On Supporting Weakly-Connected Browsing in a Mobile Web Environment[*]

Hong Va Leong[†] Dennis McLeod[‡] Antonio Si[⋆] Stanley M.T. Yau[†]
[†]Department of Computing, Hong Kong Polytechnic University, Hong Kong
[‡]Computer Science Department, University of Southern California, Los Angeles, CA 90089
[⋆]Oracle Corporation, 500 Oracle Parkway, Redwood Shores, CA 94065

Abstract

A mobile environment is weakly-connected, characterized by low communication bandwidth and poor connectivity. Conventional paradigm for surfing mobile web documents is ineffective since portions of a document could be corrupted during transmission and it is expensive to retransmit the whole document. It is important that the high content-bearing portions should be transmitted successfully so that a mobile client could at least obtain a high level content and determine if the corrupted portions need to be retransmitted. We have proposed a multi-resolution transmission *paradigm which allows higher content-bearing portions of a web document to be transmitted, by partitioning it into multiple organizational units and associating an* information content *with each unit. The client can explore the higher content-bearing portion earlier and terminate browsing an irrelevant document sooner. In this paper, we extend our previous work and propose a* fault-tolerant multi-resolution transmission *scheme which allows units of higher information content to be recovered from transmission error. The client can obtain an overall content of a web document and either terminate the transmission of the remaining portions or decide if the corrupted portions need to be retransmitted. We demonstrate its feasibility with a prototype and with simulation results.*

1 Introduction

We focus on a mobile environment in which mobile clients navigate web documents via common browsers, termed a *mobile web environment*. A mobile environment is *weakly-connected*, characterized by its low communication bandwidth and poor connectivity. Traffic generated due to web accesses in a mobile setting should consume as little bandwidth as possible. In these aspects, conventional approaches to web navigation suffer from serious limitations.

Conventional approaches to web navigation usually involve searching of web documents via some search engines, followed by human exploration of each document for relevance. By a document, it is not only referred to as simply a single web page, but it may also include a collection of hierarchically linked related pages, composing a larger document. Very often, most documents identified by a search engine are irrelevant to a user, thus wasting the precious bandwidth and the limited energy of a mobile client by transferring them. Furthermore, there is no guarantee that a client will receive a document successfully. This problem is getting more serious when the size of a web document is getting bigger as we witness a proliferation of technical documents published to the web.

In [12], we propose a *multi-resolution transmission* paradigm which allows higher content-bearing portions of a web document to be transmitted to a mobile client earlier. The approach is built upon the XML markup language which defines a structure for web documents, though it is extensible to cater for HTML documents. A document is partitioned into multiple organizational units at various *levels of detail* (*LOD*) according to its XML structure. A notion of *information content* is associated with each organizational unit, indicating the amount of information captured by the unit. Units with higher information content will be transmitted earlier during a web browsing session, as different organizational units of a document contribute to different amount of information to a user. A document can be transmitted and browsed at a coarser resolution, with the details to be filled in progressively. A mobile client is able to explore the higher content-bearing portions of a web document earlier and to determine if the document is of any interest. This could reduce scarce wireless bandwidth consumption by early terminating the transmission of irrelevant documents. We explore in this paper several alternative notions of information content.

One limitation of the multi-resolution transmission paradigm is its lack of resilience to faulty transmission. An organizational unit could get corrupted while being transmitted via a faulty wireless channel. A mobile client is not able to determine if the corrupted units are high content-bearing and thus the whole document has to be retransmitted. Since retransmitting the whole document is expensive, high content-bearing units should be transmitted with a higher success probability so that a mobile client could at least obtain a high level content of the document and determine if the remaining units need to be transmitted. We extend our approach with a fault-tolerant transmission capability so that a mobile client could recover the corrupted units sent over the unreliable network, known as *fault-tolerant multi-resolution transmission*. The client is able to obtain an overall content of a web document and either terminate the transmission of the

[*]This research is supported in part by the Hong Kong Polytechnic University Research Grant number G-V633.

remaining portions or decide if the corrupted portions need to be retransmitted. We demonstrate its feasibility with a prototype implementation and conduct some preliminary evaluation via simulation.

This paper is organized as follows. We give a brief survey on related work on web browsing and mobile computing in Section 2. In Section 3, we briefly review the notion of information content and describe its application in multi-resolution browsing. We also propose the adjustment of the information content of an organizational unit in response to a search query. The fault-tolerance mechanism in delivering more important units is delineated in Section 4. We conduct some simulated experiments on the performance of the fault-tolerance mechanism in Section 5. This paper is finally concluded with a brief discussion on our future research directions.

2 Related Work

The explosion of information available on the Internet and the user-friendliness of web browsers have dramatically changed the way information is accessed. Web surfers will simultaneously experience the excitement of boundless information and the frustration of trying to find what they actually want. It has been their common experience that they are not worried about too little information, but about too much useless information.

There have been numerous works attempting to increase the accuracy of information searching on the web [9, 15, 17], trying to realize the WYGIWYW (**W**hat **Y**ou **G**et **I**s **W**hat **Y**ou **W**ant) paradigm. A common technique is to build an index over a collection of documents found by a web search process such as Lycos [15] or WebCrawler [17], which typically searches exhaustively. Users can issue queries directly to the pre-computed index. It is, however, quite uncommon for a web index to provide relevance feedback to improve on the accuracy for future search.

A probably better approach is to establish a user profile, capturing individual users' interests. The profile is used to filter out irrelevant information identified by a search engine [1, 4, 9]. Usually, conventional searching techniques are employed to identify relevant information in response to a query. Additional filtering techniques are then employed to discard irrelevant information according to the personalized profile of a user. Mechanisms for updating a profile are also provided for the profile to adapt to changes in user interest. Performance of this kind of systems depends on how well the profile could capture and adapt to user needs. In this aspect, relevance feedback plays an important role in modifying the profile appropriately to reflect the changes in user interests.

