Developer Relations Group

Hybrid Application Cookbooks

Push Technology

Intel Corporation
Developer Relations Group
www.intel.com/drg

Disclaimer

Information in this document is provided in connection with Intel products. no license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in Intel's Terms and Conditions of Sale for such products, Intel assumes no liability whatsoever, and Intel disclaims any express or implied warranty, relating to sale and/or use of Intel products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right. Intel products are not intended for use in medical, life saving, or life sustaining applications. Intel may make changes to specifications and product descriptions at any time, without notice.

TABLE OF CONTENTS

1. Introduction

The user experience on the Internet today can be both overwhelming and tedious. With the staggering volumes of content on the Web, users often find it difficult to locate information of interest even when using sophisticated search engines. Compounding the problem is the relatively poor bandwidth and latency of the Internet. For most users, the Internet's manual search process is barely tolerable and certainly far from compelling. Driven by the need to simplify and tame these content delivery problems, "push" technologies have emerged.

One of newest phenomena to sweep the Internet, push technology provides a mechanism for content publishers to overcome the chaos, noise, and congestion on the World Wide Web. Also referred to as "netcasting," "webcasting," "channel broadcasting," "personal information delivery," and "information on demand," push technology reverses the Internet's content delivery model. Before push, content publishers had to rely upon the end users' own initiative to bring them to a website to view or download content. With push technology, the content publisher can deliver its content directly to the user's PC, substantially improving the likelihood that the user will view the content.

Push technology borrows some of the concepts used in traditional publishing, such as the subscription model, to streamline the delivery of content to the end user. Of course, using computer technology provides a more dynamic delivery mechanism than that afforded by the postal service or the neighborhood paperboy. Push content can be extremely timely, delivered fresh several times a day. In addition, push content can be customized to the specifications of each individual end user.

Why is Intel interested in push? Push, along with the Connected CD, Intercast™ and Multiplayer Internet Gaming, is one of the Internet hybrid technologies that integrate the capabilities of the high performance PC with the Internet. Push technology facilitates the delivery of multimedia content on the Internet through the use of the local storage and transparent content downloads. In combination with content delivered on CD-ROM, push technology is an excellent mechanism to deliver content updates to the end user.

The purpose of this document is to help software and content developers to incorporate push technologies into their content delivery strategy. This document will answer the following questions

- What is push technology?
- How does push work?
- What are the important features of push?
- What are the future directions for the push category?

1.1 Intended Audience

Who can benefit from this document?

- ***Website Developers***--those already publishing content on the Internet today. These developers may be interested in using push applications to more effectively bring consumers to their content, or to deliver their content to the consumer.

- ***Content Developers*** - those that publish content in any form, not necessarily on the Internet.

- ***Independent Software Vendors*** (ISVs)--Software application developers that create content-oriented titles including consumer reference CD-ROMs and games. Push technology offers ISVs the capability to continuously update the customer with the latest version and fresh content.

2 What Is Push?

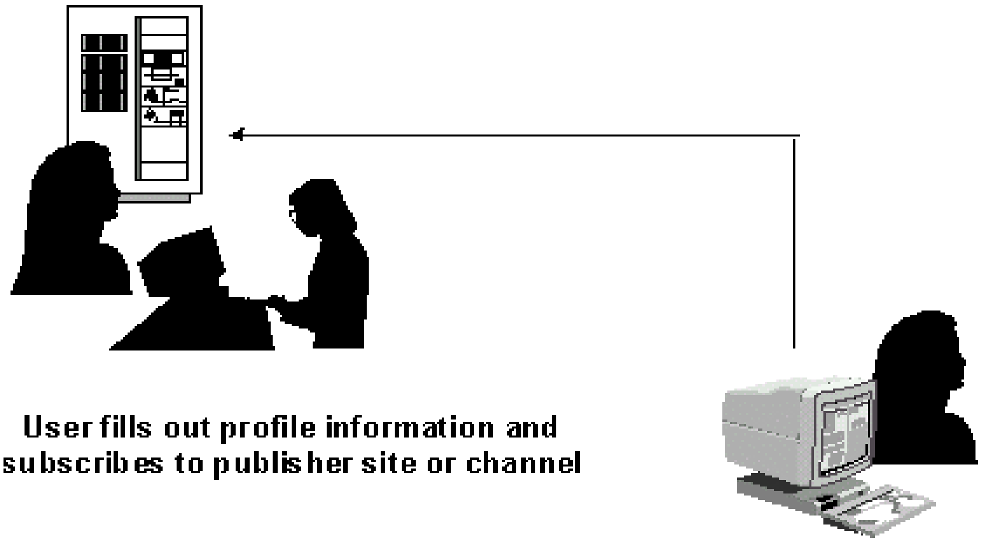
Push was created to alleviate two problems facing users of the Internet. The first problem is information overload. The volume and dynamic nature of content on the Internet is an impediment to users and has become an ease-of-use issue. The second problem is that most end users are restricted to low bandwidth Internet connections, such as a 28.8 kbps modem, making it difficult to receive multimedia content. Push technology addresses both of these problems as follows:

Information Overload: Using the Internet today, without the aid of a push application, can be tedious, time consuming, and less than dependable. Users have to manually hunt down information, search out links (live and dead), and monitor sites and information sources. The advent of search engines such as Yahoo* and Alta Vista* have met with tremendous success because they make it possible for the user to narrow the focus and expand the domain of information searches. Push applications and technology building blocks narrow that focus even further and add considerable ease of use. Using push technology, an electronic publisher with content aggregated from multiple sources applies the subscriber's interest profile to select information to deliver to the subscriber on an "automatic" basis. For example, a news bureau can deliver articles of interest to you in electronic format to your PC each morning--or at any time you require.

Low Bandwidth: The bandwidth of the 28.8 kbps modem provides very limited capability to deliver multimedia content. Assuming users are willing to wait 15 seconds for a web page to download means that the combined size of all elements on the page must be less than 45 kbytes. This size limit leaves room for little more than text and small graphics. However, push technology provides the means to pre-deliver much larger packages of content. With an unattended connection for a half hour the previous night, an end-user can receive up to 5.4 megabytes of content to view the next day. This leaves plenty of room for multimedia content such as audio, large graphics, or short video clips. Push technology enables multimedia on the Internet.

2.1 The Push Process

Figure 1: Push Subscription


User fills out profile information and subscribes to publisher site or channel

For the end user, the process of receiving push content is very simple. First, an individual subscribes to a publisher's site or channel by providing the publisher with an information profile containing demographic information as well as content preferences. The subscriber also sets up a schedule specifying when information should be delivered.

Based on the subscriber's schedule, the PC connects to the Internet and the client software notifies the publisher's server that the download can occur. The server collates the content pertaining to that subscriber's profile and downloads it to the subscriber's machine. Upon completion of the content download, the content is available for viewing and the subscriber is alerted.

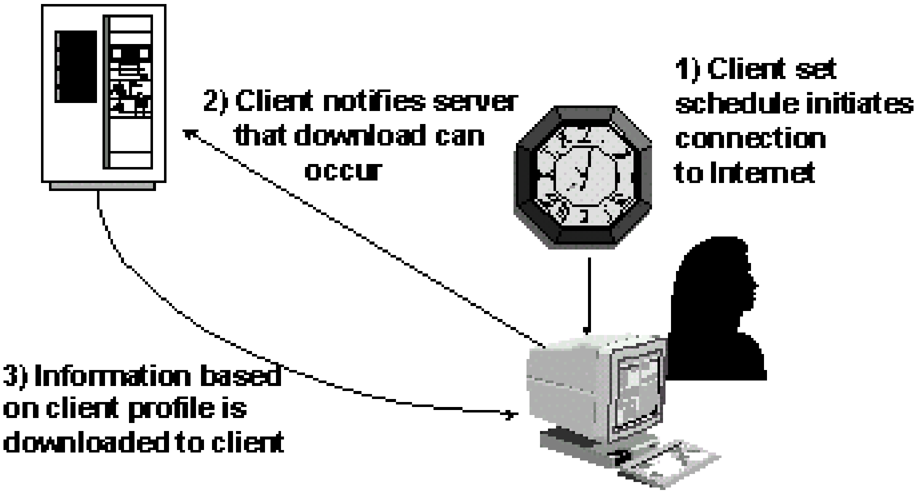

1) Client set schedule initiates connection to Internet
2) Client notifies server that download can occur
3) Information based on client profile is downloaded to client

Figure 2: Push Process Flow

2.2 Push Applications are Great Hybrid Applications

Hybrid applications combine the advantages of both the Internet and the high performance PC to create a richer experience for the end user. The Internet brings timely content, real-time interactivity and multi-user capabilities. The PC has the resources (processor, CD-ROM drive, hard drive, peripherals, etc.) for high performance video, 3D graphics, and caching. The combination is a hybrid application that includes the best of both worlds: an application that is more compelling than either a PC or Internet-only application.

