media. In this paper, we propose a feature-rich, secure wireless service delivery framework over enhanced public access points (called Smart Edge Servers), which provides the right platform for deployment of specialized services to the mobile users. The Smart Edge Server provides secure wireless access to the clients, has sophisticated media handling and storage capabilities and uses advanced techniques to manage resources available to it, such as bandwidth, power and the type of connectivity. A prototype implementation of our Smart Edge Server has been built that implements all the features discussed above.

Categories and Subject Descriptors

C.2.5 [**Computer-communication networks**]: Local and Wide-Area Networks – *Internet, Wireless.*

General Terms

Management, Measurement, Performance, Design, Security.

Keywords

access point, management, media, security, wireless, low-power

1. INTRODUCTION

Wireless network access points, known as hotspots, are proving to be a cost effective and viable solution for ubiquitous access to the Internet **[1]**. This has led to an explosive growth of wireless access points at cafes, airports, homes and offices. Typically wireless service vendors offer only basic connectivity to the Internet with minimum provision for secure communication and no provision for management of resources, such as battery lifetime on the mobile device. Commercial wireless gateways available from Orinico, BlueSocket, and NetMotion provide only secure physical access to the Internet. The HP ProCurve series of wireless access point products provides campus-wide security with support for seamless roaming across the physical network and accommodates precise network access control. The HP Wireless Connection Manager, used in the popular alliance with Starbucks, is a free software application that automatically detects, connects, and facilitates user mobility across different high-speed wireless networks. A number of researchers have proposed various methods to seamlessly handoff between the wireless networks at both macro and micro level **[19]**, **[20]**, but most have not taken mobile's power limitations and QoS needs of multimedia into account.

Concurrently there has been a paradigm shift in multimedia processing caused by digitized media. For example, the digital camera experience is very different from the traditional film camera – there are no consumables such as film rolls, job of film processing shifted to the owner of the camera, etc. In general, when it comes to handling media, the user's expectations are much like that for just another appliance – the digital media devices should just work. In addition, there has been record growth in media downloading and sharing **[18]**. There is more and more demand for secure and easy media access, with a significant fraction of it being personal media that is shared. This has resulted in introduction of new industry players, revamping of existing businesses and a complete change in usage models.

There is a tremendous opportunity for application and system services at the intersection of the above two trends. Wireless Access points, by leveraging their placement at the edge of a content delivery network, are very suitable for deploying content-based, customized services and enable special system services for the mobile client. This niche has not yet been captured by any service provider due to lack of an appropriate software infrastructure over these wireless access points. In addition, because access points deliver content to multiple clients in their environment over potentially a number of different wireless standards (e.g. WLAN, Bluetooth, GPRS), they are in the best position to carefuly manage client's resources such as battery lifetime and mobility.

WMASH'04, October 1, 2004, Philadelphia, Pennsylvania, USA.

In this work we present a secure wireless service delivery framework over enhanced wireless access points which provides the right platform for deployment of specialized services and applications to mobile users. We call these enhanced, feature-rich access points Smart Edge Servers (SeS). An SeS has several new features that are above and beyond what today's wireless access points provide, such as content adaptation, energy saving, seamless migration across different wireless links, consolidated access to distributed personal media and secure access to services.

2. SES OVERVIEW

Our solution consists of two parts namely, the server-side and the client-side. Smart Edge Server, the server-side module, is an enhanced wireless access point whose components are depicted pictorially in Figure 1. The Client side resides on a mobile device (e.g. PDA) and is a proxy with security related enhancements and resource management hooks. The SeS is further organized into three major management components: security, media and resource manager. Security manager handles all issues related to authentication and secure communication between the client and the SeS. Media manager performs a wide wariety of tasks, ranging from media content adaptation, to virtualization of media storage. Resource manager is capable of delivering a good quality of service to a client while increasing battery lifetime and seamlessly migrating wireless connection between different wireless network interfaces. Specific functions of these components are elaborated in the subsequent sections. In this paper we present a prototype implementation of the SeS. Although we use browser (HTTP protocol) interaction with a user as a sample scenario throughout the paper, our work is completely applicable and has been used with other applications and protocols.

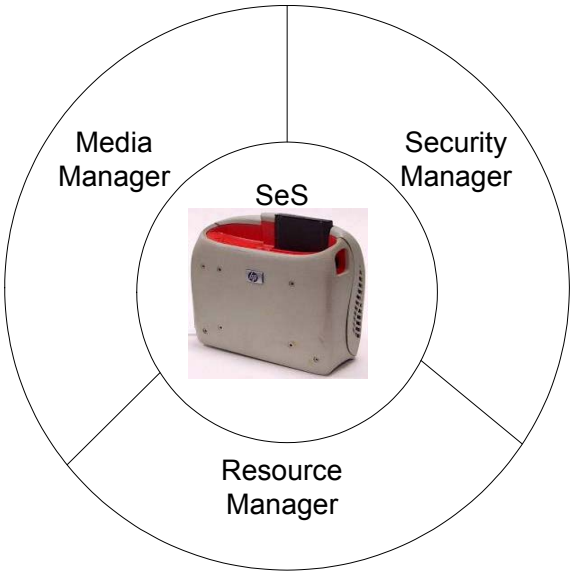

Figure 1 Main components of SeS

3. SECURITY MANAGER

There are many different Wireless ISPs deploying access point solutions at public locations **[2]**. Just as in any public wireless network, a mobile user gets network and local services by subscribing to the ISP deploying the SeS. SeS provides user authentication and authorization in such a completely un-trusted environment (public place) that has support for service level agreements among the different service providers so that the user can access services to which he has subscribed elsewhere. The primary responsibility of SeS is to verify the authenticity of the authorizations issued by the service providers prior to allowing the users access to services. This section gives an overview of the SeS security infrastructure that satisfies the above needs.

