Ubiquitous Content Formulations for Real-Time Information Communications

K.L. Eddie Law[1] and Sunny So[2]

[1] Ryerson University, Electrical and Computer Engineering
Toronto, Ontario, Canada M2K 2Y2
eddie@ee.ryerson.ca
[2] ViXS Systems, Inc., Toronto, Ontario, Canada M2J 5B5
sunny.so@utoronto.ca

Abstract. With rapid advancements in wireless devices, ubiquitous computing seems becoming a reality everyday. Active pervasive network infrastructure has been introduced to offer selective and intelligent information communications according to access bandwidths of end users' devices. In this paper, research has been extended to deal with mixture of critical and non-critical information. Data classifications are devised for the infrastructure to interpret the importance of data accurately. The design relieves computation requirements at end devices, and mediates delivered information based on users' personal preferences. Further, the operations of content adaptations are transparent to all end devices, and users always perceive high quality transactions and are satisfied with offered network services. More importantly, the resulting design further improves overall system throughput and delay performance.

1 Introduction

Portable wireless devices are widely deployed and used nowadays. More and more people are actively involved in communications while traveling. The ubiquitous computing[3] paradigm [1] appears to be a reality sooner everyday. Weiser [1] indicated that "the most profound technologies are those that disappear." In pervasive computing [1]-[3], information should permeate through networks to seamlessly reach an end user with the highest degree of embeddedness and mobility. When a person moves, his or her portable device can encounter wireless access links with fluctuating communicating bandwidths. Definitely, user always wants all active connections remain connected through the Internet without any interruption due to the mobility. Further, the size of a portable device is usually small, and has limited power supply with constrained computational resources. Hence, it is likely too demanding to add the function of content adaptation into a portable device. We should reduce the amount of computations at a device for extending the operating duration of a battery. Besides, the quality of a wireless

[3] Ubiquitous computing and pervasive computing are used interchangeably in the paper.

network connection changes frequently, dramatically, and without warning. The embeddedness feature (e.g., the connecting interface) of a mobile device may have altered (e.g., from wireless LAN to cellular network) without informing the user. Applications must cope with these changes automatically. Appropriate designs should be carried out to make a pervasive computing system context-aware. It is always a challenge to identify a reference operating platform upon which disparate potential solutions are coalescing for pervasive computing.

Active pervasive network infrastructure (APNI) [4] has been proposed to offer proactive transparent operations for pervasive computing. The infrastructure handles abruptly changing network conditions instantly based on users' preferences, and strives to retain the integrity of information by adapting the content using available computing resources on overlays or in networks. The network infrastructure should be augmented with components and algorithms for providing high-performance pervasive computing, and the goal is to provide desired interpretable content information at destinations with the highest accuracy.

Accuracy of content adaptation should be based on appropriate classifications of information. In the paper, operations on classified data on per user's basis will be examined. The criteria on executing transparent real-time adaptation are to make service subscribers perceive high-speed data transmissions, high quality transactions, and are satisfied with the services provided by the service and content providers. Thorough analysis on achieving these criteria will be reported.

2 Transaction Procedures and Data Classifications

A brief review of the active pervasive network infrastructure (APNI) [4] is discussed. The system architecture is shown in Fig. 1. It can be constucted using agent technology, or programmable node (active networks) [6, 7]). Programmable agents at the edges of networks and programmable nodes inside networks virtualize the interactions between any end users' devices and content providers. The design objective is to mediate delivered information according to a user's personal preferences and its dynamic content adaptation operations are totally transparent to both the content providers and mobile devices.

Agent technology constitutes an important part of the pervasive network infrastructure. All end users' computing devices connect to the Internet through access routers where the agents always locate. For an active communication session, a sender connects to a router with an ingress agent, whereas the egress agent is at the router that connects to the receiver. Indeed, an access router always contains both the ingress and egress agents for monitoring all connections in and out the Internet. At these points of attachments, the egress agents monitor the quality changes of devices' connecting interfaces; while the ingress agents administrate the operations inside the network infrastructure for appropriate content adaptations.

