Keyless Media Delivery and Security

Location-aware computing through Interaction Design

David A. Shamma, Kristian J. Hammond, and Nader Varjavand *

Intelligent Information Laboratory, Northwestern University,
1890 Maple Avenue, 3rd Floor, Evanston, Illinois 60201 USA
{ayman, hammond}@cs.northwestern.edu
nv@northwestern.edu

Abstract. In this article, we describe the MediaPoint system that uses constraints of WiFi mobility to provide services proactively. In particular, we outline how the user's logical location can be used to deliver media, perform keyless search, and ensure security. A user's location and other contextual information are used to retrieve task relevant information and avoid many of the problems posed by limits of mobile devices and their use. MediaPoint is a new 'smart' mobile technology that uses context and information retrieval to provide meaningful content and automate authentication without friction.

1 MediaPoint

Imagine a world where people and media are connected. Where the act of walking into a train station or airport brought the current departure times to your handheld computer (PDA). And simply running a media player in a coffee shop brought the Sundance Film Festival to your laptop.

Out and about in the world, people are already engaged in tasks such as shopping, eating, seeing movies, etc., so any system designed to work along side and assist, should not detract from the actual task at hand. In practice, the limited modality of ubiquitous devices makes this challenge non-trivial in that user interfaces must deal with the constraints not only of the device, but also aspects of the user, who may be carrying several shopping bags and a cup of coffee. In a mobile setting, context is your location and media.

Previous location-aware systems focused on the user's exact physical, geospacial location. Once again, this requires specialized equipment such as a Global Positioning System (GPS) which delivers a fairly fine resolution, anywhere from 10 – 100 meters. Low fidelity Mobile Positioning Systems (MPS) which retrieve geospacial coordinates via cellphone triangulation have increased map and direction tools [1] but little else. Our focus is on the user's engaged task; the user's *logical* location is the important factor and not their actual longitude and latitude coordinates. Fortunately, identification of logical location context can simplify the input, in many cases down to one press of a button. We built a system called *MediaPoint* to assist by removing search entirely and connect people and media.

* The authors would like to thank Larry Birnbaum for his assistance in conducting this research.

1.1 Interaction Design

Our focus is the PDA or laptop computer in a WiFi space. Once only found in local intranets, public WiFi network locations (HotSpots) have been appearing in coffee shops, airports, and even fast food restaurants. In the public spaces, the average user tends to stay for a duration, and not quickly jump in and on the network for less than a minute and leave. We identified three embodiments where MediaPoint could assist the user and provide *frictionless* access to media and content.

A Media Player Plugin. Our main motivation is a user in a coffee shop who needs to watch location specific media. Fig. 1 shows the first built MediaPoint plugin embedded in Microsoft Windows Media Player. Simply running a media player brings video tailored for the coffee shop you are sitting in. As not to disrupt the user's work flow, perhaps the media player is launched to play an audio compact disc, if the media player is busy, MediaPoint quietly makes itself visible in the player with its discovered location and a simple button to play the location's media.

Fig. 1. MediaPoint embedded in Microsoft Windows Media Player. Here, MediaPoint has discovered it's location at the Starbucks on Main Street.

On the desktop. Being inside a media player provided a clean interaction with media, but there was no notification mechanism. Users had to run the media player, in some cases, only to find there was no media. As a simple notification mechanism, we moved MediaPoint to the desktop to enable a more pervasive interaction.

Quietly, in the background, MediaPoint checks to see if the location has changed. Fig. 2 shows MediaPoint in the notification area of Microsoft Windows XP operating system. Here, MediaPoint notifies that the laptop has entered the Northwestern University Computer Science Department by displaying a balloon tooltip. To visit the URI, the user need only click on the MediaPoint icon. If the user doesn't want media, the balloon will disappear after a short timeout. Now documents, files, webpages, and media (anything addressable by a URI) could be provided based on the user's location.

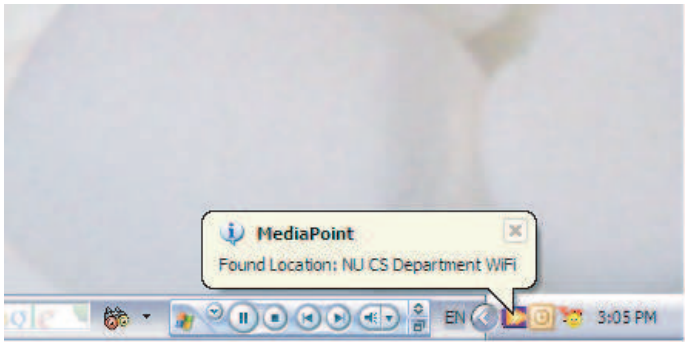

Fig. 2. MediaPoint on the desktop. MediaPoint has discovered the laptop has entered the Northwestern University Computer Science Department.