There are three critical steps which need to be occur before secure data communication can start: registration of SeS and client with internet service provider (ISP) , three party authentication (client, SeS and ISP), and the authorization for services requested by the client. Each of these steps is described in more detail below. Although in all the examples we describe our prototype implementation which uses HTTP, our security management infrastructure can operate just as well when using other approaches. For example, protocols such as PANA [16] are used for network authentication. We have also used our key management protocol to set the IPSec [24] secure associations between the client and the SeS for IP level connectivity – enabling secure multimedia and VoIP applications. In addition, SeS facilitates end-to-end secure association between the client and a remote server by acting as a VPN gateway.

In the first step of managing secure communication between the client and the SeS users are assigned a user-id and a password by a Verification Server (VS), a global entity trusted by ISPs. Any service provided by ISP, including connectivity, is further authorized by the ISP's Authorization Server (AS) to which a user must register in order to use that service. The communication between the user, AS and VS is secured using SSL after appropriate certificate validations. Client first submits <user-id, password, VS> securely to the AS of the ISP. The AS then verifies the data with the VS and responds with a generated <shared key, client-id> tuple to the user. This tuple is used for client authentication with the SeS. Our new shared-key distribution technique can be used while the client is mobile and as such provides the following benefits:

- The user-id/password provided to the user by an ISP are used minimally - only for shared key generation, as opposed to it being used every time the user authenticates to an SeS. As a result, the possibility of compromise is curtailed.
- If a device gets lost, the user requests of AS to disable access to just that one device. All other devices in the user's possession can still continue to be used without any reconfiguration since they share different keys with the same AS.
- A policy can be set at the AS whereby a shared key expires after certain duration of time. Using our scheme the shared keys can be generated again and distributed - that will foil any malicious attempt to guess a shared key by analyzing previously captured traffic between the client and AS at the time of authentication and impersonate the user.
- The dynamic generation of <shared key, client-id> provides some anonymity since user names are never disclosed to the SeS - what the SeS sees is the just the client-id..

The next step represents the core of our security infrastructure – our new three-party Key Distribution Protocol **[6]** where

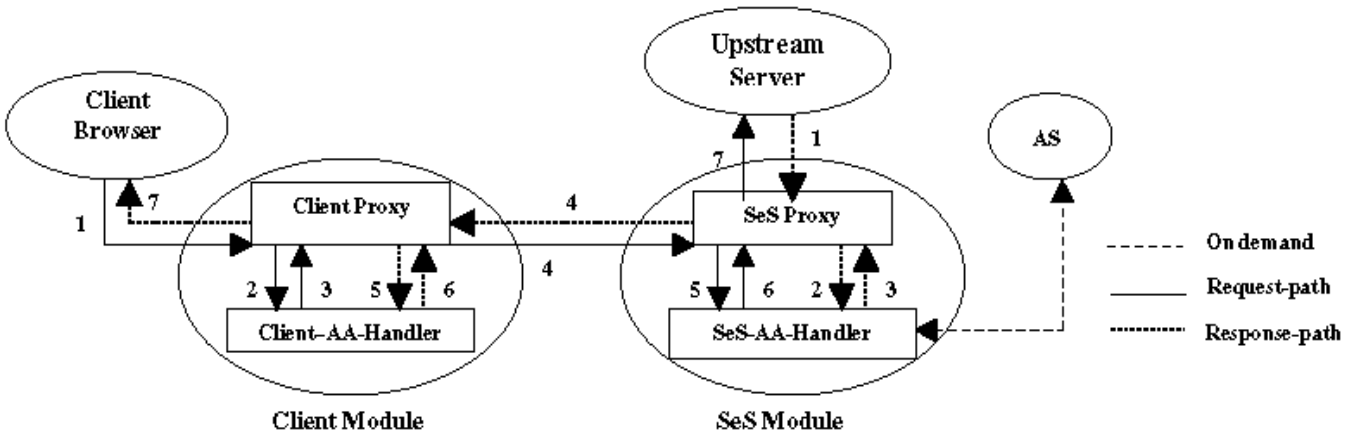

Figure 2: SeS Security Infrastructure

fundamentally two mutually distrusted entities, client and SeS, come together through AS and interact wirelessly. A detailed proof of the correctness of our 3-party key distribution protocol based on BAN logic [23] is provided in [22]. Our three-party authentication protocol accomplishes this critical step in two phases, the *authentication phase* and the *access setup phase*. In the authentication phase, the client, SeS and the ISP (AS server) mutually authenticate each other using the shared key obtained during registration and derive a temporary session key. In the access setup phase, the AS sends the client authorization information to the SeS and securely communicates a service access key to both the SeS and the client. At this point both encryption and authentication keys are generated and exchanged in the a channel secured with the service access key. There is a number of advantages of our three party authentication. Explicit and independent authentication between AS-SeS and AS-Client allows SeS to not have to authenticate with AS for each and every new client. In addition, there is minimal exchange of session keys - limiting the possibility of session key compromise and the possible replay attacks. The session keys are periodically refreshed when the client re-authentications occurs. With our protocol we aim to remove the service-theft and man-in-the-middle attack issues poised by unauthorized clients and rogue SeSs.