Push is a great example of an Internet hybrid application. Push applications can be used to deliver Internet content to a PC’s own storage resources. Because this content can be delivered in the background, the user does not pay the penalty of waiting for the download. Consequently, larger-sized content such as multimedia can be delivered, taking advantage of the PC multimedia capabilities. The net result is that push applications help to alleviate the Internet bandwidth bottleneck, facilitate the delivery of multimedia and timely content, creating a more enjoyable experience for the end-user.

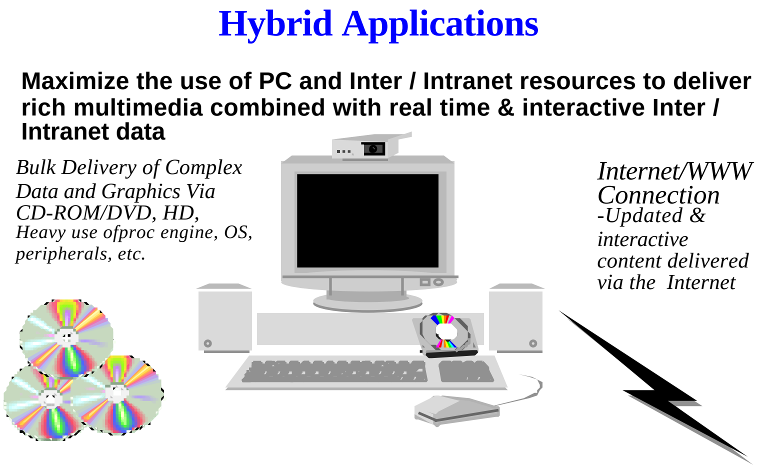

Hybrid Applications
Maximize the use of PC and Inter / Intranet resources to deliver rich multimedia combined with real time & interactive Inter / Intranet data
Bulk Delivery of Complex Data and Graphics Via CD-ROM/DVD, HD,
Heavy use ofproc engine, OS, peripherals, etc.
Internet/WWW Connection
-Updated & interactive content delivered via the Internet

Figure 3: Hybrid Applications

2.3 Comparing Push Applications

With the myriad of push solutions available today it may be difficult to discern the similarities and differences among them. One can slice the push application category in several ways. One way to compare push applications is by their audience focus. Is the application designed to address business or consumer audiences? Another way to compare applications is by their suitability for the delivery of various types of content such as multimedia, software programs, or simple notifications or messages. One may also consider the method used by the application to deliver content.

2.3.1 Audience Focus--Business or Consumer End User

Each push application has been designed to focus on a particular audience. Some are best suited for Intranet use (e.g., within a single organization) as opposed to Internet use (which implies a business-to-end user−most likely a ‘home’ end user--connection). The target audience may also determine or influence the architecture and feature set of the application. For a business audience, the push product may emphasize group communication within a closed, controlled environment. For delivering content to a consumer audience over the Internet, the push application needs to be specifically designed to work with Internet content and scale.

2.3.2 Type of Information Delivered

Push applications also vary in how they deliver and integrate different types of content. In some push applications, the information type is closely coupled with the architecture chosen by the push application vendor.

2.3.2.1 Web-based Multimedia Content Delivery

These push applications deliver web pages, or even entire sites, created specifically for delivery via push technology. The content is generally packed with multimedia and is reminiscent of television content. These applications are intended to provide the user with the same type of edited information available through television or print media. They require a browser--either external or built into a proprietary client. Examples are Pointcast PCN* and Wayfarer’s Incisa*.

2.3.2.2 Controlled or Metered General Content

The emphasis here is on ensuring that a particular version or set of content--for example, software upgrades--arrives on schedule. Users of these applications are most concerned with providing the most bandwidth- and time-effective method of secure and stable delivery of content---any size, any regularity. Marimba's Castanet* and BackWeb* are examples.