Rather than providing a user with a set of selected documents, recommender systems [3, 19] assist a user in his/her browsing behavior, interactively offering advice about which subsequent hyperlink(s) would likely contain the most relevant information. The system refines its knowledge on user interests by keeping track of whether its advice is followed.

Recent advances in wireless communication and portable computers have enabled users to access web information along the road [16]. Since wireless channels have limited bandwidth and mobile clients are constrained by limited battery life, one must consider efficient use of bandwidth and power carefully, striking a balance for the best solution for the application at hand [2, 11]. To reduce energy consumption, clock rate reduction and disk spin-down techniques have been proposed [7, 20]. To reduce bandwidth utilization, techniques for caching of data items from the server in a client's local storage have been investigated [6, 13].

Constrained by the low bandwidth wireless channels in a mobile web, techniques for improving web access performance by reducing bandwidth consumption is needed. Some researchers focus on improving web browsing performance via a caching and prefetching mechanism [10]. These approaches take advantage of the idle time of a client system, prefetching frequently accessed documents in advance and storing them in client's local storage. This reduces the latency of web browsing. Prefetching, however, demands higher bandwidth requirement and is thus not as feasible in a mobile environment with an already limited bandwidth.

Other researchers have worked on generating summarized information of a web document and presenting the summary before retrieving the whole document as a kind of filtering mechanism [14]. Lead-in sentences are often recognized as a good summary of a paragraph [5]. This allows a user to determine the relevance of a document earlier and terminate the transmission of an irrelevant document before the actual document is transmitted. However, the whole document is often not a refinement of the summary, thus consuming additional bandwidth when a relevant document is later retrieved. Techniques of protocol reduction and differencing have also been employed to reduce the precious bandwidth consumption [8].

3 Multi-Resolution Transmission

In this section, we briefly review the multi-resolution transmission paradigm and illustrate how it could be adjusted in response to a searching query. scheme as will be described in next section. The structural organization of a document could be modeled by a tree-like indexing structure, called a *structural characteristic* (*SC*). A notion of *information content* is defined as an indicator for the amount of information captured within an organizational unit [12], allowing a web document to be browsed at different LODs.

We defined several LODs: document, section, subsection, subsubsection, and paragraph, providing different degrees of detail with which a user can navigate a document. Our definition of LOD is an abstraction to the actual formatting tags. It has a straightforward implementation in the context of XML, which allows the explicit definition of document structures. For instance, in an XML document, a section LOD might be implemented using a pair of <section> and </section> tags, where section is defined as an element in an XML DTD (Document Type Definition) for document type research-paper. We are working on a mapping between HTML and XML documents which allows our approach to work on HTML documents as well.

3.1 Information Content

The set of keywords in a document will be used to determine the information content of an organizational unit. A weight is associated with each keyword which indicates its relative importance in a document. We use a logarithmic function of keyword occurrences to define this weight. For notational convenience, we denote the number of occurrences of a keyword, a, in a document, D, by $|a_D|$ and the number of occurrences of a in an organizational unit, n_i, by $|a_{n_i}|$. The occurrence vector, V_D, of the set of keywords in D, $A_D = \{a | a \text{ is a keyword in } D\}$, can be represented as $V_D = \{|a_D| \mid a \in A_D\} = \{v_1, v_2, ..., v_{|A_D|}\}$. The weight of each keyword, a, for D is denoted as ω_a. It is defined as $\omega_a = 1 - \log_2(|a_D|/||V_D||)$ where $||V_D||$ is the norm of the occurrence vector V_D. We choose the infinity norm $||V_D||_\infty = max(v_i)$. This allows the weight of each keyword to be determined without human intervention which would be infeasible in a mobile environment otherwise. The information content, p_i, of an organizational unit, n_i, is now defined to be the weighted sum of the keywords in the unit, normalized with respect to that of the document, D:

$$p_i = \frac{\sum_{\forall \text{keyword } a \in n_i} |a_{n_i}|\omega_a}{\sum_{\forall \text{keyword } d \in D} |d_D|\omega_d}.$$

Under this definition, the *additive rule* for information contents of sub-units will hold and that the total information content for the document D adds up to unity. In other words, for an organizational unit, n_j, with m sub-units, $n_{j,1}, n_{j,2}, ..., n_{j,m}$, $p_j = \sum_{k=1}^{m} p_{j,k}$.

3.2 Query-Based Information Content

The notion of information content is based on a static analysis of a document. In practice, the set of documents that will be transmitted to and browsed by a user is the result of a searching process via some search engines. The degree of relevance of a document, and thus its organizational units, to a user is usually affected by its relevance to the query initiated. We therefore extend the definition of information content in response to a search query in this paper and name the revised notion, **Q***uery-based* **I***nformation* **C***ontent* (*QIC*). Notice that while information content of an organizational unit is static, its corresponding QIC is dynamic, changing according to the definition of an initiated keyword-based query.