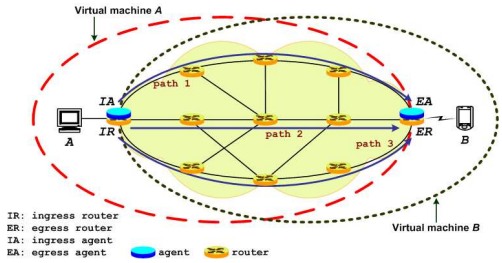

Fig. 1. Active pervasive network infrastructure.

2.1 Desirable Client/Server Transactional Model

Data transactions are usually transported using Transmission Control Protocol (TCP) [8] sessions. The associated three-way handshake procedures and sequence number acknowledgment operations at the transport layer offer the reliability that we are seeking for on data communications. The slicing window mechanism allows proper amount of data delivery under different network conditions. However, when it comes to user's application, the data encoded and decoded at the application layer may or may not be able to change. For example, images can be re-compressed but not one bit of critical data can be removed. As a result, application layer notification should be sent, for example, through a new request update message as described in Fig. 2(a).

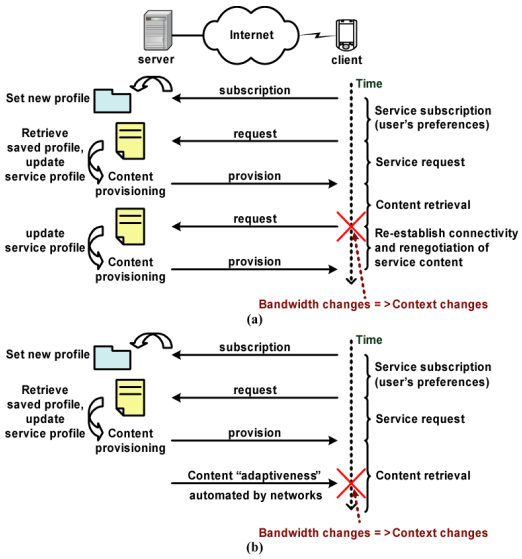

Fig. 2. Service subscription: (a) regular updates, (b) desirable mechanism.

Indeed, a client usually provides personal preferences when paying for and subscribing a service content from a content provider. The preferences can be considered the quality of service (QoS) requirements of a user at the application layer. Then when the client sends a request message to retrieve subscribed information, the provider should deliver content information according to subscriber's contract. Unfortunately, complete information delivery may be forbidden due to the latest network condition. This implies only context change can possibly complete the expected transaction. One possible method is to design a renegotiation process [5] for changing system parameters according to the new network conditions. Unfortunately, this procedure may not run automatically. Furthermore, a client may sometimes need to have certain technical knowledge to pass the new settings to the provider. Therefore, the operations may not be user-friendly and may be time consuming.

As a result, we propose a transactional model at the application layer for the pervasive network infrastructure. Fundamentally, the three-step operational procedures (i.e., the subscription, request, and provision stages) should be kept in application transactions. It is the coordinations of the peer agents for carrying out the "adaptiveness" features in the proposed framework as shown in Fig. 2(b). However, how can the pervasive network infrastructure know what to change in context?

2.2 Content Description Protocol

Context-awareness adaptation can be assisted with content classifications. However, it is always the variations of network resources that lead to the needs of content adaptations. Actually, the bottleneck links are usually the wireless downstream links connecting from egress routers to portable devices. These devices are owned by service and content subscribers. A user sends a short request message to obtain the subscribed information, that may consist of a large chuck of data. Furthermore, the requested information may possibly consist of multiple connections. For further advancement in future network transaction operations, it is desirable to introduce the concepts of content classifications with a transactional operating model.

