PERSONALIZED PREFETCHING FOR MANETS

S H Srinivasan

Applied Research Group
Satyam Computer Services Ltd
Bangalore, India
SH_Srinivasan@satyam.com

ABSTRACT

Mobile ad hoc networks (MANETs) are a very promising next generation networks. *Prediction* is an essential aspect in the deployment of MANETs. Current work mainly focuses on mobility prediction: prediction of spatio-temporal trajectories of nodes. Web page prefetching based on prediction of user's behavior has been shown to decrease latency in wired networks. Current models of prefetching focus on *what* to prefetch and not *when*. Since routing information changes with time in MANETs, the expected time of user's access is also important. In this paper we integrate prefetching algorithms with mobility models and personalized to user browsing patterns. This not only reduces latency but also ensures just-in-time prefetching.

1. INTRODUCTION

The current generation cellular networks have proved to be extremely beneficial in solving the last mile problem and providing universal connectivity. The cellular networks can be improved in several ways. 3G cellular networks are based on packet switching compared to the current circuit switched architecture. In cellular networks, only the last hops are wireless links. Multihop wireless networks are based on "all wireless" links. A special case of multihop networks are ad hoc networks [1] in which there is no a priori network structure. Nodes can join and leave the network at any time. Such networks are useful in scenarios where very little prior information exists such as parties and disaster situations. In mobile ad hoc networks (MANETs), the nodes are mobile. Since power and bandwidth are scare resources, efficient resource utilization schemes are needed. Resource utilization schemes usually have a prediction component. The position of a node is predicted based on its past behavior and domain constraints like terrain topography.

Web browsing is a very important application for MANETs. Several techniques have been used to enhance user's browsing experience. URL prefetching predicts possible user accesses and downloads web pages that the user is likely to access *before* the user actually accesses them. If the user does not access the prefetched pages, then the effort used in obtaining them is wasted. With accurate prediction algorithms, this wasted effort can be minimized. On the other hand, prediction decreases client latency.

When the nodes are mobile, the routing information for prefetched pages also changes dynamically compared to the current page. Hence the prefetched pages may need to be routed differently. Since it is sufficient if the web pages are delivered just in time, a model of user's browsing behavior is also needed. In this paper, we propose an algorithm that performs personalized spatio-temporal prefetching by integrating user models (personalization), mobility models (spatio-temporal), and prefetching.

This paper is organized as follows. Section 2 briefly describes prefetching schemes. Section 3 outlines different approaches used for tracking mobile nodes. Section 4 contains relevant information on user models. Section 5 lists routing schemes for MANETs. Section 6 describes our proposal for just-in-time prefetching. The paper closes with conclusions.

2. URL PREFETCHING

A hypertext document contains several embedded links. A user browsing a document is likely to access one of the embedded links. If the document is fetched after the click, then the user needs to wait for the document to be downloaded. The download times can be long in MANETs. An alternative is to download *all* the embedded links when the user is reading the current document. Though this reduces user wait time, this also leads to excessive waste of resources since the user is not likely to visit all the links in a document. Hence the alternative is to prefetch the *most likely* document. For this additional information is needed at the client side. Usually the server provides this additional information. (The server collects the usage statistics based on actual user accesses.) Since prefetching now is probabilistic, there is a risk of prefetching an incorrect page.

Prefetching has been shown to reduce latency. See [2]

0-7803-9332-5/05/$20.00 ©2005 IEEE

for a review of several prefetching schemes and their performance.

3. MOBILITY MODELS

Wireless access facilitates user movement. The network responds to user's mobility by changing the routing information. Handoffs in cellular networks can be facilitated by modeling user mobility [3]. Mobility modeling has been used [4] to cluster clients. The clustering has been used in deciding service replication for video streaming.

Mobility models are usually based on Kalman filters or its variations [3]. Recently, sampling-based trackers are increasingly used [4]. The sampling based techniques are more general and have lower complexity. Sampling based techniques exist for tracking multiple objects also [5].

In cellular networks, the base stations are fixed and hence provide a reference coordinate system for mobility tracking. In MANETs, all the nodes are mobile and hence GPS-based or GPS-free positioning is done. [6] uses an extended Kalman filter based tracking for MANETs.

