Wireless access to a content routing system

Junbiao Zhang
C&C Research Laboratories, NEC USA Inc.
4 Independence Way, Princeton, NJ 08540
junzhang@nec-lab.com

Remo Strotkamp
C&C Research Laboratories, NEC USA Inc.
4 Independence Way, Princeton, NJ 08540
remo@nec-lab.com

ABSTRACT

We present in this paper our experience in providing a wireless access interface to a distributed content search and delivery system. Important Wireless Application Protocol(*WAP*) gateway functions, in particular the Push function, are integrated into the access points of the system. Wireless devices can use this interface to search for distributed information and get notified of relevant new content. A prototype implementation of the integration has been completed which works successfully with *WAP* 1.2 compliant clients.

Keywords

WAP gateway, Push, Kannel, ANSWER system, content routing

1. INTRODUCTION

The Wireless Access Protocol (*WAP*) is a suite of protocol solutions for providing web content to wireless devices. It enables wireless devices with small screen sizes and slow, lossy communication channels to access web content in a more efficient and natural way. In this paper, we describe our experience in integrating the *WAP* framework into the *ANSWER* information system[2]. *ANSWER* is a distributed content search and distribution system which provides an infrastructure directly connecting information providers and consumers. A symmetrical interaction model allows users to directly search for results or get notified by the system through content distribution based on prior interest indication. The core of the system serves as a distributed information base and search engine which can be built upon either an active network or an overlay application layer network. On the edge (access points) of the system, a consistent interface is provided for both content provision and search result delivery through the use of *XML*[4]. For search users, both web access and *WAP* access are provided for submission of queries as well as display of search results and new content alerts. In order to allow for wireless access, *WAP* gateway functions are introduced into the *ANSWER* system. In particular, the push feature of *WAP* is used as a natural means to distribute new content to interested clients. We discuss in the rest of the paper how we build the most essential gateway functions into the *ANSWER* system and show through examples how wireless devices interact with the system through the *WAP* interface.

2. ANSWER: CORE AND EDGE

Due to space constraint, we shall only give a very brief overview of the *ANSWER* architecture. Interested readers can refer to [2] for a detailed description of the system. In essence, *ANSWER* is an information routing system where its core infrastructure serves as a distributed directory base to route query and content packets. Unlike in the case of conventional layer 3(e.g. IP) routing, these packets do not have to specify explicit destination addresses. The underlying network automatically directs packets based on the search or distribution criteria contained in the packets. This is achieved through the following key system components:

- An ontology based hierarchical semantic structure for content management.
- A dynamically constructed routing tree for directory information distribution.
- Content routing modules at each *ANSWER* node for directory management and packet forwarding.
- Intelligent agents in the packets for customized application routing.

On the edge of the system, *XML* is used as a uniform interface for content specification. Through the use of *DTDs* and *XSL*[5] files, content data with application specific syntax and semantics are converted into directory indices for distribution. These indices are eventually propagated into the *ANSWER* core through periodic routing information exchange among the *ANSWER* nodes. For the end users, *ANSWER* edge nodes provide interfaces for both web browsers and *WAP* enabled wireless devices. These interface modules are implemented as Java Servlets. User queries are transformed into *ANSWER* format through *JavaScript* or *WMLScript*. They are then injected into the *ANSWER* system and aggregated into query indices for distribution into the *ANSWER* core. These indices serve as indication of user interests which are used to direct content packets. Search results, when routed to the edge nodes, are converted into *HTML* or *WML* format using application specific *XSL* files.

3. WIRELESS ACCESS TO ANSWER

In order to support wireless access to the *ANSWER* system, we extended the *ANSWER* edge nodes with functions which enable a smooth and efficient *WAP* interface. These include:

- Identification of wireless devices
- Integration of *WAP* gateway functions
- Conversion and organization of search results
- Delivery of new content through *WAP* Push[3]

While the first function involves nothing more than checking the related HTTP headers about the *WAP* agents, the other functions are worth some extra explanations.

3.1 WAP gateway integration

Since the *ANSWER* edge nodes have a web interface, it is possible to use an arbitrary *WAP* gateway to access the *ANSWER* system. However, given the simple functions that wireless devices perform to interact with the *ANSWER* system, it is unnecessary to have a full featured *WAP* gateway. Further, directly integrating some of the *WAP* gateway functions into the *ANSWER* edge nodes makes the wireless accesses more efficient, as search requests and content deliveries do not have to go through some centralized gateways which could become system bottlenecks. As an example to these arguments, consider the *WAP* Push feature and its use in the *ANSWER* system which we will elaborate in a later section. It is the most important feature for our system since it serves as a means to deliver new content to interested users. However, if we rely on a Push gateway to notify users each time there is new result available at an edge node, the gateway will be flooded with PAP (Push Access Protocol) requests. If instead we build the Push gateway function in the edge nodes, not only can the bottleneck problem be alleviated, but the PAP protocol can be eliminated all together. This is because each edge node is the only source of Push requests for its gateway and thus there is no longer the need for external protocol interactions. For our purpose, we used the open source Kannel *WAP* gateway package as the reference. Since Kannel was implemented in C and did not support the Push function, our integration effort involved both re-implementing features in Java and building the gateway functions.