Inside a Web Page. When people move around, they know where they are, but not precisely. Many cases, especially in more urbanized areas, they do not know the particulars of where they are like the street address or zip code. These particulars are needed to find directions to a location or nearby movie theaters and their listings.

Towards this end, we designed an embodiment of MediaPoint that lives as a embedded control in a webpage. Once the control has identified the location, data associated with that location, such as the zip code, can be send down to the browser proactively. Simply visiting the movie theater's webpage from a MediaPoint enabled location will bring up the local listings automatically.

1.2 Logical Location Signature

MediaPoint constructs a signature of the user's logical location and uses it as the identifier to deliver content without requiring the user's active participation. The signature can be obtained through interactions with wired or wireless networks. From this we are able to identify "known" locations using IP-address digital signatures and provide information that is linked to the idea of the place they are in rather then the physicality of it. Data and information about the signature is stored in various databases that bring

media back to the user. Viewing content can be as detailed as content channels leveraging off the known user's location or as simple as a push localized to the signature's known zip code in the database.

Media and content can be delivered via many applications (media players, web content, multi-media messaging, etc.) with different infrastructures and transports. In order to preserve transparency, MediaPoint communicates via a secure, well-formed protocol which can be embedded in an application appliance or encapsulated in a web object delivered within a webpage or web application or form.

From where it sits on the client, MediaPoint makes a representation of its view of the gateway and external network, using the native transports of the application appliance. It sends this representation as a query to a server, which in turn sends back the appropriate result, based on the request and the requester.

2 Relation to Previous Work

Digital media and content, however accessible, requires people to find the media and specify their locale. In many cases, the user must find the needed information either by exhaustive web searches or by visual hunt and peck through portals and data aggregators. The simplest case, where the user remembers and enters the Uniform Resource Identifier (URI), still presents a challenge.

Many wireless fidelity network (WiFi) devices, such as tablet computers, PDAs, and cell phones, have limited input capabilities. Voice and handwriting recognition do not perform accurately enough to perform several searches or enter a URL. Voice recognition poses an additional problem in crowded public spaces. For phones and other devices with small keypads, several assisted input technologies, most notable is T9 Text Input® [2], have been successfully developed. These predictive input modalities are optimized for Short Message Service (SMS) and tend to give more advantage to the expert users [3]. In addition, they provide no assistance for entering URLs. More so, exhaustive querying is often complicated by smaller displays and slower processing speeds with these devices. The opportunity exists for a "frictionless" approach that makes use of any and all context information linked to these devices to more precisely address user needs.

Previous work on ubiquitous interfaces required people to be in a ridged space (workspaces, intelligent classrooms, etc.) and used context to discern actions and gestures in the space as meaningful interaction. This context can be based of a set of observations from the input [4] or judged by a planner [5]. In either case, the environment is tightly constrained. Such systems and architecture are too complicated and constrained to 'drop in' to existing WiFi networks.

3 MediaPoint Object Architecture

In all of its embodiments, MediaPoint is the manager between the media and application. MediaPoint sits anywhere between the client and the network as it is important that the network signature is the view from the client, and not some higher level proxy. This is not saying that the proxy itself could not be a MediaPoint enabled client rather

it is important to connect the end-point to the server. Also along these lines, any piece of hardware (such as the network interface card itself) or software can be a MediaPoint enabled client.

To facilitate this, the lightweight MediaPoint client can be embedded into an existing application as a linked object (e.g. .NET, COM, etc.) either programmatically or via an extensible application architecture. Fig. 3 shows an example of how the MediaPoint object can be made into a plug-in for an extensible media player. In this example, MediaPoint first gets the gateway and network information from the client operating system (OS), hitting the network if necessary to find additional information if not cached by the OS. This information is used to generate a unique identifier or signature, which it securely sends to a database of locations. The database hands back a Uniform Resource Identifier (URI) to MediaPoint which passes it to the player's display and plays the media. The lookup does not require the user to invoke the database query manually and the media is only played if the player is currently inactive (no other media is being loaded, cached, or played).