2.3.2.3 Intelligent Targeted Notification or Messaging

These push applications deliver headline-style information to entice a user to visit a website. Examples include notices of special sale prices, a headline to a news story, or a reminder of a "live" event that will occur soon. These application do not deliver the entire set of content but rather contain links to related, but more extensive content on the a website.

The content that these applications deliver is usually repetitive in nature and updated quite frequently (multiple times in an hour, in some cases). These applications need not be browser-based, but often are. In a business-to-business scenario this type of service may involve group communications and messaging. In a consumer model, it may be a ticker tape, news or weather update headline field. Examples are Lanacom's Headliner* and Intermind's Communicator*.

*Other brands and names are the property of their respective owners

3 How Does Push Work?

3.1 How Push is Really Automated Pull

The Web currently requires the user to poll sites for new or updated information. This manual polling and downloading process is referred to as "pull" technology. From a business point of view, this process provides little information about a user, and offers little control over what information is acquired. It is the user who has to keep track of the location of the information sites, and the user has to remember to continuously search for informational changes--a very time consuming process.

The push model alleviates much of this tedium. Interestingly enough, from a technical point of view, most push applications are still "pull" and just appear to be "push" to the user. A more accurate description of this process is automated pull. Most push applications require a subscription and an information request profile from the user before they can begin filtering information. They also require the client to poll the server when it is able to accept information. The software initiates the "pull" according to a user-defined schedule (once a day, every three hours, etc.), and the server responds with the information to match the request profile.

There are a couple of true push technology applications--one being Wayfarer's Incisa* product. When the Incisa client is started, it opens a connection to the Incisa server and it stays open until the client is stopped. Another true push technology is AirMedia Live* from Air Media. This technology uses the broadcast radio spectrum to transmit to their client applications. The client application is a combination of a radio receiver and a software application. The radio receiver is always receiving information from the server/transmitter.

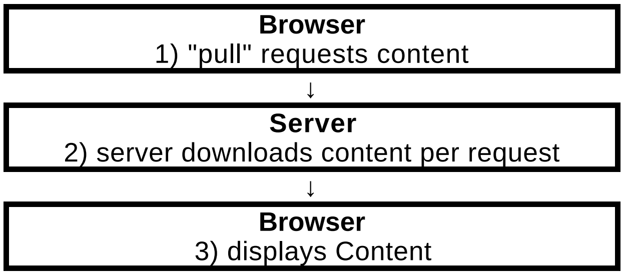

Browser
1) "pull" requests content
↓
Server
2) server downloads content per request
↓
Browser
3) displays Content

Figure 4: Current Environment--Pull

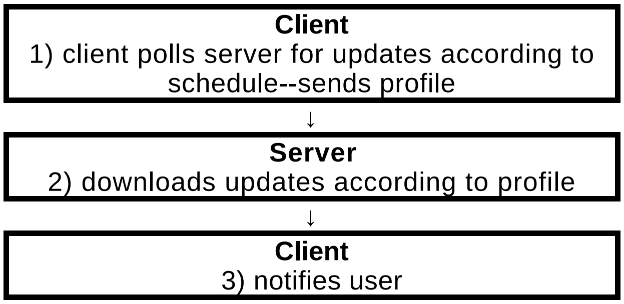
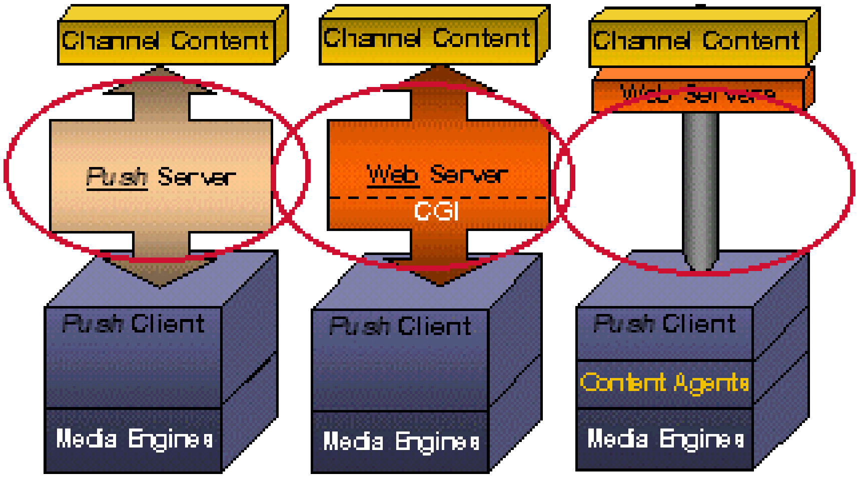