As a last step prior to data communication, our security manager sets up the authorization for various services between the client, SeS and the ISP. At the AS, in order to enforce access control for services, the users are classified into realms based on the <user-id, VS> registered with the AS. This service authorization information is also communicated to the SeS during the key distribution protocol for local decision-making. When a user accesses a service owned by an AS, AS_i, the client software initiates a user authentication with AS_i transparently. The authentication and authorization is completely implicit and the security transformations are handled by the server-side and client-side software. Neither the client applications nor the application server need to be aware of the authentication and authorization related details. Our security infrastructure also supports establishment of trust when a user requests a service that is spread across multiple SeS, potentially in different administrative domains [5].

Figure 2 outlines sequence of communication between the client and the SeS through our security management layer here represented by client and server proxy & handlers. Three paths are represented: on-demand path through which both client and the SeS can periodically re-authenticate with the AS, the request path that leads from client to SeS, and the response path from SeS to the client. Here we outline the seven steps illustrated in Figure 2 that are needed for the secure data transmission along the request path (mirrored seven steps are also shown for the response path):

1. The user's HTTP-Client-Software creates a HTTP request and hands it over to the Client Proxy.
2. The Client Proxy accepts the HTTP request and upon receiving, hands it over to the client's authentication and authorization handler (Client-AA-Handler).
3. The Client-AA-Handler, optionally, encrypts the HTTP request using the encryption key. It then computes a Message Authentication Code (MAC) of the request using the authentication key. The MAC, applicable client-id, and the HTTP request are returned to the Client Proxy. The Client-AA-Handler also initiates the protocol described in [6] in case if the session-key is not there or has expired.
4. The Client Proxy frames an additional HTTP header (called AUTH_SPEC_HDR) containing the signature and the client-id. The complete request is then forwarded to the SeS Proxy.
5. The SeS Proxy accepts the HTTP request, and passes it on to the SeS' authentication and authorization handler (SeS-AA-Handler).
6. The SeS-AA-Handler parses the AUTH_SPEC_HDR HTTP header. It uses the client-id as a key to the key table to retrieve the encryption/authentication keys for this client-id and applies reverse transformations on the data. The complete request is then returned to the SeS Proxy along with the authentication and authorization status of the client.
7. The SeS Proxy checks the authentication status of the request, and, if authentic, does the normal HTTP proxying of the request; else, returns back error response to the Client Proxy.

In this section we have outlined how security manager sets up safe communication link between a client and the SeS. The next section presents how SeS manages transfer of media content with a client once the secure connection is established via our security manager.

4. MEDIA MANAGER

SeS views media as a single entity, part of a collection, that is either streamed or is accessed as a chunk of data. Many small appliances sold today have wireless interfaces (e.g. cameras have Bluetooth) whose usage is limited to synchronization with a nearby PC. Our approach greatly increases the potential for data sharing and media accessibility because SeS acts as a secure bridge through which a mobile user can upload data/image/video onto the personal storage without wires. SeS media manager has a number of capabilities, it can:

- *Adapt content:* Each mobile device has a different set of capabilities, thus any media content has to be adapted in order to be presented in a best way. This adaptation is done on the SeS as it has many more resources than a typical client.
- *Interact:* The users can access services that are either local (to the SeS) or remote. SeS supports a browser-based model for interaction from a client, typically a laptop or a handheld.
- *Organize:* Users view their media as a set of collections (as described in the previous section). They can organize their collections by either using tools provided by SeS or by third party solutions operated over the Web folders interface provided by the virtual store.
- *Upload:* Input-generating appliances like cameras can upload their media to the user's repository. The appliance contains a meta-data file that holds keys and other descriptors that provide the SeS information on where the media is to be placed. We developed the notion of *casual-download*, seamlessly uploading pictures from a media device when an SeS is discovered in its immediate neighborhood. The appliances are not always network enabled but have reasonable storage capabilities. This storage is viewed as a cache and when the appliance comes 'within range' of a SeS, the media can get uploaded. SeS provides support for BlueTooth and dock-based devices in addition to flashcard-based uploads.
- *Experience:* For a given media, the SeS generates meta-data descriptors (using MPV) that allow MPV-aware appliances to handle the media. Thus, SeS enables media-centric experiences for a wide range of appliances such as TV, audio players, and laptops.
- *Share:* Users can print photographs and create picture albums on CD-ROMs as SeS enables physical sharing of media too.

We next describe the content adaptation and distributed personal media capabilities (enabling interaction, organization, uploading, and sharing).

4.1 Content Adaptation

Mobile devices are typically limited in computation power, display size and even application capabilities. Much of the content that is available on the wired network is targeted at larger devices, and thus needs to be adapted for better viewing on small client devices. Adaptation of web content can be done potentially in one of the three points of a content delivery network - the client side, the content server or the edge server (SeS). It is not feasible to perform the content adaptation on client devices (e.g. video scaling on PDA) due to their limited computing power and memory; while server side adaptation has practical problems due to the vast number of content providers and diversity in client devices. On the other hand, there are many advantages of edge-side content adaptation. Availability of cached content (before and after adaptation) for use by more than one client helps in faster delivery of the content. The content at the SeS is amenable to adaptation based on the local physical context and network characteristics. Further, specialized machines for specific types of filtering (virus scanning, video transcoding) can be connected to the SeS to perform load balanced filtering under heavy traffic. The rule-based online content adaptation infrastructure at the SeS provides a framework for deploying content adapting services – making the content viewable on the diverse devices and also provides advertising opportunities to the ISP.