4. USER MODELS

There have been several models for user's browsing behavior. [7] mentions that a user can be classified as "surfer" or "conservative". Surfers hop from documents to documents. Conservatives on the other hand have a tendency to revisit a given collection of documents again and again. The same reference also cites several models for user behavior.

The user models needed for personalized prefetching require more fine grained information. Since pages are displayed only after the user requests them (by clicking the corresponding hyperlink), prefetching a page just in time requires an estimate of when the user will request the page. This adds an additional dimension to the classical prefetch. Classical prefetch is interested in *whether* the user will click the link. In personalized prefetching, *when* a user clicks is also important. Hence an additional level of modeling is needed.

5. MANET ROUTING

Routing in MANETs can be classified into two categories [8, 9]: table driven (proactive) and source-initiated (reactive). The proactive protocols update the routing tables whenever there is a topology change. For reactive protocols, routing table update is triggered in response to packet forwarding. The routing table entries are always valid for proactive protocols but the routing table update overhead is high. The up-to-date information may not be used if there is no traffic toward a particular node. Reactive protocols have higher routing latency but there are no unnecessary routing table updates.

Table driven protocols are: destination-sequenced distance vector routing, cluster-head gateway switch routing, and wireless routing protocol.

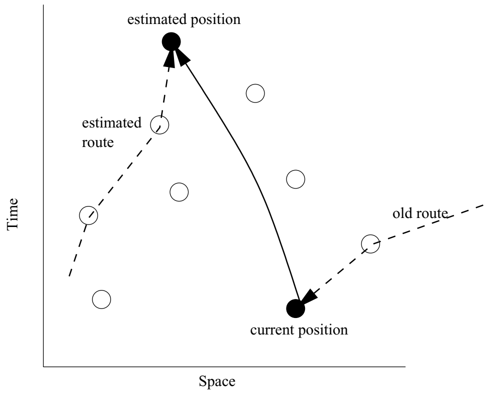

Fig. 1. Schematic diagram of spatio-temporal prefetching. The node under consideration is shown as dark circle. Neighboring nodes are shown as open circles. The trajectory of the node under consideration is shown as unbroken line. The routing is shown as broken lines. The trajectory is estimated for time T – the estimated time of user's next click. URL P is requested to be delivered at the estimated position using an estimated route.

6. PERSONALIZED SPATIO-TEMPORAL PREFETCHING

To the best of our knowledge, mobility modeling has not been incorporated with routing in MANETs. To prefetch a page just in time, the following information is needed. See figure 1.

1. The URL, U, the user is likely to access.
2. The time, T, at which the user is likely to request the page.
3. The likely position, X, of the user at T.

When nodes are mobile, the routing changes with time. Page P has to be delivered at time T and the routing information depends on the user's spatial position X. Hence information about P, T, and X is needed for just-in-time delivery.

We use a probabilistic prefetch model for P which uses additional metadata from the server.

As mentioned before, existing literature on prefetching focuses on *which* pages to prefetch and not *when* to prefetch. The estimation of time of prefetch requires a model for user's browsing habits. For time estimation, we assume that the page P is given. Given a hypertext with embedded link

P, the spatial position of the P in the displayed document can be easily obtained using the form factor of the device, browser settings (like font), document layout, etc. Let L denote the "position" of the embedded link P in the document. (This assumes a linear or one dimensional layout. For complex document layouts, other parametrizations are needed.)

6.1. User model

The browsing behavior depends on the form factor of the device (PDA, laptop, etc.), mobility pattern (walking, in a car, etc.), environment (working, in a discussion, etc.), etc. On small form factor devices, reading tends to be slow even for fast readers. Some factors like display form factor are fixed for a device. Factors like speed of movement can be calculated for a given session. Hence an initial estimate of user's reading speed is estimated using the above two parameters.

As the user starts browsing, other parameters like scrolling speed and actual link clicks are measured. Using these parameters, a Markov model is used to predict the time instant, T, of next click. The model is trained on actual user data.