In the protocol design, when a user subscribes certain services, personal preferences should be posted to content providers. An appropriate content description protocol may need to be designed to take advantages of the design. A three-step procedure is shown in Fig. 2(b). For example, a subscription message is sent to a TCP port number (e.g., port XYZ) defined at provider's server. The packet format should be standardized or well-defined for any content and service providers to undergo content adaptation for pervasive computing. The egress agent on a router carries out two functions: 1) intercept, and 2) interpret the request message. The functions of the subscription, request and provision messages in Fig. 2(b) are outlined with basic fields depicted in Table 1. Whenever a subscription packet goes to port XYZ is received at an edge router, the agent intercepts the message. Firstly, the egress agent appends the bottleneck link's parameters, e.g., the available residual bandwidth and measured/estimated

Table 1. Transaction protocol messages.

Message name	Function and Fields	
Subscription	Carries subscriber's preferences	
	service port number	service provider's defined port number for the service
	content number	service provider's defined content number
	content nature	real-time critical data
		real-time non-critical data
		non-critical data
		images (are preferably kept)
		images (can be removed)
	service type	must receive
		preferably receive
		can receive
		can drop first
	(more 'content number, content nature, service type' fields)	
Request	A user initiates a service request to provider's port number	
Provision	Subscribed service content is being delivered in "provision" packets	
	'content number, content nature, service type'	

delay. The appending operation is to inform the ingress agent connecting to the service provider to administrate proper content adaptations inside network infrastructure. Conceptually, the two agents are negotiating QoS settings for the application when content adaptation is needed. It operates differently from traditional end-to-end adaptation schemes. If active network paradigm is used, the active routers between the two agents update the constraint parameters, if appropriate. When the subscription message arrives at the server's ingress router, the ingress agent removes the appended network constraint information, and delivers the original subscriber's subscription message to provider.

The ingress agent should have registered the network costraint parameters and user's content preferences. Subsequently, it can interpret a subscriber's subscription contents for administering the operations in infrastructure to distribute pervasive computations for the receivers. Typically, an ingress agent do not undergo any pervasive computations to avoid itself becoming the computation bottleneck in networks. Hence in the proposed model, messages are intercepted, examined, and modified, if needed, by agents in accordance to the client context, network service availability, user preferences, and network QoS situations. This eliminates the need for a client to be aware of the complexity of networks, thereby maintaining a server virtual machine model. On the other hand, the request and provision messages are simplified since QoS services at the application layer and content adaptations are provided by transit servers or routers (through the delegation from the agents) in lieu of the server. This hides the requirements from the clients, thereby achieving a client virtual machine model.

2.3 Data Classifications and User's Preferences

Content in transit being adapted to access context and available resources should have the involvements of end users' decisions. Otherwise, the adaptation process cannot provide the highest perceptual and perceived values to end users. Human, physical and cognitive environmental factors affect the tailoring of a

computational model to offer ubiquity, embeddedness, and adaptiveness. Some setup parameters are shown in Table 1. In this section, more thorough discussion on designing user's preferences should be made for implementation in active pervasive network infrastructure. Basically, we should consider the

- real-time constraint:
  - if yes, then its expected value should be set;
- criticality of information:
  - if yes, then function of reliability should be enabled;
- ranking of information:
  - $R_1 > R_2 > \cdots$ (four ranking classes in testbed, see Table 1),
  - R_1 belongs to the "must receive" class,
  - may be ranked by providers to avoid disputes;
- allowed cognitive distortion of images, voices and video:
  - if accepted, then the rate of distortion should be indicated;
- preferred presentation language:
  - for example, English, French, German etc.

2.4 Content Adaptation for Real-Time Delivery

Mobile users always expect information being retrieved to arrive briefly after they send the service requests. For example, many servicemen rely on real-time service tickets to determine the next service locations. Thus, real-time communications are more desirable for certain mobile users. Therefore, the pervasive computing system should provide an expedited delivery service as an important service feature to end users. In this design, a user marks down W as his or her limit of patience on waiting time. Or it can be considered as the maximal time limit within which the content allow a user to make reasonable interpretation, and it may have an acceptable perceptual value when received. The W is called expected real-time constraint.