Client
1) client polls server for updates according to schedule--sends profile
↓
Server
2) downloads updates according to profile
↓
Client
3) notifies user

Figure 5: Push as Automated Pull

So if push applications are not really push what is so different about them? The difference is the automation of the process for both the publisher and the subscriber.

3.2 Information Delivery Models

There are at least three (3) information delivery models by which push applications can be categorized. These models differ in the costs incurred for purchase and uptime, the ease of use, the ease of integration into a publisher's existing information delivery structure, the customization capabilities, the branding adherence, and in other areas that a publisher should examine with regard to need.



Push Server Model   Web Server Extension - CGI   Client Agent Model

Figure 6: Information Delivery Models

3.2.1 Push Server Model

The most common delivery model is the PUSH SERVER model. This is a turnkey solution that provides a client, server, and development tools. Costs associated with the server may include the number of connections and/or the number of packets sent. Branding may or may not be an issue. A proprietary client is supplied. These applications may use a proprietary protocol.

3.2.2 Web Server Extension - CGI

A closely related model is the WEB SERVER EXTENSION model. In this model, the push vendor does not buy a server, so the costs are associated with the number of extensions sold. Feedback and demographic information can be directed to an external server, so that information can be retained by the push vendor (transfer costs may need to be negotiated separately). Installation should be less extensive since no true server is involved. No proprietary client is required; these run within the user's regularly installed browser.

3.2.3 Client Agent Model

The final model does not need to have a server at all, except to update client software and retain user demographics. This model uses a “client agent” to retrieve the information from sites. Each agent is designed to provide different search results. This model can allow for an anonymous relationship between the vendor and the subscriber. Costs are associated with the agent customization. The user is responsible for deployment, and controls the search type extensibility.

*Other brands and names are the property of their respective owners

4 Push Technology Features

All of the push applications share a common set of features and capabilities. This basic feature set includes the ability for the user to create an information profile (i.e., to define what information he or she would like to receive.) The content publisher has a similar capability to define which content to deliver to each end user. The client application must also be able to connect, notify the server of its presence, , and receive the information for the user.

Most packaged push applications extend the common feature set considerably. A state-of-the-art content or file-type push system will provide the subscriber with control and messaging options, security, and intelligent use of bandwidth/connection. These push applications include features that are also important key components in ISV application development--features that embody the messaging, control, and transfer mechanisms. These components work in conjunction with client software to automate a poll of known sites and information.

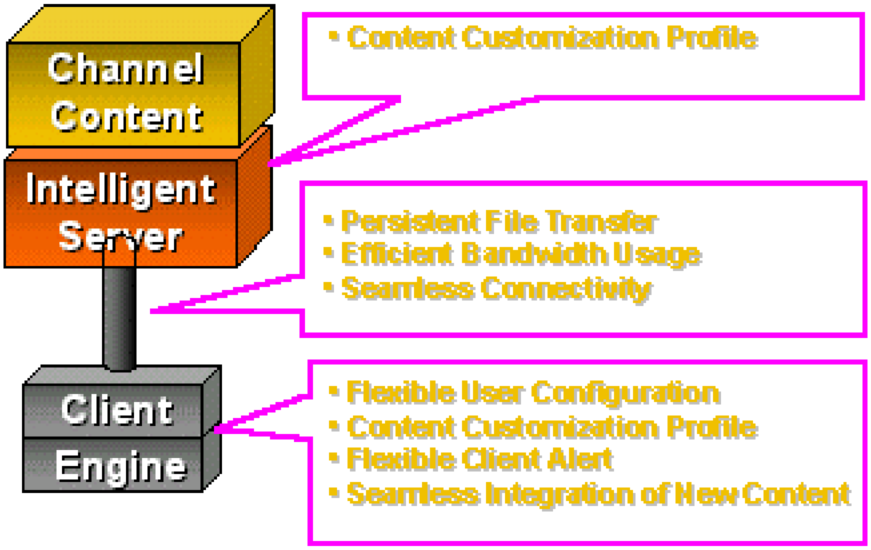

Channel Content
Intelligent Server
Client Engine
• Content Customization Profile
• Persistent File Transfer
• Efficient Bandwidth Usage
• Seamless Connectivity
• Flexible User Configuration
• Content Customization Profile
• Flexible Client Alert
• Seamless Integration of New Content