Our content adaptation framework is based on the IETF- OPES model supporting the ICAP[3] protocol, an RFC for encapsulating HTTP messages. One of the core components of SeS, the HTTP proxy, is enhanced with an ICAP client that optionally encapsulates all HTTP requests and responses and sends them over for potential modification to an ICAP server (either local or remote). The ICAP server then consults an ICAP rule engine to determine the appropriate filters to be applied on the request/response and adapts accordingly. Typically the SeS administrator configures the rules for adaptation – based on the user policy, QoS requirements, availability of local context dependent services, ISP needs and any specific agreement with a content provider. Figure 3 shows the different active entities of our content adaptation framework in a lifecycle of a request and the longest path a request could take before the response finally reaches the client. The SeS Client (SC) component is configured as the browser's proxy and so initially directs all HTTP requests to the SeS proxy. Additional HTTP headers are used to communicate the client information such as the type of OS and browser specific details.

Various content adaptors (proxylets) installed at the ICAP server may either perform the filtering themselves or utilize a web resource elsewhere to perform the modification. The rule engine performs a rule-based decision to determine which filter(s) to apply, based on the attributes of the request/response. Every rule specifies a regular expression that should match against the value of a HTTP header, and a particular proxylet as the action. The rules are typically set by the SeS administrator using a web-based control provided for remote management. New proxylets can be authored and installed using the same web-based configuration module.

In addition, a set of well-defined proxylet API has been defined to simplify the authoring of proxylets. The rule engine is also integrated with the local Authorization server wherein the SeS administrator would have set access controls for certain filters.

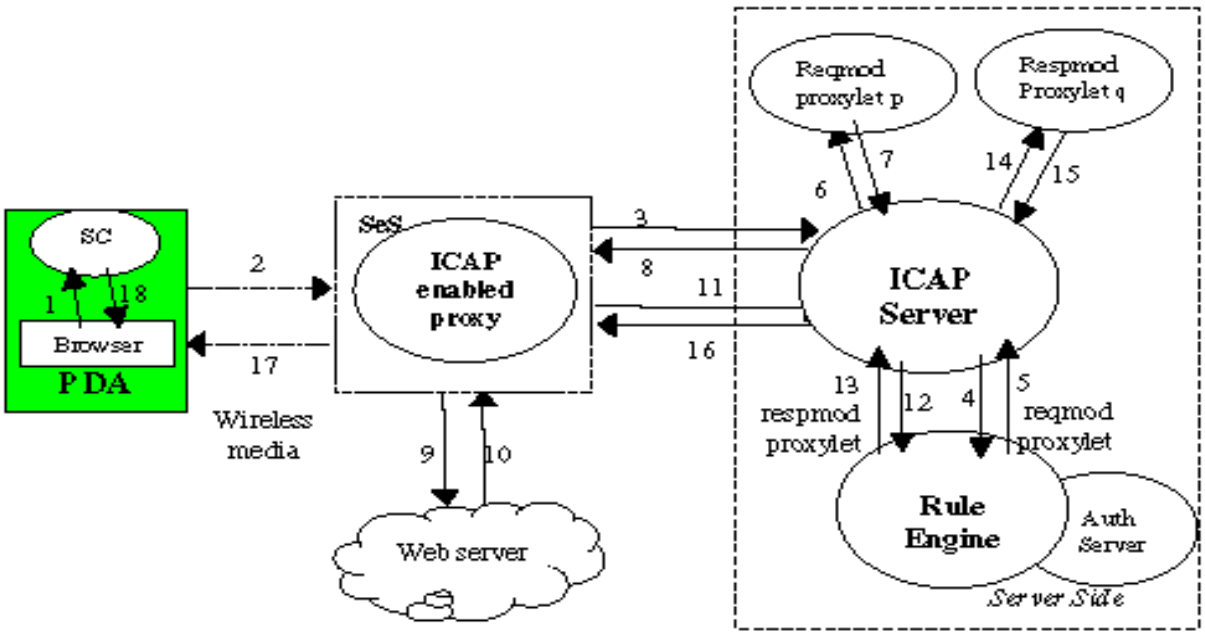

Figure 3: The Online Content Adaptation Framework

Based on the credentials of the client accessing the Server and the filter permissions, the rule engine sends the list of the filters to be enabled for a specific request.

Our infrastructure also supports automatic discovery of available filtering services and applies them based on user preferences. There is also a provision for service level agreements between the filter service provider and the access provider to provide integrated accounting information for a roaming user[13].

4.2 Distributed Personal Media

Distributed Personal Media Service (DPM) is an example application service built on top of the SeS modules. Typically a mobile user has his data on several different devices - home PC, office workstation, laptop, PDA, Camera etc. Ease of access, sharing, synchronization and archival of this distributed data ensuring confidentiality is a real concern. The virtualized storage of SeS enables a consolidated view of all the different pieces of the user's personal media. Further, since the naïve user is comfortable viewing the data as a collection of media as opposed to a directory of files, DPM provides a media collection abstraction of our virtualized storage.

The core component of the DPM is a Storage Virtualizer. We consider each unit of personal media (on different devices) as a "Physical Repository (PR)" and provide a virtual consolidated media repository, called as "Virtual Repository (VR)", to the user. The physical repository could either be a local file system, a network accessible remote file system, a full-fledged storage appliance, disk space bought over the Internet or yet another Virtual Repository. A Virtual Repository is created for every new session initiated by the user and includes the Physical Repositories currently configured in the user's policy. Multiple PR file system interfaces will essentially be supported by the VR – the currently supported ones include standard POSIX file system and WebDAV protocol.