When a path is given, a client needs to send a request to retrieve subscribed information. Suppose that the total round-trip delay between client and server is T. Although it should rarely happen that the expected real-time constraint of a client is smaller than the round-trip delay, i.e., $W < T$, the server service can never satisfy a client's expectation traditionally. But with the content adaptation mechanism, the size of receiving information can be changed, supposingly, based on the ranked information in data. Consequently, it may possibly satisfy the W requirement. In the following, let assume the receiving data can be truncated or compressed for developing an algorithm. In ubiquitous computing, the last mile (i.e., the wireless access link) is usually the slowest link in a connection. The size of packet should then be adapted to go through the bottleneck link and the resulting size is denoted as s_A.

Since high-speed connection is usually arranged for high-performance server system, the propagation and transmission delays between server and ingress agent, and the processing and queueing delays at server are assumed to be negligible. Besides, the processing and queueing delays at client are ignored while

the downlink delay from the egress agent to client may likely be the bottleneck parameter. Suppose $t(\cdot)$, $p_p(\cdot)$, $p_c(\cdot)$, and $q(\cdot)$ are the transmission delay, propagation delay, processing delay, and queueing delay, respectively.

Further, the $t(ER)$ denotes the transmission delay of a packet from egress agent to receiver, i.e., $t(ER) = \frac{s_i}{B}$ where s_i is the size of packet i and B is the downlink bandwidth of access channel. Similarly, IE and SI indicate the values of a variable from the ingress to egress agents, and from sender to the ingress agent, respectively. The notations are swapped in the reverse direction.

There may have multiple paths between any two agents. But when a path is given, the associated $p_p(\cdot)$ is fixed. Furthermore, if the size, s_i, is unchanged, then the $t(\cdot)$ is also fixed. However, the sizes of packets may get modified and adapted to constrained network resources. Consequently, $\sum t(\cdot)$ can be a varying parameter. Similarly, the programmable nodes may have different processing times for packets with different data and program code. Then, the parameters, s_i and $\sum p_c(\cdot)$, also vary.

Round-trip time, γ, measurements are performed between agents regularly. The sizes of measurement packets are set to be minimal; therefore, we obtain a baseline measured reference result which is

$$\gamma = \sum_{i=\{EI,IE\}} \{t(i) + p_p(i) + p_c(i) + q(i)\}\,. \tag{1}$$

Hence for a transaction with content information, the size of a packet may vary with additive transmission and processing delays. The total delay is then

$$T = t(RE) + t(ER) + p_p(RE) + p_p(ER) + \gamma + \sum_{i=\{EI,IE|data\}} \{t(i) + p_c(i)\}\,. \tag{2}$$

The $t(RE)$ is also negligible for the sent request message from client. Since the downlink is the bottleneck, we have $t(ER) = \frac{s_A}{B}$ upon carrying out content adaptation. Hence, we can bound the the total delay given in Eqn. (2), i.e.,

$$W > \frac{s_A}{B} + p_p(RE) + p_p(ER) + \gamma + \sum_{i=\{EI,IE|data\}} \{t(i) + p_c(i)\}\,. \tag{3}$$

Therefore, we obtain the resulting estimated size of content adapted packets which is

$$s_A < B \cdot \{W - p_p(RE) - p_p(ER) - \gamma - \sum_{i=\{EI,IE|data\}} [t(i) + p_c(i)]\} \tag{4}$$

$$< B \cdot \{W - \gamma\}\,. \tag{5}$$

To deliver content in real-time, the total delay across the networks should be smaller than W as shown in Eqn. (3). Acutally, the sizes and number of packets sending through the networks may be reduced noticeably. The extra processing delay can possibly be predictable from a user's profile. Indeed, the ingress agent arranges proper operations within networks based on the calculated upper bound on s_A as shown in Eqn. (4). By reducing content size as indicated, real-time delivery of information can be achievable.

