AD^{2}P: An Asynchronous Data Delivery Protocol in Ad hoc Wireless Networks

Kyungseo Park, Ramez Elmasri, and Wook Choi
Department of Computer Science and Engineering
The University of Texas at Arlington
Arlington, TX 76019-0015
{kpark,elmasri,choi}@cse.uta.edu

Abstract

In this paper, we propose a protocol, called Asynchronous Data Delivery Protocol (AD^{2}P), in multi-hop wireless networks for processing time-uncritical data which needs to be stored in or retrieved from a database in the Internet using the existing TCP-based applications.

1 Introduction

Due to the users' ever-increasing needs for accessing the Internet 'anytime' and 'anywhere', mobile hosts in ad hoc wireless networks need to have an access to the wired network environment. In order to achieve this, a (mobile) host can act as a (fixed) gateway within the ad hoc wireless network domain. But in ad hoc wireless networks, the nodes frequently experience link disconnections due to node mobility. Meanwhile, TCP is still the major protocol when a user uses Internet. But since TCP is designed for wired networks, it cannot distinguish network congestion from link disconnection, which results in decreasing transmission rate [1]. These two drawbacks make data transmissions between wireless and wired networks inefficient and unreliable. To overcome this unreliability and inefficiency in such data transmissions, we propose a novel protocol, called AD^{2}P, for time-uncritical data delivery which does not require immediate response from the final destination. The AD^{2}P is based on a *Forward or Delay (FoD)* algorithm which realizes congestion control and rerouting technique. AD^{2}P is a modified version of ADTP (Asynchronous Data Transport Protocol) [2].

2 Asynchronous Data Delivery Protocol

In this paper, we make the following assumptions; i) nodes are aware of the location of Internet gateway and their own location based on Global Positioning System (GPS), ii) routing is based on a location-aided greedy routing scheme maintaining local neighbors, denoted by Nh_{v_i}, such as GEDIR (Geographic Distance Routing) [4], and iii) routing protocol knows the congestion ratio of the local neighbors. AD^{2}P runs in the application layer and selects a local neighbor as the intermediate destination to forward a message. Thus, a message delivery in AD^{2}P is composed of multiple hop-by-hop data deliveries (see Figure 1).

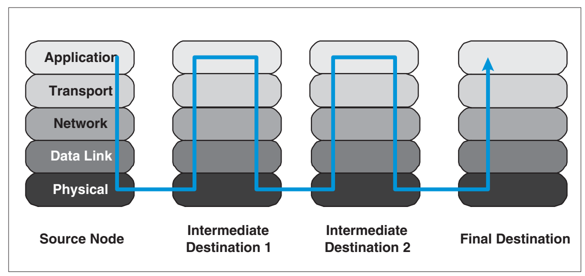

Figure 1. Hop-by-Hop Message Delivery

The architecture of AD^{2}P consists of four main components, proxy manager (PM), data transfer agent (DTA), input local data agent (ILDA), and output local data agent (OLDA). The PM manages proxy nodes by periodically accessing the one-hop local neighbor information maintained in the routing layer. Upon receiving a message, the DTA invokes the PM by sending a request. If the PM returns a null value due to the lack of proxy nodes, the DTA invokes ILDA, which places the message in a queue. Otherwise, it relays the message to the next-hop node chosen by the PM. The OLDA dequeues the message (e.g., query/result) queued by the ILDA, and passes it to the DTA.

We use a location-aided routing protocol that uses GPS, and we detect congestion ratio in each node by measuring the number of packets in the queue. All nodes periodically broadcast their congestion ratio as well as their coordinates. Thereby, a node can collect its local neighbors' location information and congestion ratio, denoted by $cr = \frac{current\_queue\_traffic\_load}{total\_queue\_length}$. Upon receiving a message, the *FoD* algorithm, which is a main part in AD^{2}P, is invoked to decide whether to delay or forward the mes-

Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks (LCN'03)

```
function best_proxy (v_i, v_d) begin
  D ← EuclideanDistance(v_i,v_d)
  for (all v_p ∈ Nh_{v_i})
      Local_Neighbors ←
        { v_p | EuclideanDistance(v_p,v_d) ≤ D ∩ v_p.cr < δ}
  end for
  if (Local_Neighbors = φ)
  then
      return null
  else
      for (all v_p ∈ Local_Neighbors)
          Max_LET ← {v_p.let|Max(LET(v_i, v_p))}
          Proxy ← {v_p|Max((1 − v_p.cr) + v_p.let / Max_LET)}
      end for
      return Proxy
  end if
end function
```