We denote a query, Q, by a vector, analogous to that of a document. The query Q contains a set of keywords, which we call the *querying words*, $A_Q = \{a | a \text{ is a keyword in } Q\}$. Normally, all words in Q which are not stop words [12] should be considered as keywords and the number of occurrences of a in Q is $|a_Q| = 1$. This forms an occurrence vector, V_Q for Q. Sometimes, a user might want to emphasize a particular keyword by repeating it in order to give it a higher weight during a search process so as to bias the searching procedure towards certain words. We take the weight of each querying word into account, so as to be symmetrical to the processing of the document. The weight of a querying word a is denoted as $\omega_a^Q = 1 - \log_2(|a_Q|/||V_Q||)$ if $|a_Q| \neq 0$ and it is zero otherwise. The QIC, q_i^Q, of an organizational unit, n_i, in D with respect to Q is now defined to be the combined weighted sum of the keywords in the unit, normalized with respect to D and Q:

$$\begin{aligned} q_i^Q &= \frac{\sum_{\forall \text{keyword } a \in n_i} |a_{n_i}|\omega_a \cdot \omega_a^Q}{\sum_{\forall \text{keyword } d \in D} |d_D|\omega_d \cdot \omega_d^Q} \\ &= \frac{\sum_{\forall \text{keyword } a \in n_i \cap Q} |a_{n_i}|\omega_a \omega_a^Q}{\sum_{\forall \text{keyword } d \in D \cap Q} |d_D|\omega_d \omega_d^Q}. \end{aligned}$$

Notice that the *additive rule* also holds for QIC.

One might argue that QIC of some organizational units may become zero, due to the absence of a querying word. We could choose to trade slight computational efficiency for a more general definition of QIC by replacing the product between the weights from document keyword and querying word with their sum. To ensure that individual weights are in comparable scale, we associate a scaling factor, ρ, with ω_a^Q. This **M***odified* **Q***uery-based* **I***nformation* **C***ontent* (*MQIC*), $\tilde{q}_i^Q$, is defined as:

$$\tilde{q}_i^Q = \frac{\sum_{\forall \text{keyword } a \in n_i} |a_{n_i}|(\omega_a + \rho\omega_a^Q)}{\sum_{\forall \text{keyword } d \in D} |d_D|(\omega_d + \rho\omega_d^Q)},$$

where $\rho = \sum_{\forall \text{keyword } a \in D} |a_D| / \sum_{\forall \text{keyword } a \in Q} |a_Q|$. Notice that the *additive rule* still holds for MQIC.

3.3 Structural Characteristic Generation

To generate the SC for a document, the document is pre-processed and a keyword-based logical index is established for each organizational unit. The SC is created by deriving the information content of each organizational unit from the logical index. Traditional information retrieval mechanisms could be employed for pre-processing. In brief, it can be structured as five modules: *document recognizer*, *lemmatizer*, *word filter*, *keyword extractor*, and *structural characteristic generator*, operating in a pipelined fashion.

The document recognizer converts an XML document into a plain text document, taking consideration of formatting information including the hierarchical document structure and those specially formatted words. The lemmatizer converts document words into their lemmatized form. The word filter eliminates non-meaning-bearing words, usually referred to as "stop" words. The keyword extractor performs a frequency analysis on the potential keywords. In addition, certain specially formatted words, such as boldfaced and italized, also qualify as keywords. The structural characteristic generator computes the information content of each organizational unit and generates the SC.

Notice that the QIC of each organizational unit is determined every time the search engine receives a searching query from a user. Since the weights of keywords, ω_a, of a document, D, remain unchanged across queries, only the contribution by querying words need to incorporated. As the number of querying words is small, the number of terms involved in the computation is small and thus, the computational overhead of QIC is quite low. This vector space model has been shown to be competitive with alternative methods [4].

Sect./Subsect./Para.	IC p	QIC q^Q	MQIC $\tilde{q}^Q$
0	0.04891	0.15119	0.13635
0.0	0.04891	0.15119	0.13635
0.0.0	0.04891	0.15119	0.13635
1	0.11773	0.33231	0.30771
1.0	0.11773	0.33231	0.30771
1.0.1	0.03924	0.18087	0.16784
1.0.2	0.04899	0.12543	0.11461
1.0.3	0.01327	0.00000	0.00158
1.0.4	0.01623	0.02601	0.02369
2	0.11619	0.09738	0.10083
2.0	0.11619	0.09738	0.10083
2.0.1	0.05327	0.03402	0.03897
2.0.2	0.01404	0.13131	0.01256
2.0.3	0.04888	0.05023	0.04932
3	0.60311	0.29368	0.33276
3.0	0.15982	0.11589	0.11690
3.0.1	0.04710	0.07853	0.07087
3.0.2	0.06370	0.02422	0.02933
3.0.3	0.04902	0.01313	0.01670
3.1	0.11941	0.17778	0.17737
...	...	...	...
3.2	0.19599	0.00000	0.02329
3.2.1	0.03931	0.00000	0.00467
...	...	...	...
3.3	0.06293	0.00000	0.00748
...	...	...	...
3.4	0.06496	0.00000	0.00772
...	...	...	...
4	0.11196	0.12543	0.12209
4.0	0.11196	0.12543	0.12209
...	...	...	...

Table 1: Information content of a draft paper

We demonstrate the SC generation using an early draft of this manuscript. The information contents of organizational units and their respective QIC using a query $Q = \{\text{browsing, mobile, web}\}$ are illustrated in Table 1. The abstract is considered as Section 0. Paragraphs not belonging to any subsection are grouped under a virtual subsection.

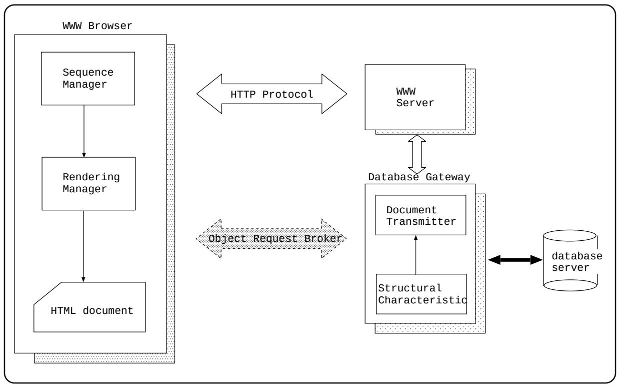

Figure 1: Prototype architecture

We have implemented a Java prototype for multi-resolution transmission, based on the CORBA infrastructure [12]. The architecture is depicted in Figure 1. The client renders each organizational unit incrementally at the proper position in the browsing window when the unit is received.