3.2 Search result conversion

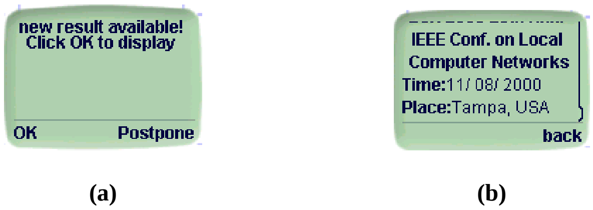
As we mentioned earlier, the query results from the *ANSWER* system are coded in *XML* format for easy conversion into different display languages using *XSL*. For *WAP* devices, results are converted into *WML* cards where each result item corresponds naturally to a single card. Given the distributed nature of the *ANSWER* system, the results of a search are delivered to the user progressively as content summaries from different parts of the system become available. With the help of the edge nodes, the whole result sets can be displayed properly by the wireless devices. The edge node groups the results as they become available. Based on certain threshold(e.g. time passed or number of results available), it delivers the initial set of results as a deck of cards to the wireless device. The last card in the deck would ask the user whether he/she wants to see more results. More new decks from the edge node will then be fetched if the user decides to continue. Otherwise, the temporary storage of the result set is erased from the edge node.

3.3 New content delivery

One of the distinctive features of the *ANSWER* system is the symmetrical interaction model between content providers and consumers. That is, not only can query packets be routed to the proper content destinations, but content packets can be delivered to users who expressed their interests before. New content notification is thus an integral part of the wireless access. From *WAP* 1.2 and higher, a Push function is specified. It allows a user to send information asynchronously to a *WAP* enabled device. To do so, the user sends a request containing the id of the intended device to a Push gateway. The gateway then locate the device and deliver the message. With this push feature, we can very smoothly deliver new content notifications to *WAP* users. As a very important part of our *WAP* integration, we have implemented the Push gateway functions in the *ANSWER* edge nodes. When a *WAP* device submits an *ANSWER* query to an edge node, its device id together with the query id are stored at the node. When new results are available at a later stage(after certain time threshold), if the query has not been canceled by the user, the results are converted into a *WML* deck, a Push message containing a link to the deck is then sent to the device. Such a message will be either delivered as an alert or put in the user's mail box.

3.4 Examples

One of the applications we developed on the *ANSWER* framework is the conference announcement and search application. Conference organizers can announce the conferences using a special *XML* format. A user can search for specific conferences through a web browser or a *WAP* device based on different criteria. For *WAP* devices, search results will be returned as a deck of cards with each card representing one conference. Users may also be notified of relevant new announcements through alert messages on the *WAP* devices. Figure 1 shows the screen shots of a *WAP* phone when new result becomes available and the user chooses to view the result. Since we are not aware of any Push capable *WAP* phones, the *WAP* phone in this example is a simulator in the Nokia *WAP* Toolkit.


new result available!
Click OK to display

IEEE Conf. on Local Computer Networks
Time:11/08/2000
Place:Tampa, USA

(a) **(b)**

Figure 1: New result display

4. STATUS AND FUTURE WORK

We have finished integrating the basic *WAP* gateway functions into the *ANSWER* system. A prototype system which uses the *ANSWER* framework has been tested with both *WAP* phones and *WAP* simulation softwares. We are investigating various aspects of mobility management for wireless access in the *ANSWER* system. In particular, we plan to integrate support in the *ANSWER* system so that 1) mobile devices can locate the "nearest" edge nodes, and 2) user sessions can remain uninterrupted, e.g. search results can still be properly delivered even when the user moves or temporarily disconnects from the *ANSWER* system.

5. REFERENCES

[1] WAP Forum, "Wireless Application Protocol Architecture Specification", http://www1.wapforum.org/tech/terms.asp?doc=WAP-100-WAPArch-19980430-a.pdf

[2] J. Zhang and M. Ott, "ANSWER: Information Routing Based on Active Networks", The 5th Asia Pacific Conference on Communications, Beijing, China, October, 1999

[3] WAP Forum, "Push Architectural Overview", http://www1.wapforum.org/tech/terms.asp?doc=WAP-165-PushArchOverview-19991108-a.pdf

[4] W3C, "Extensible Markup Language(XML) 1.0", http://www.w3.org/TR/1998/REC-xml-19980210.html

[5] W3C, "Extensible StyleSheet Language(XSL) 1.0", http://www.w3.org/TR/xsl/

[6] T.R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing", Padova workshop on Formal Ontology, Mar, 1993