Figure7: Push Features

4.1 Seamless Connectivity

This refers to the client's ability to <u>easily</u> connect to the Internet.
Users connect to the Internet in many different ways, such as via an online service provider like AOL and CompuServe. They connect using analog and ISDN modems, or using direct connections via corporate LANs. The push client application must be able to recognize all of these connection types and be able to seamlessly establish a connection to the Internet. More sophisticated applications will have the ability to sniff out existing configurations without user input. If there is more that one possible connection method, the push application should provide a mechanism for the user to select or prioritize the connection

method. “Seamless” means establishing the connection with minimal end user interaction or no user interaction in the case of unattended operation such as a midnight download. Unattended operation will become increasingly important for the delivery of larger data packages with multimedia content.

4.2 Flexible User Configuration

This refers to the user's ability to schedule connections and allocate resources. One key to making a push application successful is to ensure that the user is still in control of the PC, and can configure the interaction of that PC with the Internet to provide non-obtrusive collection of information. The publisher or push vendor should supply the user with the ability to schedule the connection time, length, etc. In addition, the client should control what type of information to transfer, the size, where to store it, what to save, what to replace, etc. This feature is becoming increasingly important as recent press reports claim that the population of remote users without permanent connections is increasing in relation to desktop users.

4.3 Content Customization Profile

This refers to the subscriber's customization of his or her information profile and to the publisher's customization of both the content and the delivery parameters of the content.

Subscriber: The ability to select topics and sub topics from a channel publisher or content provider which are used to select and filter information for download.

Publisher: The ability to divide content into categories that will appeal to users, as well as to specify when, how, and for how long content is displayed. Includes the ability to target that information to a specific audience based on user profile information. (Push technology is a wonderful opportunity for publishers because it allows the publisher to accurately target the audience most likely to provide a return on its investment, either through advertising or through the sale of content.)

4.4 Persistent File Transfer

This refers to the ability to stop a data transfer at any point and to continue from that point when the connection becomes available again. This feature is most important when delivering large files. Because push applications excel at the transfer of rich multimedia, many are designed to go one step further than the standard information transfer mechanisms such as ftp or http. Those with proprietary transfer capabilities or close client/server interactions can provide the capability to save file transfer information on the client if the connection is lost. On the next poll for information from the server, the information transfer can pick up where it left off.

4.5 Efficient Bandwidth Usage

Client Centric***:***

The client’s maximizes the use of available bandwidth via the use of idle time downloading. BackWeb* has a technology called Polite Agent™ that can use idle time to download files from the server.

Server Centric***:***

The server specializes in succinct and specialized information transfers, such as Minimal Updates: the ability to send minimal data by re-using components that already exist on the client. One vendor (Marimba*) uses MD5 (Message Digest 5) checksum as a distinct signature for a file. Thus even if two files are named differently, their sameness will be noted and the component will be reused as opposed to downloading an additional copy.

4.6 Seamless Integration of New Content

This refers to the application's ability to transfer and integrate new or updated content with existing content. This replaces the user's ability to download a file and install it. These applications automatically determine what information, or what portion of the information, to download, where to store it, and what to replace. Some can actually execute the replacement data. Pointcast* is the perfect example of an application based on this ability.

4.7 Flexible Client Alert Capability

This refers to the client's ability to notify a user that new information exists or has arrived. This notification can range from simple dialog boxes to task bar icons to animations with sound and video. The type of alert depends on the type and importance of the information being transferred. (If the client alert is not configured properly to the type of data and user preferences, the user is apt to unsubscribe.)

4.8 Security

This refers to the ability to ensure asset integrity and authorship, and that assets are only received by the intended client (or become unusable). Also must ensure that the assets that are invoked on the end user's system are secure within that system, like JAVA.