4 Fault-Tolerant Transmission

The Internet is quite unstable in terms of connectivity. Occasional disconnection during transmission of web information is common and the browser will get "stalled". This situation will get worse in the context of a mobile environment. Data received will get corrupted very easily. We would like to "enhance" the reliability of delivering organizational units by introducing redundancy so that more important organizational units of a web document can be received successfully with a much higher probability, providing a user with enough information to determine if the corrupted units need to be retransmitted. Our approach is based on cryptographic encoding of information [18], so that redundancy is transmitted at the end of clear text to enable termination of redundancy transmission when enough information is received.

4.1 Fault-Tolerating Encoding

We assume that a document can be divided into M pieces, each of which is a fundamental unit of transmission over the wireless network. These pieces are called *data packets*, with a size of s_p bytes each. Data packets are received either *intact* (without error) or *corrupted* (with detectable error). A missing packet can be detected when the next packet is received, since the wireless channel is FIFO but unreliable. Simple sequence number as used in the datalink layer transmission protocol suffices in this context. We propose to adopt the cyclic redundancy code (CRC) for the detection of packet corruption, since it has a low computational cost and a high error coverage.

A property of the technique in [18] is that a file can be divided into M "raw" packets. Via a matrix multiplication procedure, these M packets can be transformed into $N \geq M$ "cooked" packets such that if "any" M out of the N cooked packets can be collected, the original file (all the M raw packets) can be reconstructed via another matrix operation based on polynomial code. The transformation adopted in [18] is cryptographic in nature, such that collecting any $M-1$ cooked packets is completely useless in recovering the raw packets. A slight modification is to adopt the Vandermonde polynomial in the transformation stage, followed by making the upper portion of the multiplying Vandermonde matrix into an identity matrix via elementary matrix transformation. This will ensure that the first M cooked packets will appear in exactly the same form as the raw packets (i.e., in clear text), thus saving recovering effort. Furthermore, it allows a portion of the original information to be used once they are available, without the need to wait until M different cooked packets are collected.

Assuming that the probability a packet will be corrupted is α and that the corruption events of individual packets are independent, the number of packets, P, to be collected before the original file can be reconstructed (using M cooked packets) follows a negative Binomial distribution:

$$Pr(P = x) = \frac{(x-1)!}{(M-1)!(x-M)!}\alpha^{x-M}(1-\alpha)^M,$$

with an expected number of required packets, $E(P) = M/(1-\alpha)$. To ensure a successful probability of $S =$

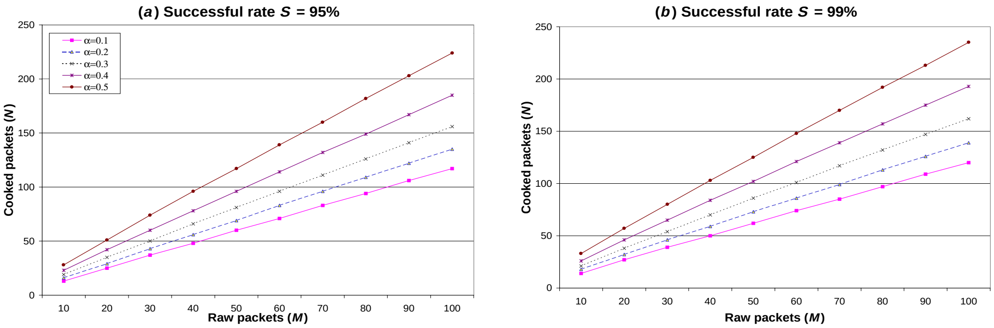

Figure 2: Number of cooked packets needed

95%, one have to transform a file of M raw packets into N cooked packets such that:

$$Pr(P \leq N) = \sum_{i=M}^{N} \frac{(i-1)!}{(M-1)!(i-M)!} \alpha^{i-M} (1-\alpha)^M \geq S.$$

This inequality can be solved for N given M and S, yielding an optimal number of cooked packets.

4.2 Fault-Tolerating Multi-Resolution Transmission

Using the encoding scheme discussed above, a document, D, can be transmitted pretty reliably over a weakly-connected wireless channel in an order defined by QIC. For instance, transmitting a document, D, at the document LOD is equivalent to the conventional paradigm of transmitting D sequentially. Assume that the document has a size of s_D. To transmit at the document LOD, the whole document can be broken into $M = \lceil \frac{s_D}{s_p} \rceil$ raw packets. Depending on an estimated channel failure probability, α, and a desired success probability, S, D is transformed into $N \geq M$ cooked packets. To make a judicial choice of N with respect to M in order to ensure a given success probability S, we illustrate the value of N against M at different α values in Figure 2.

Figures 2a and 2b depict the findings when S = 95% and S = 99% respectively. We notice that the number of cooked packets required is pretty much of a linear relationship with the number of raw packets. This leads us to adopting *redundancy ratio*, $\gamma = \frac{N}{M}$, as a guideline for choosing N. We re-plot the results of Figure 2 with γ against α in Figure 3, using a medium value of $M = 50$. We also illustrate the variation of γ when M varies down to 10 and up to 100 in the figure. We can observe that the range of γ for different values of M does not change too much. For practical reason, we may thus consider γ as a function of α. To balance the amount of redundancy with successful transmission probability, the value of γ could be defined as an adaptive function of the observed summarized value of α, using perhaps a kind of EWMA measure [6].