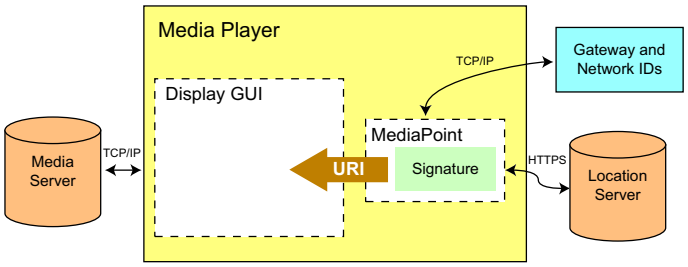

Fig. 3. A high-level view of the MediaPoint protocol. In this instance, MediaPoint is embedded as a plug-in inside an extensible media player.

As the previous example brings media to the user, the proactive lookup also returns other information along with the URI. In a thin client or web page, MediaPoint can be embedded as a web object (ASP.NET, ActiveX, Applet, etc.); such information can be used to refine searches without requiring the user to specify their exact city or zip code. Fig. 4 shows how a MediaPoint control can be added to a webpage. In this example, the MediaPoint object is pushed down from the web server embedded in an HTML page. After the object is instantiated, it generates the signature and queries the location database, as in the previous example. At this point, the MediaPoint object has the location information, and makes it accessible to the parent web page to use for display, further queries, and/or other scripts.

In either the lightweight or the thin MediaPoint objects, the conversation flow is preserved.

1. Find needed network information.
2. Generate a "unique" signature.
3. Query a database with the signature.

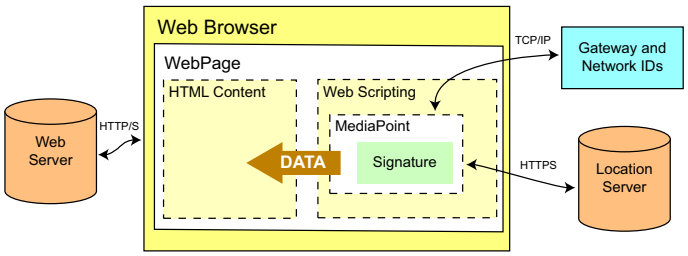

Fig. 4. An example of MediaPoint as a thin client. In this instance, a MediaPoint control is embedded as an object in an HTML page and called via a client side scripting language (JavaScript, VB, etc.).

4. Provide data and URI for the client.

As noted in step 2, a "unique" signature is generated. The resolution of this key is up to the developer. The key need not be entirely unique, but it is important that the possibility of two locations having the same ID be sufficiently remote. Added security would make extra information visible from the client (trace routes, gateways, WiFi IDs, searching for known services on the local subnet, etc.).

4 Location Secure Applications

MediaPoint's architecture enables it to be added to any existing network security policies, intranet proxies, or virtual private networks via the client operating system's native transport. Basic security is ensured via standard secure socket communication protocols. The signatures are also encrypted to prevent location spoofing attempts. Fig. 5 shows an example of secure key delivery.

4.1 Endpoint Security

Once these basic security layers are in place, MediaPoint can verify that a specific URI only be viewed at a given location. For example, a Wi-Fi venue can provide streaming content only at specified locations. MediaPoint does this by encrypting the URI that is sent back down from the database. Once MediaPoint has the encrypted URI, it is decoded with a pass-phrase which is based off of the network signature. Other clients (MediaPoint enabled or otherwise) will not be able to decode the URI or view media or content.

The location of the media is protected by the client's endpoint location. Additional security could encrypt the stream or content itself. To illustrate this, we will look at MediaPoint in a document environment, rather than an online transaction, though both systems could be used concurrently. Embedded in a word processing document or spreadsheet, MediaPoint checks the network signature when the document is opened. Fig. 6

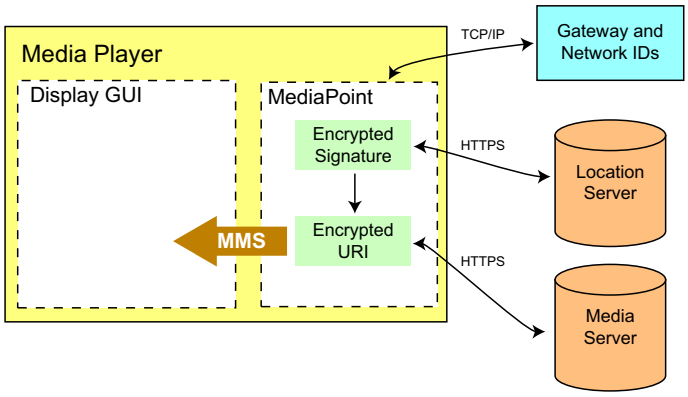

Fig. 5. MediaPoint with a secure signature delivery. Here, the unique key is encrypted with any standard method (DCS, RSA, etc.) as a signing mechanism to thwart spoofing attempts.

shows an example of this in a spreadsheet. If the signature matches that of a secure location, the document is then decrypted and displayed. If the signature/location is insecure, the document remains encrypted.