A VR is created on the Smart Edge Server (SeS) closest to the mobile user since the caching and replication algorithms to improve the availability, performance of this storage are best deployed at the edge of a content delivery network. Even the VR provides multiple interfaces for the mobile user – the standard WebDAV protocol, content specific web views through a web service, disk drive for Windows and a mountable file system for Unix clients. The user views a single consolidated store of his media without being aware from which of his personal computing devices the data is being fetched. Any local modification on the personal data is reflected back at the (probably) remote site automatically – the policy for writing back and ensuring consistency is based on the caching/replication policy in effect on the SeS.

Further, we have integrated our security infrastructure described in an earlier section with our virtualized storage to provide user specific views of the storage. Authorization controls can be exercised either at the PR or at the VR. At the PR, WebDAV server is modified to verify the user access permission with the authorization server and formulate an appropriately masked reply as a response to the PROPFIND method of WebDAV. On the other hand, the SeS administrator typically performs the access controls for the VR.

The set of physical repositories to be included in a user's virtualized storage is dependent on the pre-defined policy information that can be specified in multiple forms. The user can specify the network address of his home SeS (also called a Family Data Centre) and DPM would pick up the list of PR's from there. Alternatively, device specific views can be provided through dynamic upload of policy information when the user is first authenticated. A session manager on the SeS manages the user sessions accommodating dynamic *import* of PRs.

A user typically collects all his digitized pictures, audio and video files as albums or collections for ease of access and sharing. Our Distributed Personal Media (DPM), an application over the virtualized storage, provides a collection/album view of personal media – as opposed to a set of media files. This service is typically intended to access and share personal media. Sharing of

personal media with friends and relatives needs to be regulated and that regulation should not be very visible. This is ensured in a DPM. For example, if the data viewed is a photo album, a friend who needs enforcement of authentication controls would get a view of the personal media with unauthorized albums masked out. He will be completely unaware of the existence of the unauthorized album. In essence, this results in different views of the virtualized storage for different users. The authorization controls can be exercised at the level of collections using a Web Interface by the owner of the media. The owner can essentially define user-groups (friends, relatives, owner and so on) and allow sharing of the collections for specific groups only.

The collection information is available as a MPV (multi-photo video standard)[17] file. The grouping of media files to form specific collections could therefore be either logical (metadata based, content type, explicit grouping) or physical (based on the their location in the file system). MPV is also used to provide a configurable presentation of a collection based on the media type.

Another interesting view of DPM is as a WebDAV folder as the virtual repository itself is served over the WebDAV protocol. Here, the DPM can be thought of as alleviating the disk space limitation of a client device by providing a consolidated disk drive consisting of all the user's data. If the user creates a file on this disk drive, it would physically reside on a remote machine but provide a local access mechanism to help local applications work on the remote data without any explicit configuration. We also envisage a usage wherein some temporary store is leased out of the local disk space of the SeS and periodically backed up at a remote site (determined by the user policy information).

5. RESOURCE MANAGER

Main resources that SeS manages are client connectivity, power and Quality of Service. As today's clients have multiple different wireless network interfaces (WNICs), the SeS makes decision on when a device should move from one to the other WNIC depending on QoS and power demands. In this section we first describe the technique we used to enable seamless migration from one WNIC to other. We follow up with a detailed description of the policy that decides when migration should occur, and when the currently used WNIC should be in low power state.

5.1 Seamless Wireless Migration

Today's mobile devices support multiple wireless interfaces. Different links offer diverse characteristics in terms of range, speed and power consumption. As new wireless technologies are developed, they will be added to the SeS while the existing link layers can be kept for compatibility with older clients. For the same reasons, the client is also likely to include multiple wireless link types. Thus, the client and the SeS will often have a number of wireless links in common. Depending upon network/device conditions, application and user needs, the best link may be change during a communication session. Our Connection Diversity (CD) framework provides a link level abstraction for seamless connectivity across the diverse physical networks.

A key feature of CD mobility support is that it does not require any support in the infrastructure. This makes it easy to deploy and it enables inter-domain mobility (mobility across different ISPs). Further, our CD framework maintains the same session across multiple links if the interface switch occurs within a single SeS without any application level support. The CD's name resolver allows the client to interact with the SeS without having to know its DNS name or IP address. It also enables the client to discover other clients and refer to them with a short local name. Lastly, the CD defines API that enable applications on the SeS to get information about specific clients, to know which wireless link they are currently using, and to get event notifications when this changes.

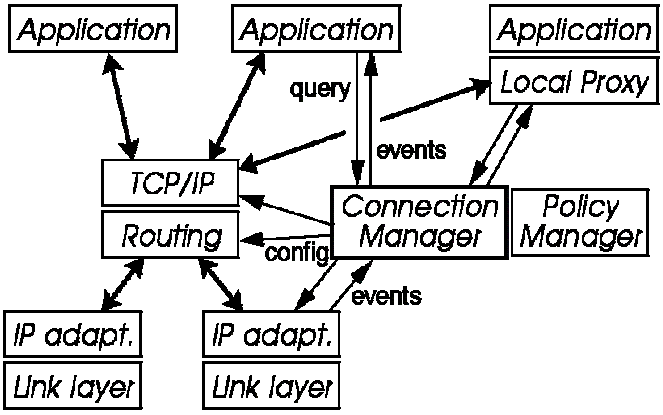

Figure 4: The Connection Diversity Framework

The CD framework shown in Figure 4 is a set of components and interfaces in the client and the SeS, that abstract the various wireless link interfaces and simplify their use. The SeS provides the clients with proper IP configuration via DHCP, runs the connection manager (CM) to monitor and handoff connections, and optionally provides HTTP proxy autoconfiguration. Each client is handled individually, and multiple clients can be supported on the same interface simultaneously. In our current prototype, the SeS keeps all of its interfaces active at all times.