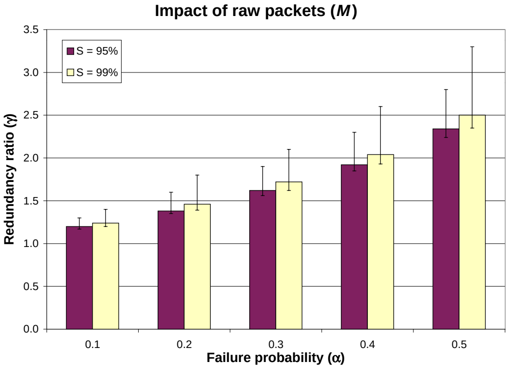

Figure 3: Redundancy ratio γ versus failure α

When transmitting a document at a lower LOD other than the document LOD, the organizational units at the appropriate level are ranked and transmitted according to QIC. Let the organizational units at the required LOD be $\{n_1, n_2, ..., n_m\}$, to be transmitted in the order of $\langle n_{j_1}, n_{j_2}, ..., n_{j_m} \rangle$. To transfer D to a client, the permuted sequence of organizational units, $\langle n_{j_1}, n_{j_2}, ..., n_{j_m} \rangle$, are transformed into N cooked packets. The client will discard corrupted packets, upon receiving the cooked packets. The transmission can be terminated when any one of the following three conditions occurs: the client receives sufficient number of cooked packets to reconstruct the whole document; all cooked packets are received; the user has determined that the document is irrelevant and hit the "stop" button.

If a client is not able to receive enough intact cooked packets to reconstruct the document after all cooked packets are transmitted, the client is suffering from a "stalled" transmission. Usually, a user will have to reload the document from scratch by hitting the "reload" button. Alternatively, the client may detect the stalled situation and request the server for a retransmission of the document. In a mobile context, this may be an overkill, since the document might get stalled only due to missing a few packets. A better alternative is to "cache" the intact cooked packets received and use them to reconstruct the document when a retransmission of corrupted packets occurs. The local storage of the client could be utilized to store the partial document so as to increase the chance of getting the M intact cooked packets required to reconstruct the original document. This *caching* approach is very useful especially when we adopt the Vandermonde approach which allows the cooked packets to be used immediately after they are received. This mechanism can be implemented using client and server side interceptors, where alternative mechanisms such as compression or ARQ are also implemented [8].

5 Evaluation

In order to quickly generate a portrait of an overall behavior and performance of our proposed scheme, we have developed a simulation model for the study. Our simulation study is mainly focused on the impact of transmission errors of a wireless channel on the performance of our fault-tolerance mechanism.

Parameter	Description	Value
s_p	Raw size per packet	256
s_D	Size per document	10240
O	Overhead (CRC+sequence number)	4
M	Number of raw packets	40
N	Number of cooked packets	60
B	Bandwidth (kpbs)	19.2
δ	Skewed factor in information content	3
I	Irrelevant documents	50%
F	Info content to determine relevance	0.5
α	Probability of a corrupted packet	0.1
γ	Redundancy ratio N/M	1.5

Table 2: Parameter settings

The default experimental parameter settings are depicted in Table 2. We assume that each simulated document has a size of 10240 bytes, divided into 40 raw packets, each of size 256 bytes. Raw packets are transformed into cooked packets, each has a size of 260 bytes. The wireless channel has a typical bandwidth of 19.2kbps. Each simulated document is composed of 5 sections; each section is composed of 2 subsections; each subsection is composed of 2 paragraphs. We model the information content of each paragraph by a uniform distribution. We use a skewed factor, δ, to model the ratio between the highest information content of a paragraph and the lowest information content of a paragraph. In our experiments, δ is set to 3. Each simulated browsing session will visit 200 random documents, with a certain percentage of documents, I, defined to be irrelevant. Each irrelevant document will be discovered to be irrelevant by a client after a total information content of F has been received at the client. The default value of I is 50% and F is 0.5.

We also model a stalled situation in our simulation model. A stalled situation is defined as not being able to receive the required M intact cooked packets for the document at the end of the download. After a stalled transmission, a retransmission of the document will be initiated, either with or without utilizing the cache. We refer to the default HTTP approach of not utilizing the cache as the NoCaching approach and the one using the cache, the Caching approach.

The mean *response time* taken to visit a document in a session is measured. The same experiment is repeated 50 times and the average of the 50 mean response times is taken in plotting our curves. In general, we observe quite small standard derivations over the 50 repetitions, giving tight confidence intervals to our results.

5.1 Experiment #1

In our first experiment, we study the performance difference between Caching and NoCaching under various redundancy ratio γ. We compare the case where a percentage, I, of the 200 documents browsed are irrelevant with the case where all documents are relevant, i.e., downloaded to their entirety. All documents are transmitted at the document LOD in this experiment, modeling conventional transmission paradigm. The results of this experiment are depicted in Figure 4.

The first row of Figure 4 depicts response times when all documents are relevant in a browsing session, while the second row indicates the performance when only half of them are relevant, i.e., $I = 0.5$. A document is discovered to be irrelevant when it is halfly loaded, i.e., $F = 0.5$. The first column reflects the scenario where the client does not maintain any cache to capture the good packets across unsuccessful loading sessions and the second column indicates the use of cache. It is clear that the impact of the cache is very significant, especially when the error rate of the channel is high. By contrast, the amount of irrelevant documents is not playing such an important role. We can briefly conclude that the use of cache in a highly unreliable wireless channel is very effective and must probably be implemented. Concerning the accuracy of the experiment results, the standard deviation over the 50 repetitions is only between 1% to 5% of the mean in most trials. In many cases, it is even below 1%. The 95% confidence interval for the response times is thus very small.