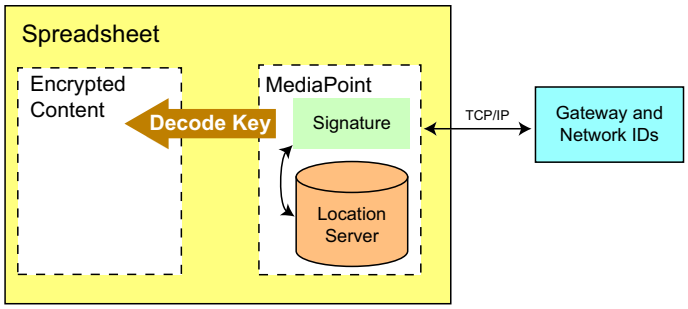

Fig. 6. A spreadsheet document is encrypted and contains an instance of the MediaPoint object. MediaPoint delivers the decode key when the document is opened if it is in a known secure location. In this instance, the spreadsheet itself knows the secure locations that are allowed.

4.2 Threat Detection

MediaPoint can also be used to detect a rogue or hostile network. In previous examples we have discussed, the focus was on ensuring the presence of a trusted network. Here we use MediaPoint to find and detect unwanted networks. For example, once a secure connection has been established, if a new network adapter appears which MediaPoint

does not know, the sensitive documents will close down and encrypt. Under hostile network attacks, such as Bluejacking (where a Bluetooth network is secretly connected and compromises the machine), clear, unencrypted data is proactively secured.

4.3 Object Security

Having the client know its location also protects appliances and other objects in the world. Any networked appliance or device (autonomous vacuum cleaners, robot dogs, and any other WiFi enabled device) would know its *home* locations. If a device is taken outside its home, its behavior would change. An autonomous vacuum knows to tread more cautiously at an unknown location, a robot dog could become mean outside its home, email home, or cease to operate.

5 Future Work

MediaPoint set out as a tool for artistic empowerment; bringing films to people, based on where they are and no matter where they are. Linked neither to a gallery nor broadcast on a strict time schedule, it is an invisible installation with the location as its medium. MediaPoint delivers media to any presentation device (laptop, PDA, tablet) proactively to viewers at selected locations. Developed to bring media to viewers in WiFi 'Hotspot' locations (coffee shops, airports, etc.), MediaPoint is a tool for artistic empowerment; creating a new kind of film and viewing dynamic.

Viewers enter a MediaPoint enabled location with a WiFi device and simply run their media player. MediaPoint determines where they are based on their logical location information hidden in the network and plays media specifically for that location. It is all proactive and requires no (typically exhaustive) searching or visual Internet hunt. MediaPoint assists by removing the search entirely, connecting people and media with no friction whatsoever. We are building a larger scale installation towards this end to provide keyless media delivery as well as additional security layers on top of existing Digital Rights Management (DMR) protocols.

References

1. Sood, S., Hammond, K.J., Birnbaum, L.: Low-fidelity location based information systems. In: Proceedings of the 9th international conference on Intelligent user interface, ACM Press (2004) 325–327
2. Grover, D.L., King, M.T., Kusher, C.A.: Reduced keyboard disambiguating computer. U. S. Patent Office, Tegic Communications, Inc., Seattle (1998) Patent Number US5818437.
3. James, C.L., Reischel, K.M.: Text input for mobile devices: comparing model prediction to actual performance. In: Proceedings of the SIGCHI conference on Human factors in computing systems, ACM Press (2001) 365–371
4. Ballagas, R., Ringel, M., Stone, M., Borchers, J.: iStuff: a physical user interface toolkit for ubiquitous computing environments. In: Proceedings of the conference on Human factors in computing systems, ACM Press (2003) 537–544
5. Flachsbart, J., Franklin, D., Hammond, K.: Improving human computer interaction in a classroom environment using computer vision. In: Proceedings of the 5th international conference on Intelligent user interfaces, ACM Press (2000) 86–93