Consider a page with n links $l_1, l_2, \cdots, l_n$. See figure 2 for the page layout and figure 3 for the page traversal model. $l_1, l_2, \cdots, l_n$ are the hyperlinks in the page. t_i is the time to reach link l_i after traversing link l_{i-1}. If no links are clicked, the expected time to read the entire page is

$$q_1(t_1 + q_2(t_2 + \cdots + q_{n+1}t_{n+1}))$$

and the time to reach link l_i

$$q_1(t_1 + q_2(t_2 + \cdots + q_i t_i))$$

But each link has a probability of clicking associated with it. Let p_i be the probability of clicking link l_i. If a link l_i is clicked, then its traversal takes place depending on the document length and the hyperlinks it contains. Let T_i be the expected time spent in page l_i and the links it contains. Hence the time to reach link l_i is

$$\sum_{k=1}^{i}(t_k + p_k T_k)\prod_{j=1}^{k} q_j$$

and the time T to read the page is

$$T = \sum_{k=1}^{n}(t_k + p_k T_k)\prod_{j=1}^{k} q_j + t_{n+1}\prod_{j=1}^{n+1} q_j \qquad (1)$$

The above model has the following parameters p_i, q_i, and t_i. The web server can estimate p_i. The estimation of q_i requires knowledge of user browsing mode ("surfing" or "conservative") which is available only to the client. The estimates of t_i require knowledge of display size, etc. which is also available only with the client. Hence the recursive computation of Ts require information available with both the server and the client. Since the client does not contain sufficient resources to perform the calculations, we let the server estimate T.

The following issues need to be addressed in the above formulation.

1. Calculation of T. It can be seen from equation 1 that computation of T is iterative. Ideally, the Ts should be computed using information of T_i from the entire web. Because of resource limitations, it is not possible for each server to make this calculation using the entire web data.

2. Normalization of T. Note that the time to reach a link depends on the display size, font size, etc. Since Ts are calculated by the web servers, it is difficult to incorporate this information at the server. Hence we use generic assumptions in the calculation of T by the server. The value sent by the server is normalized by the client depending on actual user behavior.

3. User mode: surfing or conservative.

When calculating T, the web server considers only the pages it contains. All the external pages are ignored. Also, the server breaks any cyclic references. This makes the calculation of T simpler. The calculations are performed for the "conservative" mode. That is, all q_is are set to 1. The t_i are estimated based on the model of a standard reader (250 words per minute, no scrolling, etc.) The p_is are estimated according to the actual usage statistics. Based on these assumptions and simplifications, the server is able to calculate T using equation 1.

When the server sends a page to the wireless client, it sends T_ks for each hyperlink l_k it contains. The device calculates a slow down factor σ based on its display size, user's scrolling, etc. The time estimate T returned by the server is multiplied by σ to arrive at the time at which the user is likely to visit a link. The user can be in surfing or conservative mode. This information is also used in the estimation of σ. For surfing, we use $q_k = 1/2$ and for conservative mode, $q_k = 1$.

6.2. Predictive routing

We now extend the routing protocols to incorporate time information. Each packet contains the expected time at which the packet should be delivered. Since each node tracks the movement of neighboring nodes, the routing tables also contain the time duration for which the routing information is valid. Since the routing table contains entries based on predicted node positions, we call it *predictive routing*.

The conventional routing table has the following format.

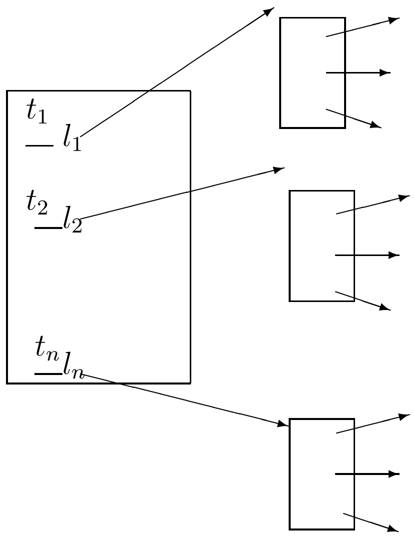

Fig. 2. Page layout. $l_1, l_2, \cdots, l_n$ are the hyperlinks in the page. t_i is the times to reach link l_i after traversing link l_{i-1}.