The central piece of the framework is the Connection Manager (CM). The role of the CM is to discover, evaluate, setup and monitor the various paths between the client and the SeS on behalf of the various applications. It directly manages the various link interfaces and includes abstraction modules specific to each link layer used for tight integration with each link interface. The CM performs link discovery to find which paths are available, activates it and configures link interfaces on-demand to enable their use, monitor them for failure, and disconnect them when idle. The CM try as much as possible to use link specific methods for those tasks, for example over BlueTooth it uses the link native discovery, and over 802.11 it uses packet drop events to detect connection failures. It also use generic methods as backup, such as monitoring incoming and outgoing IP packets to detect link idleness and failures. The Policy Manager (PM) component selects the most appropriate link to connect from the client to the SeS based on the current policy, applications requirements and link availability. The CM currently can manage IEEE 802.11b, BlueTooth BNEP and IrDA links. The client component is also integrated with other long-range wireless protocols such as GPRS.

The CD framework provides mobility support to migrate client connectivity seamlessly not only between link layers but also between two SeSs. Mobility between two SeSs is handled using an application layer based hand-off protocol [21] implemented in the Connection Manager of the client. The application uses a

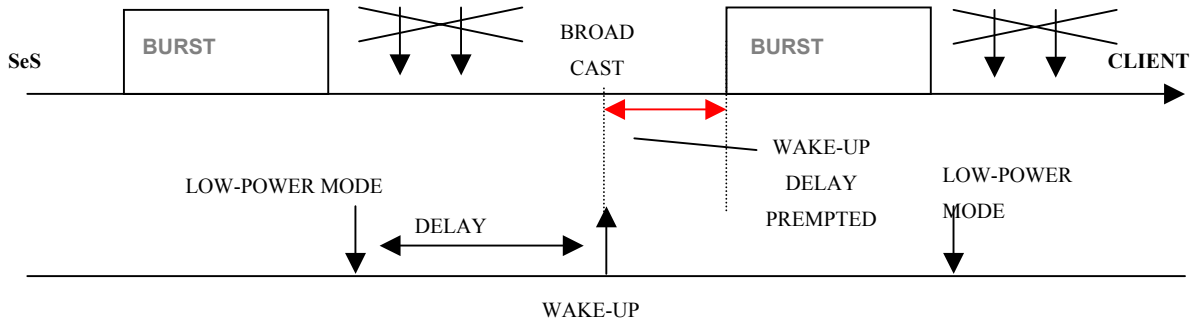

Figure 5: Communication between the client and the SeS for better QoS

direct connection to the Internet, and interacts with the CM to detect connectivity changes and adapts to them. Mobility between two interfaces of the same SeS is handled using a simple vertical handoff protocol [11] implemented in the CM of both the client and the SeS. It switches IP routing between interfaces on both sides of the wireless links – client and SeS.

There is many differences between our approach and traditional mobility protocols such as Mobile IP. Our solution doesn't require any infrastructure support (no Home Agent) and works through NAT and firewall, which significantly eases deployment and allows inter-domain mobility (mobility across different ISPs). Our solution also makes the client local, so it can take advantage of the locality aspects of the SeS, such as a local cache and location aware services. The network performance of our handoff protocol is typically better than MobileIP because the connection is always direct and not through a home agent.

5.2 QoS and Power Management

We have exploited the above feature of wireless migration to provide better client resource utilization. Mobile clients typically have limited battery-lifetime and communication abilities. Our measurements indicate that a large fraction of battery energy is spent for communication (as much as 50%), with more than 90% of that spent listening for any incoming traffic. Thus careful scheduling of communication, and management of multiple wireless interfaces can offer large improvements in battery lifetime of the client device with no perceived performance degradation. We believe that SeSs are great candidates for efficient scheduling as they are not power constrained, and know both wired and wireless network conditions. The SeS obtains the client device characteristics, and monitors communication patterns [8]. Based on this, it seamlessly switches to the appropriate wireless interface on each client, schedules communication sessions and directs when the client enters a low-power state.

The goal of Resource Management (RM) is to enhance the Quality of Service (QoS) while maximizing the battery lifetime of the client devices. RM's primary task is to determine what network interface is most suitable for the client's needs and how to manage its power and performance states. When an application starts on a portable device, RM pre-selects those Wireless Network Interface Cards (WNIC) for data communication whose average throughput is greater than the data consumption rate of the application. As conditions on SeS, wireless link and the client change, the SeS can choose to switch to another WNIC, and also adapt the times when data is transferred between SeS and the client. This ensures that the QoS requirements of the client's applications are satisfied.

Figure 5 shows how communication occurs between the SeS and the client device. Data is transferred between the client and the SeS in large bursts and then client's WNIC transitions in a low-power mode until the next agreed point of communication with the SeS. During this time much energy can be saved thus enabling longer battery life on the client. The client wakes up on time to receive the next burst of data and thus does not cause any degradation of service. In addition, contention for the wireless medium is now reduced so that SeS is able to support more clients simultaneously. Thus, this efficient control and scheduling of transmission increases the battery life of the client device, increases the accessibility of SeS and improves the quality of service in multiple client environments. The control-related handshaking between the client and the SeS occurs with each data burst.