We observe, in general, that the redundancy ratio $\gamma = 1.5$ is a good choice to yield a reasonable performance, for a small to moderate error rate, α, or when caching is enabled. Only when caching is disabled and when α is over 0.3 will we require γ to be increased, perhaps up to a value of 2. Therefore, we will adopt a default value of 1.5 for γ in the rest of our experiments.

5.2 Experiment #2

In this experiment, we would like to study the effect brought about by early terminating the transmission of irrelevant documents on the performance of document browsing. It is expected that the larger the value of I, the faster is the response time. However, we would like to quantify how the performance is affected. This experiment is divided into two sets. In the first set, we fix the value of F to 0.5 and vary the percentage of irrelevant documents, I. In the second set, we fix the value of I to 0.5 and vary the value of F. Again, all documents are transmitted at the document LOD. The results are illustrated in Figure 5. The first row depicts the impact of varying I while the second row depicts the impact of varying F.

As the percentage of irrelevant documents, I, increases, response times decrease since more documents can be discarded when an information content to the amount F has been received. With Caching, the performance is acceptable even with a high error rate, while it is poor under a high error rate with NoCaching. The curve is quite linear in nature, due to the fact that the response time is a weighted average of the relevant documents and irrelevant documents, thus, should be

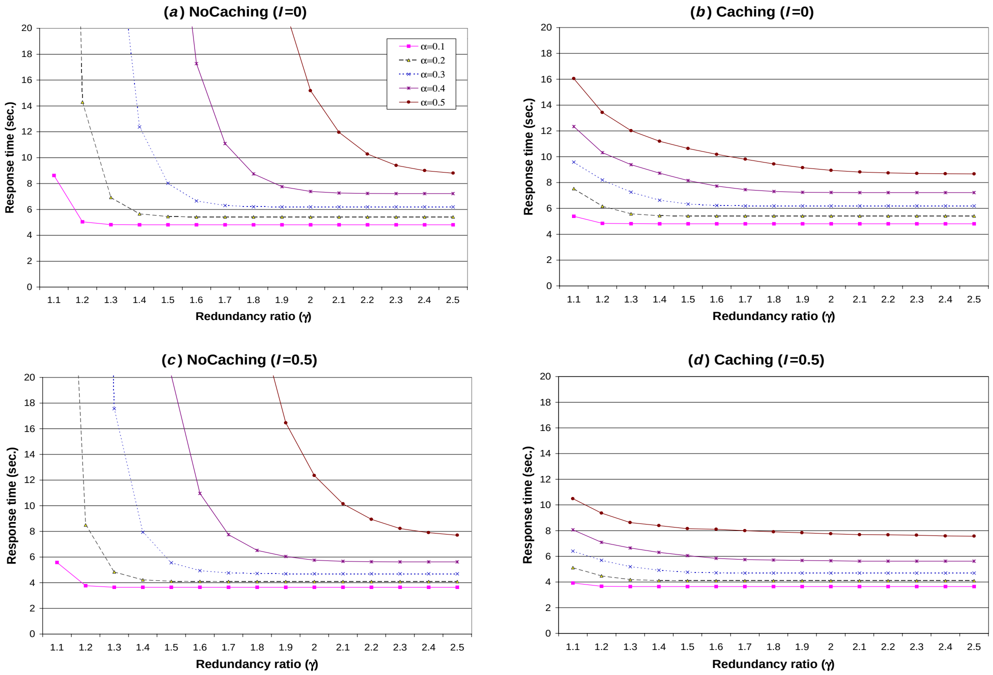

Figure 4: Performance of Caching *versus* NoCaching by varying redundancy ratio γ

linear with respect to I.

When we fix I and vary F, we no longer observe a linear relationship between response times and the required information content. The point for $F = 0$ is artificial, since a document is not downloaded at all. The increase in response times is slow at the beginning. But when more information content is required, the increase becomes more rapid and it finally slows down to almost flattened. This is due to the fact that it is easy to get a low information content even in the presence of failures. Therefore, only a few initial uncorrupted cooked packets in clear text are sufficient to determine the relevance of a document, yielding a low response time. As F increases, the clear text packets are no longer sufficient to provide the required information content; additional packets must be received. In order for the reconstruction to be carried out, at least M intact packets must be received. This leads to a substantial jump in the number of packets required and in turn, response times as well. This jump has been averaged out in our numerous document accesses and repetition of experiments, but the trend of a faster raise is still observable. Finally, when all M packets are needed, the received information content will be 1, sufficient to serve all purpose, leading to a flattened curve towards the end.

5.3 Experiment #3

Our third experiment studies the benefit brought about by multi-resolution browsing in discarding irrelevant documents early. To remove the effect of loading relevant documents (which are loaded to their entirety) for a fair comparison, we assume that all documents are irrelevant and that they can be discarded once an information content of amount F has been received. We test with various LODs at several channel failure rates. We experiment with document, section, subsection, and paragraph LODs since our simulated documents do not have subsubsection defined. To indicate the relative merits of each of the different LODs in the multi-resolution browsing approach, we measure the *improvement* in response times over the conventional approach, namely, browsing at the document LOD. The improvement at a particular LOD is the ratio between the response times at the document LOD and at that LOD. The results using $\alpha = 0.1$, 0.3, and 0.5 are illustrated in Figure 6. We illustrate the results for Caching only, since it is the better way to transmit documents in a mobile environment. The results from NoCaching exhibit a very similar behavior.

From Figure 6, we observe that an LOD at the paragraph level leads to a better performance due to the earlier receipt of the most amount of information content, while the performance of LOD at the document level is not as good. With a typical value of F around 0.1 to 0.3, the improvement for the paragraph LOD is quite significant, in that the transmission at the document LOD is about 30% to 50% slower. The section and subsection LODs can also bring an improvement of about 10 to 30%. We can also observe that the improvement is not as sensitive to the failure probability, since it causes a similar adverse effect to transmission at all levels of detail.