Figure 2. Pseudo Code for Best_Proxy Algorithm

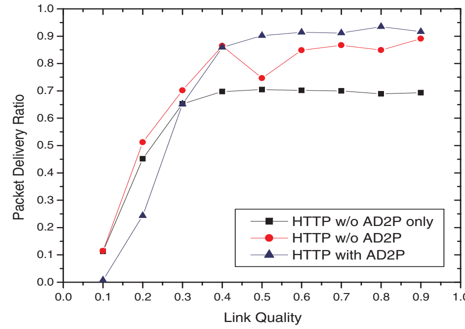

Figure 3. Packet Delivery Ratio Related to Link Quality

sage. The decision is based on the return value from a function *best_proxy*, which is shown in Figure 2 where v_i is the source or one of intermediate nodes; v_d is the final destination and v_p is a proxy node. If the best_proxy function returns a proxy node, FoD forwards the message to the proxy node. Otherwise, FoD delays forwarding the message in order to avoid putting more traffic load on the already-overloaded nodes or links, which may lead to either the message drop or high retransmission rate.

In the function best_proxy, the method of choosing the best reliable proxy node for transmitting a message to the destination server is to use a heuristic method that satisfies two metrics; i) congestion ratio and ii) LET(Link Expiration Time)[3]. LET is an expected time for two nodes to maintain their connection based on their current positions and heading directions.

3 Performance Evaluations

In our experiments, a discrete time event-driven simulator, called SIMLIB, is used. One hundred nodes are randomly distributed in the 800m x 800m size network and two nodes are randomly selected as a source and a destination, respectively, for a communication pair. For the movement pattern, nodes choose a random direction between 0° and 360° with the velocity ranging from $0m$/sec to $30m$/sec. A location-aided routing protocol, GEDIR [4], is used in this simulation and the congestion ratio and velocity of local neighbors are collected during the local neighbor discovery phase.

We chose the packet delivery ratio which is the number of received packets divided by the number of sent packets, as a main performance metric. First, we use a single traffic load (i.e., *HTTP w/o AD^2P only*) and then we use two different types of traffic loads (i.e., *HTTP w/o AD^2P* and *HTTP with AD^2P*) simultaneously to see the difference where '*HTTP w/o AD^2P*' represents HTTP packets with no help of AD^2P and '*HTTP with AD^2P*' represents HTTP packets routed by AD^2P (also called AD^2P packets). In Figure 3, X-axis represents the link quality whose value means the probability of the link stability. And the Figure 3 shows that the packet delivery ratios of the 'HTTP w/o AD^2P' and 'HTTP with AD^2P' packets are higher than that of 'HTTP w/o AD^2P only' packets. This means that AD^2P algorithm can alleviate network load, thereby increasing the packet delivery ratio of both HTTP and AD^2P packets. That can realize a win-win situation. As the second simulation, we set X-axis to the congestion ratio and the result is similar to the previous one. Finally, end-to-end packet delays are measured. The end-to-end delay of AD^2P packets is generally 3∼11 times bigger than that of HTTP packets. However, we focus on delivering time-uncritical data so that the delay is not considered to be important. In other words, the point is that the packets can reach the destination eventually even if there are some delays in some nodes.

4 Conclusions and Future Works

We proposed a novel hop-by-hop data delivery protocol, called AD^2P which decides delaying or forwarding a message depending on local congestion ratio and LET in ad hoc wireless networks. Since the AD^2P is designed for reliable message delivery under network congestion and frequent link disconnections, our simulation result showed that AD^2P has the highest message delivery ratio among all the simulated protocols. We plan to improve measuring LET further using a more predictable mobility model and validate the performance of AD^2P.

References

[1] K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash, "A Feedback-Based Scheme for Improving TCP Performance in Ad Hoc Wireless Networks", *IEEE Personal Communications*, February 2001, pp. 34-39

[2] J. D. Kwak, R. Elmasri, W. Choi, R. Ye, and A. Mahareddy, "ADTP: A New Protocol for Data Processing over the Internet", *Fourth International Workshop on Network-Based Information Systems (NBiS 2001)*, 2001.

[3] S. Lee, W. Su, and M. Gerla, "On-Demand Multicast Routing Protocol in Multihop Wireless Mobile Networks", *Mobile Networks and Applications*, Vol. 7, no. 6, 2002, pp. 441-453

[4] I. Stojmenovic, "Position-Based Routing in Ad hoc Networks", *IEEE Communication Magazine*, July 2002, pp. 128-134