SeS has to communicate to the client ahead of time how large data burst to receive before going into a low power mode and when to wake up for the next burst. The size of the buffer directly affects the energy spent in communication. If the size of the buffer increases, the average power dissipation of the communication device diminishes due to longer sleep periods and thus less overhead in transition between power states. RM pre-selects WNICs for a particular application based upon their average throughputs and the data consumption rate of the application. The changes in the rates are observed using maximum likelihood estimator. The WNIC that offers minimum power dissipation with regards to communication and RAM is selected. RM also defines the appropriate low-power state of the WNIC along with the switching points. Additionally, it can dynamically switch the selected WNIC if a change in its throughput and/or the average data consumption rate of the application is detected.

6. SES PROTOTYPE

A prototype of our Smart Edge Server, based on off-the-shelf embedded Linux box and our proxy-based software including all of the modules described in the previous sections has been demonstrated within HP. All the local services residing on the SeS are built using Coolbase Web Application server that supports authoring web dynamic services either as C or Python classes [15].

Figure 6: Demo scenario

Figure 6 shows our demo setup. SeS is in the middle with various clients surrounding it: speakers for streaming audio, laptop showing secure access to media storage, Bluetooth camera uploading data to SeS, IPAQ playing back a movie clip and printer with a printed photo. The SeS management infrastructure is designed so that better joint decision making between the three managers (resource, media, security) is possible. In this way the the managers act in concert when it comes to deciding when and over what link to send video frames (resource manager), what size video should be sent (media manager) and what level of encryption should be used (security manager).

As a part of our demo we show an authenticated and authorized mobile user accessing the Internet securely through SeS – using either Wavelan/802.11b or Bluetooth (IPAQ playing video, speakers playing music and laptop showing photos in Figure 6). The content adaptation rules are configured so that MPEG video delivered to the client device is scaled down to fit the size of the screen (Shrek on IPAQ in Figure 6). We also demo a tailored HP PhotoSmart 812 digital camera uploading a captured picture over Bluetooth to a selected album on user's Distributed Personal Media when the camera is in the vicinity of our SeS (camera is next to SeS, the pictures are displayed on the laptop and one is printed on the printer). Similarly, we have shown easy transfer of pictures acquired by Nokia 3650 cell phone through Bluetooth. We next discuss the performance attributes of each component in turn.

Security manager's main contributions are in the area of key distribution, authentication protocol for setting up secure sessions, and authorization techniques. The algorithms for authenticating data communication between the SeS and client and encryption are also used by other standard security protocols, and thus their performance is comparable to what is currently out in the market. Our three-party Key Distribution Protocol has two areas in which very minor performance overhead occurs:

1. The authentication process involves a few protocol messages as described in [6]. This overhead is incurred infreuently and as a result can be neglected.
2. The authorization process involves querying the SeS-AA-Handler (solid lines 5 & 6 as shown in Figure 2) for each URL requested by the client (in case of HTTP). The overhead here is a call between the SeS Proxy and the AA-Handler. Some of this overhead can be mitigated by caching information at the proxy.

Table 1: Results showing the effect of Content Adaptation

Transfer Characteristic	**No Content Adaptation**	**With Content Adaptation**
Movie Size [bytes]	37589390	19484040
Requests per second	0.16	0.31
Time per request: [ms]	6206.488	3233.952
Transfer rate [Kbytes/s]	591.45	588.41
Mean Connection Times [ms]		
Connect:	9	4
Processing:	6196	3229
Waiting:	15	46
Total:	6205	3233

We next compare the performance of our system with and without the content adaptation framework that is a critical part of media manager. A major part of content adaptation is transcoding, which typically results in significantly smaller overall media size transferred. Results in Table 1 show that with our framework we are able to shorten the connectivity time to wireless by more than a factor of two, which also gives a significant reduction in client's energy consumption and an improvement in the overall wireless bandwidth available to the other clients communicating with the SeS.

Table 2: Typical handoff characteristics

Link layer latencies	**IrNet**	**802.11b**	**BlueTooth**
Discovery period	3s	10s	60s
Connection setup	0.8s	0.3s	0.8s
Link breakage detection	1s	0.1s	0.5s

Seamless wireless migration performance is mostly governed by the characteristics of the individual link layers and latency of the events triggering handoff. The values outlined in Tables 2 and 3 are typical of our implementation. The handoff time between link layers of the same SeS is exactly the sum of the link breakage detection time on the old link and connection setup time on the new link [11]. Handoff between two SeS takes more time ; Table 3 shows the breakdown of a handoff from BlueTooth to 802.11b between two SeS [21]. Clearly the extra buffering is needed at the client to compensate for the time it takes to handoff. Resource management framework we developed takes care of scheduling so that handoff is seamless from the application level, and at the same time power is conserved.

Table 3: Handoff from BlueTooth to 802.11b

1.	BlueTooth link breakage detection	700 ms
2.	802.11b link and monitoring setup	98 ms
3.	DHCP to configure 802.11b link	1789 ms
4.	Proxy API processing time	6 ms
5.	WPAD, DHCP to get Config. URL	109 ms
6.	WPAD, query proxy.pac via HTTP	10 ms
7.	Parsing, connect to upstream proxy	4 ms
	Total Elapsed Time:	2716 ms

Lastly, we performed measurements with SeS and IPAQ/Linux client that support both 802.11b and Bluetooth interfaces. The power measurements are collected with a DAQ card at 10ksamples/sec. We have used TCP for all data communications and bnep for Bluetooth. Results for an MPEG4 video (320x160 clip) running at 15frames/s, shown in Figure 7, highlight savings of 65% in energy consumption when using our resource management (RM) over the best possible savings with 802.11b MAC layer power management (802.11b PM), with no degradation in QoS. MAC layer power management for 802.11b typically achieves significantly smaller savings due to broadcast traffic. Measurements presented in [8] show that in medium to heavy broadcast traffic, 802.11b PM achieves at most 10% savings in energy consumption. In contrast, our resource management algorithm does not suffer from the broadcast traffic problem. As a result, its savings can be significantly higher in realistic conditions than in ideal conditions shown in Figure 7 for 802.11b PM.