5.4 Experiment #4

Our fourth experiment aims at quantifying the impact of the skewed factor, δ, on our multi-resolution browsing approach. We repeat Experiment #3 by fixing α at 0.1 and varying δ from 2, 3, 4, and 5. The results are depicted in Figure 7.

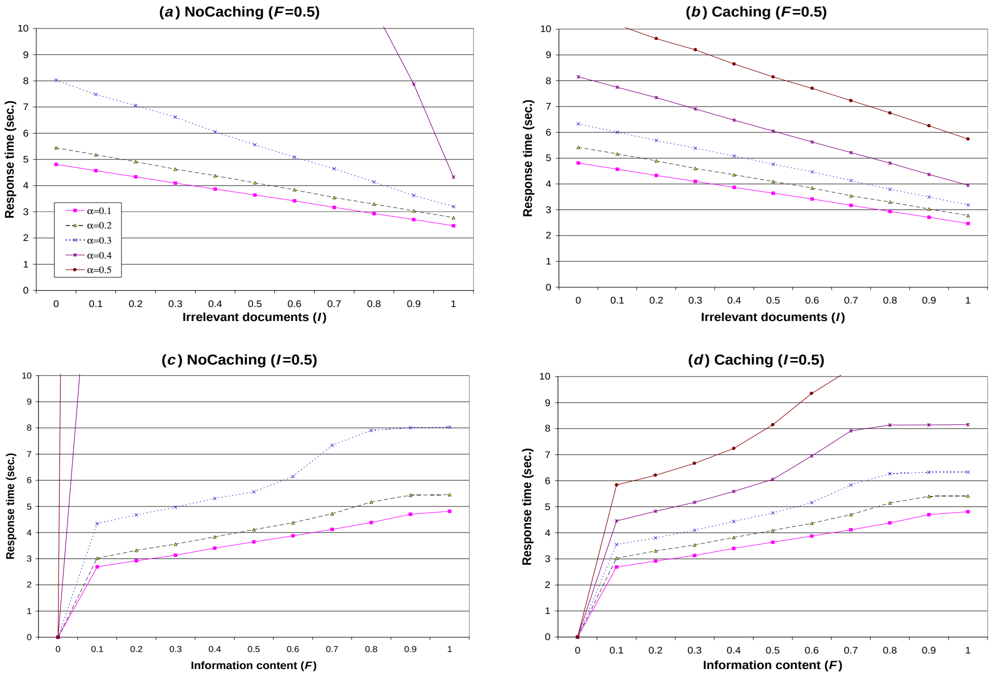

Figure 5: Impact of varying I and varying F

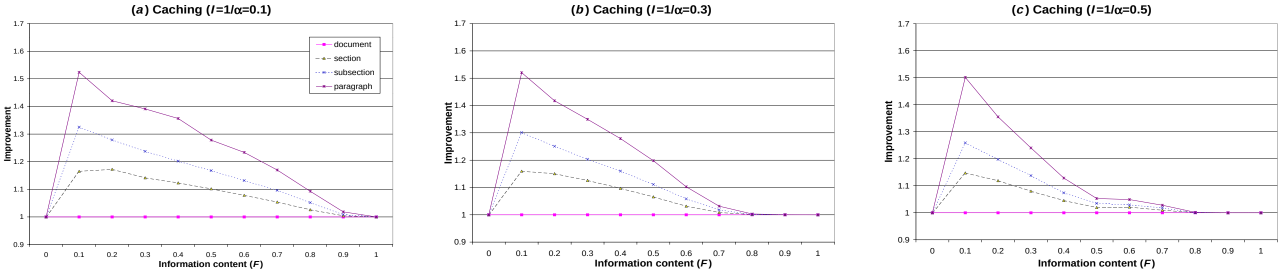

Figure 6: Performance of multi-resolution transmission based on LOD

We observe that the higher the skewed factor δ, the more improvement the multi-resolution transmission approach can bring. The peak of improvement occurs when $F = 0.1$ or 0.2. The reason being that at a higher δ, information contents of organizational units become more non-uniform. Since the transmission of a document is based on the amount of information content of its organizational units, the more content-bearing units will get to a client much earlier. By contrast, with a lower skewed factor, information content of each unit will be relatively similar and the resultant transmission order will be close to conventional sequential transmission.

6 Discussion and Future Work

We have presented a mobile web system for transmitting and browsing web documents over a faulty wireless channel. Based on the notion of information content and its variants, it presents users with the main document content before presenting supplementary information. A redundant transmission scheme is also provided to increase the recoverability of a corrupted document due to unreliable wireless channels.

Currently, our prototype assumes a well-defined organizational structure on a web document defined by XML. However, in a web environment, there exists a large number of unstructured documents. We are working on algorithms to extract the structure of an HTML document from its content. Alternative ways of defining the information content of a document would be explored. With respect to a collection of related pages in the form of a cluster, we are also investigating intelligent prefetching based on information content and user-profiling, utilizing the unused wireless bandwidth being left idle. We are also conducting experiments to measure the throughput of our system in browsing web documents when compared with traditional web browsing paradigm. We would like to obtain more user experiences in browsing web documents using our system and perhaps consider the concept of “intuition level” of each organizational unit in addition to its information content in defining the transmission order.