4.9 Application Protocol

This refers to the IP protocol family, upon which the Internet is based. The two main protocol transports within that family are TCP and UDP. Although TCP is reliable, it's not timely because its main focus is on making sure packets arrive at their destination. It will degrade performance to achieve that goal. UDP cares less about reliability and more about moving packets through a network as quickly as possible. This means that packets may be lost in the process of achieving maximum throughput and it is the application's responsibility to make the transport reliable. Many corporate firewalls are configured to block incoming UDP packets and incoming TCP/IP connections.

Most applications use the HTTP (TCP Port 80) protocol. The advantages of HTTP is that it is able to flow through firewalls (using outgoing connections) and the transmission is reliable based on TCP/IP. Browsers are based on HTTP. The disadvantage is that features to make applications "real-time" are not easily created.

Some push vendors have developed proprietary protocols that use the UDP transport. This means that these applications have difficulty with network environments secured with firewalls, and require special setup--if they can be used at all. (This depends on environment security requirements.) The advantage of proprietary protocols is that they allow some key features to be used--persistent file transfer, efficient bandwidth utilization techniques, etc.--that are much more difficult to implement via HTTP (at this time).

*Other brands and names are the property of their respective owners

5 Using Push Applications

5.1 Using Off-the-Shelf Push Applications

The easiest way to use push technology is to use existing push applications. Independent software vendors (ISVs) can adopt one of the current applications (e.g., BackWeb and Marimba), either without requiring any additional functionality, or by including it in their product. It is possible for a push vendor to deliver a 'private label channel' to an ISV.

TIPS FOR USING EXISTING APPLICATIONS

- Determine what content needs to be pushed(not everything does). Just push the content that users request or the publisher needs them to see. Information that doesn't need to be "timely" may not need to be pushed.
- Determine the content format for most effective presentation.
- Choose a model based on the type of content delivered, but also consider features and model characteristics that fit your business and environment.
- Review corporate branding options--can the application accept your branding on the client as well as the server content?
- If no application or technology fits, think about altering the presentation of your data to use the application more effectively.

5.2 Integrating Push into Applications

There is a compelling case for ISVs to incorporate push technology into their applications. Push technology provides the means to easily update the end-user with new software and content. Now it is much easier for both the software developer and customer to establish a direct relationship. The business model ramifications certainly are intriguing.

One way to integrate push capability into application software is by integrating an off-the-shelf push application. There are issues involved in working with third-party software. For example, an ISV has added dependencies and may not have total control over all the deliverables. In addition, there are trademark and copyright considerations, documentation differences, and support programs to consider. Each ISV must analyze their business needs to see what fits best.

Push capabilities can be developed from scratch, or one can use a Software Developer's Kit (SDK), that provides the component building blocks for push in a documented set of Application Program Interfaces (APIs). An ideal SDK would provide all of the extended push components discussed earlier:

- Seamless connectivity
- Flexible user configuration
- Content customization profile
- Persistent file transfer
- Efficient bandwidth usage
- Seamless integration of new content
- Flexible client alert capability
- Security

Although relatively few SDKs have been released so far, push vendors recognize the need to produce more. As these technologies grow, additional SDKs will become available. Expect a plethora of offerings from vendors who are now scrambling to understand this environment and all of the technologies involved.

6 Future Directions

6.1 Platform Initiatives Affecting Push Technology

6.1.1 OnNow Technology

In order to enable unattended push content delivery during the wee hours of the morning it is necessary for the end-user to leave their PC turned on. Many people are hesitant to do so, concerned over the additional power usage or wear on the system.

The push technology community should take note of the OnNow platform initiative (http://www.microsoft.com/hwdev/pcfuture/onnow1.htm). This initiative involves the industry-wide coordination of PC platform contributors to provide the PC with the ability to

- be ready immediately when the ON button is pressed
- appear to be off when user is not actively engaged but respond to internal wake-up triggers generated by software or external devices such as the keyboard, mouse or phone
- have operating system and application software coordinate efforts to deliver effective power management
- have devices respond to and be affected by power state changes

How Push Applications Will Be Affected:
Push applications and technologies will want to tie their scheduling capabilities to the OnNow extension APIs. OnNow permits off-hour and non-supervised Internet access and maintenance, while facilitating effective power management by returning the PC to its original power management mode.