3 Experiments and Discussions

A prototype testbed has been set up to validate the adaptive designs of the pervasive network infrastructure. All routers are Pentium III computers. The platform is implemented using active network socket programming (ANSP) interfaces [7] which basically are a set of Java APIs for the ease of protocol implementations.

3.1 Content Adaptation for Real-Time Delivery

The goal of the experiments is to examine the performance of real-time adaptation using the pervasive network infrastructure. When a mobile user moves, the access interface (the bottleneck link) of the device to the Internet changes. Indeed, the network infrastructure is the best medium to detect and monitor these changes in user access context. In order to carry out experiments flexibly, a `tc` script on class-based queueing (CBQ) in Linux is used to emulate the effect of varying last mile bandwidth.

Real-Time Delivery of Web Pages In order to facilitate real-time delivery, selected information content of web pages can be pre-fetched and compressed. In the experiments, deliveries of web pages are tested against different user's expected real-time contraint W. The constraint measures from the instant that a user send a request for a web page till the instant that the page is displayed.

Recalling that the size of packet adapted to a bottleneck link, s_A, in Eqn. (4), then a loose upper bound in Eqn. (5) can be used as $\sum_{i=\{EI,IE|data\}} [t(i) + p_c(i)]$ cannot be estimated. But, if these transmission and processing delays can be measured, then a tighter bound should be deployed.

In the testbed, 100 Mbps switched Ethernet connections are used. If a packet is delivered along path i and its size is s_i bytes, then $t(s_i) = \frac{8 \cdot s_i}{100 \times 10^6} = \frac{s_i}{12{,}500{,}000}$. Even though the routers are using store-and-forward and there are m_i routers in a path, the transmission delay of $m_i \cdot t(s_i)$ is not significant as the value of m_i is usually small. On the other hand, the compression operations on content may be a time-consuming process in the experiments. Actually, it has been measured in the testbed, and it is equal to 1 msec per 1340 bytes of original bitmap data. Therefore, we have $p_c(s_i) = \frac{s_i}{1{,}340{,}000}$. Furthermore, the sum of $p_p(RE)$ and $p_p(ER)$ is found to be negligible. Therefore, the real-time constraint in the web page delivery experiments becomes

$$s_A(s_i, m_i) < B \cdot \{W - \gamma_i - m_i \cdot t(s_i) - p_c(s_i)\}. \tag{6}$$

Suppose that there are k slices and the set of selected paths is K, $|K| = k$. The path with the longest delay should have impact on the performance of real-time delivery issue. Hence, we have

$$s_A < B \cdot \left\{ W - \max_{i \in K} \{\gamma_i - m_i \cdot t(s_i) - p_c(s_i)\} \right\}. \tag{7}$$

But if the routers along do not carry out the extra processing function, e.g., compression, then $p_c(s_i) = 0$.

A stock-quote web page is used in the experiments, which contains a number of intra-content components and they are ranked into three classes only.

- The text, made up of HTML markup tags and the body of information, contains information essential to a user including current share price, daily high/low, trading volume, and so on. The text is classified as rank 1 (most important) and is always included in deliveries;
- A stock price graph is classified as rank 2. It can be compressed to 397 different sizes when needed. The relationship between compressed size and compression parameter is encoded as the meta-information of the bitmap file. Then, APNI can choose the closest adapted size, and the nodes operate appropriate compression accordingly. Compressed sizes at lower compression parameters have a larger granularity (i.e. more discrete in compressed sizes among higher compression levels). After the maximum compression, the graph has a size of 13590 bytes (37.79% of the original). If the s_A left after rank 1 component is less than the maximally-compressed size, the stock price graph is dropped.
- Graphic buttons and banners are classified as rank 3.

By calculating the desired adapted size s_A with Eqn. (7), a selection of in-page components is obtained. Ideally, the resultant size should vary linearly with s_A. However, this is not possible since the adaptation granularity is non-continuous. Conversely, the resultant adapted size should stay below the ideal curve such that the real-time delivery can be achieved with the maximum amount (i.e. perceptual value) of information delivered to the user. Please note that when s_A is below 6600, both the ideal and real curves stay flat. This is because rank 1 components are always transmitted to preserve the minimum amount of perceptual value for every page.