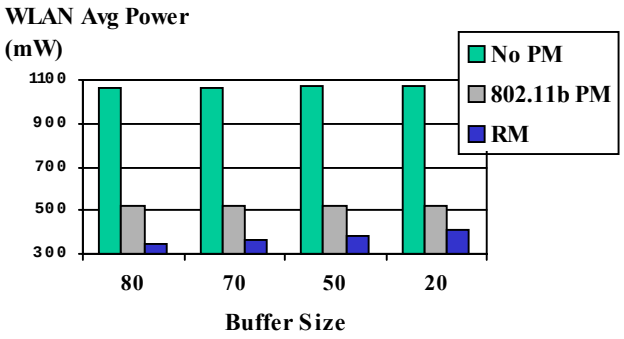

Figure 7: WNIC power consumption for MPEG4 video

In another experiment, we analyzed the performance of RM when the application data consumption rate changes by creating an application trace consisting of MP3 audio, email, telnet, WWW and MPEG2 video. We found that RM offers a factor of 2.9 times improvement in power savings over just employing Bluetooth with park mode, and a factor of 3.2 times higher than standard 802.11b power manager. Moreover, RM enhances the QoS since wireless interfaces are switched to match the data usage pattern of the application. We have also demonstrated seamless migration of connectivity from low bandwidth GPRS to 802.11 with a Linux client receiving an MP3 audio stream with similar savings while keeping MP3 decode real time.

As can be seen from the above discussion, our Smart Edge Server provides secure wireless access to the clients, has sophisticated media handling and storage capabilities and uses advanced techniques to manage resources available to it, such as bandwidth, power and the type of connectivity. As a result, the SeS platform provides immense opportunity for research and development in the area of mobility and media delivery.

7. REFERENCES

[1] Business Week, Special Issue on Wi-Fi, http://www.businessweek.com/magazine/toc/03_17/B3830031 7wifi.htm

[2] Wireless ISPs, http://www.bbwexchange.com/wisps/florida-wisps.asp

[3] Internet Content Adaptation Protocol, http://www.ietf.org/rfc/rfc3507.txt

[4] A Python ICAP Framework, http://icap-server.sourceforge.net

[5] Devaraj Das. IPSec-based Delegation Protocol and its Application, Future Trends in Distributed Computing Systems, IEEE Computer Society, May 26-28, 2004. Suzhou, China.

[6] Prakash Reddy, Venky Krishnan, Kan Zhang, Devaraj Das: Authentication and Authorization of Mobile Clients in Public Data Networks. Infrastructure Security International Conference, InfraSec 2002, Bristol, UK, October 1-3, 2002, LNCS 2437 and HPL-2002-213

[7] D. Das, G. Manjunath, V. Krishnan, P. Reddy : "Hotspot! – a service delivery environment for nomadic users system architecture", Hewlett Packard Technical Report, HPL-2002-134, 2002 and NCC2004, Indian Institute of Science, Bangalore,

[8] T. Rosing, A. Acquaviva, V. Deolalikar, S. Roy: " Server-driven Power Management", PATMOS 2003.

[9] W. Quadeer, T. Simunic, J. Ankcorn, V. Krishnan, G. De Micheli, "Heterogeneous wireless network management", PACS 2003.

[10] T. Simunic, L. Benini, P. Glynn, G. De Micheli: "Event-Driven Power Management", IEEE Transactions on CAD, pp.840-857, July 2001.

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

[12] Casey Carter, Robin Kravets & Jean Tourrilhes. Contact Networking: A Localised Mobility System. Proc. of MobiSys 2003.

[13] Geetha Manjunath, Venkatesh Krishnan, *A Content Adaptation Framework that supports mobility*. HPLabs Technical Report, to appear in 2004.

[14] Geetha Manjunath, Venkatesh Krishnan, *Distributed Personal Media* , HPLabs Technical Report, to appear in 2004.

[15] Devaraj Das, Geetha Manjunath, Venkatesh Krishnan, *Dynamic Web Services using Coolbase Appliance server*, HPLabs Technical Report, 2004.

[16] D. Forsberg et al, “Protocol for Carrying Authentication for Network Access (PANA)”, IETF Draft, Feb 9, 2004

[17] Multi Photo Video standard, www.osta.org/mpv

[18] “How much information 2003”, UC Berkeley report, http://www.sims.berkeley.edu/research/projects/how-much-info-2003/, 2003.

[19] K. Wong, H. Wei, A. Dutta, K. Young, “Performance of IP Micro-Mobility Management Schemes using Host Based Routing,” WPMC , 2001.

[20] Z. Jiang, K. Leung, B. Kim, P. Henry, “Proxy Servers Based Seamless Mobility Management,” WCNC, 2002.

[21] J. Tourrilhes. “L7-mobility : A framework for handling mobility at the application level,” Proc. of PIMRC, 2004.

[22] D. Das, "Design and Implementation of an Authentication and Authorization Framework for a Nomadic Service Delivery System", MS Thesis, Indian Institute of Science, March 2003.

[23] M. Burrows, M. Abadi, and R. Needham, “A Logic of Authentication,” Technical Report 39, DES SRC, 1990.

[24] R. Atkinson. "Security Architecture for the Internet Protocol", RFC 2401, IETF, August 1995.