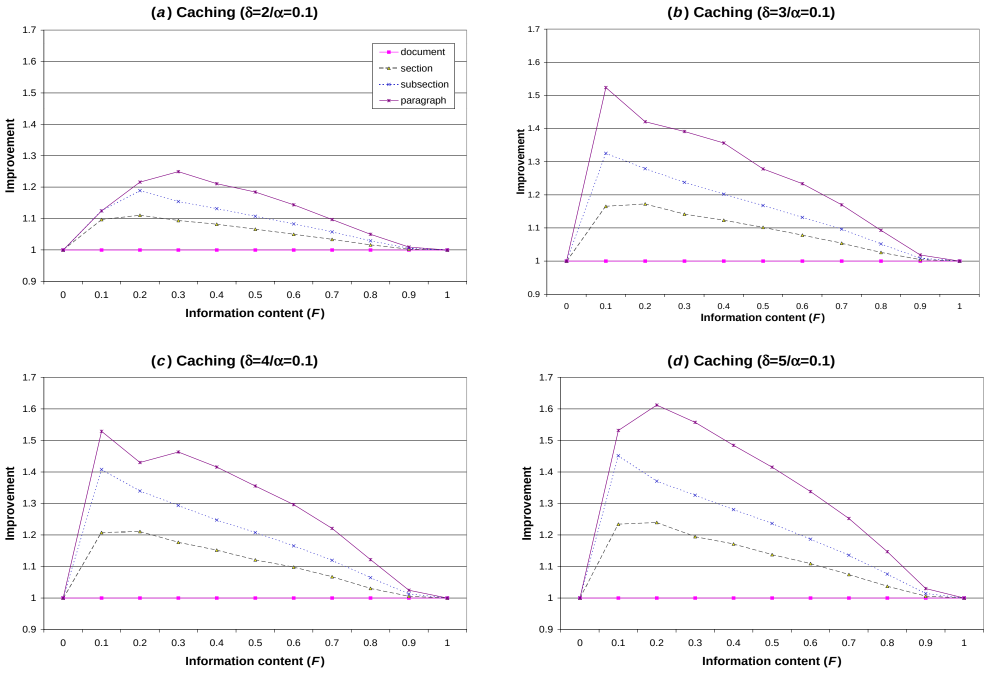

Figure 7: Impact of skewed factor on LOD-based transmission performance

References

[1] M. Ackerman, D. Billsus, S. Gaffney, S. Hettich, G. Khoo, D.J. Kim, R. Klefstad, C. Lowe, A. Ludeman, J. Muramatsu, K. Omori, M. Pazzani, D. Semler, B. Starr, and P. Yap. Learning Probabilistic User Profiles. *AI Magazine*, 18(2):47–56, 1997.

[2] R. Alonso and H. Korth. Database System Issues in Nomadic Computing. In *Proceedings of the ACM SIGMOD Conference*, pages 388–392, 1993.

[3] R. Armstrong, D. Freitag, T. Joachims, and T. Mitchell. WebWatcher: A Learning Apprentice for the World Wide Web, 1995.

[4] M. Balabanoviċ. An Adaptive Web Page Recommendation Service. Technical Report SIDL-WP-1996-0041, Stanford University, 1996.

[5] R. Brandow, K. Mitze, and L.F. Rau. Automatic Condensation of Electronic Publications by Sentence Selection. *Information Processing and Management*, 31(5):675–685, 1995.

[6] B.Y.L. Chan, A. Si, and H.V. Leong. Cache Management for Mobile Databases: Design and Evaluation. In *Proceedings of IEEE International Conference on Data Engineering*, pages 54–63, 1998.

[7] F. Douglis, P. Krishnan, and B. Bershad. Adaptive Disk Spin-down Policies for Mobile Computers. *Usenix Computing Systems*, 8(4):381–413, 1995.

[8] R. Floyd and B. Housel. Mobile Web Access Using eNetwork Web Express. *IEEE Personal Communications*, 5(5):47–52, 1998.

[9] E.J. Glover, W.P. Birmingham, and M.D. Gordon. Improving Web Search Using Utility Theory. In *Proceedings of the CIKM'98 Workshop on Web Information and Data Management*, pages 5–8, 1998.

[10] E. Horivitz. Continual Computation Policies for Utility-Directed Prefetching. In *Proceedings of the 7th International Conference on Information and Knowledge Management*, pages 175–184, 1998.

[11] T. Imielinski and B. Badrinath. Mobile Wireless Computing: Challenges in Data Management. *Communications of the ACM*, 37(10):18–28, 1994.

[12] H.V. Leong, D. McLeod, A. Si, and S.M.T. Yau. Multi-Resolution Transmission and Browsing in Mobile Web. In *Proceedings of the CIKM'98 Workshop on Web Information and Data Management*, pages 13–16, 1998.

[13] H.V. Leong and A. Si. Database Caching over the Air-Storage. *The Computer Journal*, 40(7):401–415, 1997.

[14] I. Mani and E. Bloedorn. Maching Learning of Generic and User-Focused Summarization. In *Proceedings of AAAI*, pages 821–826, 1998.

[15] M.L. Mauldin and J.R. Leavitt. Web-Agent Related Research at the CMT. In *Proceedings of ACM International Conference on Networked Information Discovery and Retrieval*, 1994.

[16] T. Parker. Mobile Wireless Internet Technology Faces Hurdles. *IEEE Computer*, pages 12–14, 1998.

[17] B. Pinkerton. Finding What People Want: Experiences with the WebCrawer. In *Proceedings of Second International WWW Conference: Mosaic and the Web*, 1994.

[18] M.O. Rabin. Efficient Dispersal of Information for Security, Load Balancing, and Fault Tolerance. *Journal of the ACM*, 36(2):335–348, 1989.

[19] E. Spertus and L.A. Stein. A Hyperlink-Based Recommender System Written in Squeal. In *Proceedings of the CIKM'98 Workshop on Web Information and Data Management*, pages 1–4, 1998.

[20] M. Weiser. Some Computer Science Issues in Ubiquitous Computing. *Communications of the ACM*, 36(7):74–84, 1993.