Destination	Next hop interface
A	X
B	Y

Any packet to A is routed to X and packets to B are routed to Y. The routing table changes when there is topology change (for proactive algorithms) or when there is a packet to a destination after topology change (for reactive algorithms).

In predictive routing, the routing table entries associated with a duration field as shown below.

Destination	Next hop interface	Time duration
A	X	$[t_1, t_2]$
A	Y	$[t_2, t_3]$
B	Y	$[t_2, t_3]$

If the delivery time, t, of a packet destined to A is in the range $[t_1, t_2]$, the packet is forwarded to X. If $t \in [t_2, t_3]$, the packet is forwarded to Y. Y, in turn, performs predictive forwarding based on t.

7. CONCLUSION

In this paper, we have proposed the notion of *personalized spatio-temporal prefetching* of web pages to reduce latency in mobile ad hoc networks. We have also proposed several novel techniques for characterizing user behavior and the integration of mobility and prefetching. Implementation of the ideas discussed in this paper will require services from the *pervasive grid* [10] infrastructure.

8. REFERENCES

[1] C. E. Perkins, *Ad Hoc Networking*. Addison Wesley, 2000.

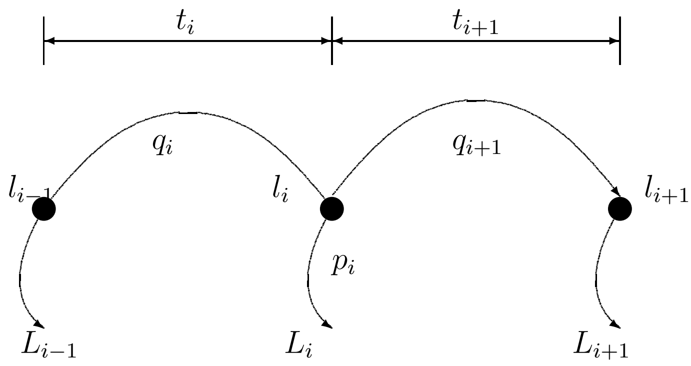

Fig. 3. Page traversal model. As before, $l_1, l_2, \cdots, l_n$ are the hyperlinks in the page. q_i is the probability of the user reaching link l_i from link l_{i-1}. In the "surfing" mode, q_is are small. In the "conservative" mode, q_is are close to 1. The time to traverse the segment between l_{i-1} and l_i is t_i. The hyperlink l_i leads to page L_i. The probability of clicking l_i is p_i.

[2] D. Duchamp, "Prefetching hyperlinks," in *USENIX Symposium on Internet Technologies and Systems*, 1999.

[3] X. Wang and Z. Yang, "Joint mobility tracking and hard handoff in cellular networks via sequential Monte Carlo fitlering," in *INFOCOM*, 2002.

[4] B. Li and K. H. Wang, "Nonstop: continuous multimedia streaming in wireless ad hoc networks with node mobility," *IEEE Journal on Selected Areas in Communications*, 2003.

[5] C. Hue, J.-P. L. Cadre, and P. Perez, "A particle filter to track multiple objects," in *IEEE Workshop on Multi-Object Tracking*, 2001.

[6] Z. R. Zaidi and B. L. Mark, "A mobility tracking model for wireless ad hoc networks," in *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC'2003)*, 2003.

[7] C. R. Cunha and C. F. B. Jaccoud, "Determining WWW user's next access and its application to prefetching," Boston University, CS Dept, Tech. Rep. TR-95-011, 1997.

[8] E. M. Royer and C.-K. Toh, "A review of current routing protocols for ad hoc mobile wireless networks," *IEEE Personal Communications*, vol. 6, no. 2, 1999.

[9] Z J Haas et al, "Wireless ad hoc networks," in *Wiley Encyclopedia of Telecommunications*. Wiley, 2002.

[10] S. H. Srinivasan, "Pervasive wireless grid architecture," in *WONS*, 2005.