A stock-quote web page is used in the experiments, which contains a number of intra-content components and they are ranked into three classes, $R_1 > R_2 > R_3$. The sizes of packets are adapted according to the Eqn. (4) as both the $t(\cdot)$ and $p_c(\cdot)$ are measured and calculated.

Fig. 3(a) shows the original web page. Fig. 3(b) shows a resulting page that the R_2 components are compressed for preserving R_3 components. When the bottleneck bandwidth or the real-time constraint is further reduced, the R_3 components are dropped and the R_2 components may further be compressed in Fig. 3(c). In Fig. 3(d), extreme condition occurs and no further image reduction can be possible. Thus, only the R_1 component, i.e., the text file, is delivered.

4 Conclusions

In the paper, we explore the possibility of offering real-time content adaptation on set of data streams using the active pervasive network infrastructure. With different relative importance among data sets, traffic control and discrimination

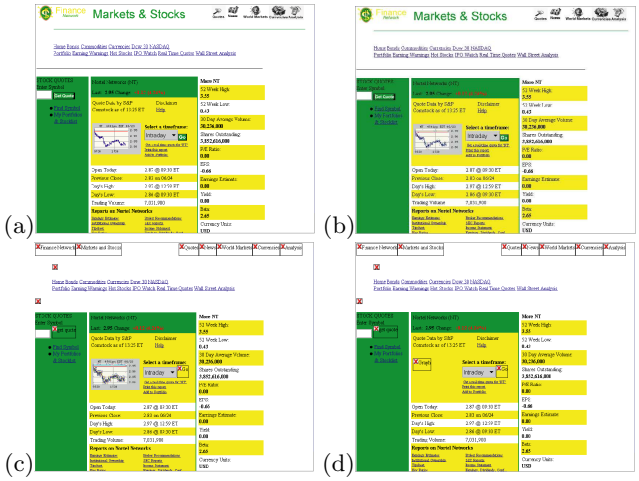

Fig. 3. Real-time delivery (a) original page, (b) moderately compressed (the stock graph), (c) heavily compressed (removals of R_3 components), (d) most components dropped but critical data (the stock price).

with different operations on content adaptations have been examined. Piggyback extension to users' preferences messages is proposed to smoothly enhance the pervasive network infrastructure design. Content adaptation is achieved transparently to both clients and server systems. Real-time delivery services overcome stochastic network situations and abruptly changing bottleneck link bandwidth problem while retaining information integrity and preserving critical data at the best of the limit of an environment.

References

1. Weiser, M.: The computer for the twenty-first century. Scientific American **265:3** (1991) 94–104
2. Satyanarayanan, M.: Pervasive computing: vision and challenges. IEEE Personal Communications **8:4** (2001) 10–17
3. Banavar, G., Bernstein, A.: Software infrastructure and design challenges for ubiquitous computing. Communications of the ACM **45:12** (2002) 92–96
4. Law, K.L.E., So, S.: Pervasive computing on active networks. The Computer Journal **47:4** (2004) 418–431
5. Chan, A.L., Law, K.L.E.: QoS Negotiations and real-time renegotiations for multimedia communications. IEEE Inter. Conf. Computer Communications and Network 2002, Miami, USA (2002) 522–525
6. Tennenhouse, D., Smith, J.M., Sicoskie, W.D., Wetherall, D.J., Minden, G.J.: A Survey of Active Network Research. IEEE Communications Mag. (1997) 80–86
7. Law, K.L.E., Leung, R.: A Design and implementation of active network socket programming. Microprocessors and Microsystems Journal, Elsevier Publisher **27** (2003) 277–284
8. Postel, J. (ed.): Transmission Control Protocol. IETF, RFC 793, Sept. 1981.