OnNow enabled PCs also makes it easier for push subscribers to access the content immediately when they turn on their systems.

6.2 Browser Push vs. Stand Alone Push Applications

Push technologies will drastically change the way consumers access and use both business and entertainment information and software. Marimba, BackWeb, PointCast, Intermind, Wayfarer, and a host of other third-party push vendors have created a baseline of functionality not currently addressed by browsers. Microsoft and Netscape have both responded by describing future products that incorporate push technologies. In the final analysis, it is probably the desktop that can most influence the structure and success of push technologies. Third-party vendors will try very hard to distinguish their products through value-added features. Both Microsoft and Netscape's desktop architectures will provide plug-ins for third party push vendors.

6.3 Microsoft

With Internet Explorer 4.0, Microsoft introduces a new built-in push capability called "Webcasting." Webcasting provides to each user automatic delivery and offline access to website content.

The core of Webcasting is Microsoft's proposed Internet broadcast standard ***CDF*** (Channel Definition Format). Microsoft has submitted CDF to the World Wide Web Consortium (W3C). Proponents claim that CDF will eliminate the need for multiple browsers, plug-ins, or proprietary clients to receive pushes from information delivery vendors. CDF is designed to turn any Web server into a push channel without additional work or technology, in effect creating push channels that are, in essence, automated pull from the client. CDF is an application of the Extensible Markup Language (XML). XML makes it possible to add new markup types to an HTML browser without having to add new HTML tags. More than 30 tool, media, and service companies are backing CDF.

Webcasting is an open, scalable solution that works with any HTTP server, any HTTP proxy, any HTML website, and any Web authoring tool, and scales up to multicast push solutions. Webcasting enables any existing website to be "pushed" without requiring any re-authoring of or modifications to the site. This "push,"--or "smart pull"--is accomplished by crawling the site on a scheduled basis. Content authors can then optimize and personalize this Webcasting experience by authoring a "Channel," i.e., by creating a single file that indexes the content on an existing site. This file uses the CDF file format, for indexing and "pushing" structured content on the Internet.

For more information see: http://www.microsoft.com/ie/ie40/press/push.htm.

6.4 Netscape

Netscape has announced a new component in the Netscape Communicator* suite called Netcaster (previously known by the code-name Constellation). Netcaster enables push delivery of information to the desktop and offline browsing.

With Netcaster, any channel can be anchored to the desktop, making it a webtop. A webtop does not integrate with the Windows desktop like Internet Explorer 4.0 does because it is meant to be a cross-platform network interface, not a proprietary, Windows-only interface. The content for webtops is based on HTML and any channel can become a webtop. With webtops, content providers and OEMs can take over desktop real estate that traditionally has been owned by the operating system. Part of the Netcaster architecture includes the Netscape Channel Finder. Channel Finder is the user interface for accessing channels on the Internet. Channel Finder is dynamically served off the Netscape website, so users can subscribe to a list of channels right from the Netcaster user interface.

Netscape's Netcaster differs from Microsoft's CDF technique in that netcasting is implemented with the existing open standards of HTML, Java, and JavaScript. Netcaster does not rely upon a CDF file. Netscape matches the CDF functionality by allowing content developers to use JavaScript to specify netcasting parameters.

For more information about Netscape's Netcaster technology see:
http://www.netscape.com/flash1/comprod/products/communicator/netcaster_faq.html

*Other brands and names are the property of their respective owners

7 Summary

As the World Wide Web grows in size and complexity the importance of new information delivery models such as push increases. The push delivery model provides an easy way for end users to get the information they want from the Internet and creates a more effective way for content publishers to reach users. The Internet is a great way to deliver timely information, and push technology carries timely content delivery one step further.

Push is one of the Internet hybrid technologies (as are the Connected CD, Intercast™ and Multiplayer Internet Gaming). Hybrid applications are those that integrate the capabilities of the high performance PC with the Internet. Push technology helps to enable the delivery of multimedia content on the Internet through the use of the local storage and transparent content downloads. Traditional applications can be enhanced with push technology, providing an excellent mechanism to deliver new content and content updates to the end user.

8 For More Information

For additional information, please review the Hybrid Developer Website at http://developer.intel.com/drg/.