ATP: A Reliable Transport Protocol for Ad Hoc Networks

Karthikeyan Sundaresan,
Vaidyanathan Anantharaman,
Hung-Yun Hsieh,
Raghupathy Sivakumar

IEEE Transactions on Mobile Computing, 2005
Outline

- Introduction
- ATP design
- ATP protocol
  - Intermediate node
  - Receiver
  - Sender
- Performance evaluation
- Conclusions
Introduction

- Ad hoc networks are characterized by several factors:
  - Lack of a fixed infrastructure
  - Mobility
  - Shared channel
  - Limited bandwidth

- Wired network protocols at the different layers inappropriate for ad hoc networks
At the transport layer, TCP suffers poor performance:

- Time varying link characteristics and node mobility

- Topology changes often occur and lead to packet losses and delays

- TCP misinterprets as congestion and reacts by reducing the transmission rate
Introduction (cont.)

- a new transport protocol: ATP (ad hoc transport protocol)
  - Tailored toward the characteristics of ad hoc networks
ATP design

- ATP is consists of
  - Rate-based transmissions
  - Quick-start during connection initiation and route switching
  - Network supported congestion detection and control
ATP design (cont.)

- No retransmission timeouts
- Decoupled congestion control and reliability
- Coarse-grained receiver feedback
ATP design (cont.)

- Layer Coordination
  - Use lower layer information and explicit feedback
  - ATP uses **feedback** for three different purposes
    - Initial rate feedback
    - Progressive rate feedback
    - Path failure notification

- Rate-Based Transmissions
ATP design (cont.)

- Decoupling of Congestion Control and Reliability
  - Congestion control uses feedback from the network
  - Reliability is ensured through SACKs
Assisted Congestion Control

The intermediate nodes provide congestion information in terms of the available rate.

Each node maintains two parameters:

- $Q_t$: the average queuing delay experienced by packets traversing that node

- $T_t$: the average transmission delay experienced by the head-of-line packet at that node
ATP design (cont.)

- Every epoch period, the receiver sends rate feedback to the sender.

- The sender, based on its current rate, and the rate specified determines to increase, decrease, or maintain its rate.
ATP protocol

- Intermediate Node
  - For every outgoing packet, an intermediate node updates its $Q_t$ and $T_t$
    
    $$Q_t = \alpha \cdot Q_t + (1 - \alpha) \cdot Q_{\text{sample}},$$

    $$T_t = \alpha \cdot T_t + (1 - \alpha) \cdot T_{\text{sample}},$$

  - Feedback field $D$ consists of the maximum $Q_t + T_t$ at the nodes the packet has traversed through
ATP protocol (cont.)

- Header Formats

Fig. 10. ATP packet header formats.

(a) Data packet header. (b) Feedback packet header.
ATP protocol (cont.)

- When a connection starts-up or sender sends a probe packet
  - except there is no other traffic around the node

\[ D = \eta \times (Q_t + T_t) \]
ATP protocol (cont.)

- ATP Receiver: provide periodic feedback
  - Rate Feedback
    \[ D_{avg} = \beta \cdot D_{avg} + (1 - \beta) \cdot D. \]
  - Flow Control Feedback
    - Observe the rate \( R_{app} \) which the application process in-sequence data
    - Compare with the rate feedback to determine \( D_{avg} \)
ATP protocol (cont.)

- Header Formats

Fig. 10. ATP packet header formats.

(a) Data packet header. (b) Feedback packet header.
ATP protocol (cont.)

- Reliability Feedback
  - Use selective ACKs (SACKs)
  - Not provided for every incoming data packet, but on a periodic basis
  - Use a larger number of SACK blocks than TCP-SACK (20 SACK blocks)
ATP protocol (cont.)

- ATP Sender
  - Quick-Start
    - Use quick-start to probe for the available network bandwidth in a RTT
ATP protocol (cont.)

Initial Rate Estimation:
Sender
1. send probe packet

Intermediate node
2. Compute $Q_t + T_t$ for packet
3. if $(\text{Avg}(Q_t) + \text{Avg}(T_t)) > \epsilon$
   
   $$\text{Avg}(Q_t) = \alpha \cdot \text{Avg}(Q_t) + (1 - \alpha) \cdot \text{Current}(Q_t)$$
   $$\text{Avg}(T_t) = \alpha \cdot \text{Avg}(T_t) + (1 - \alpha) \cdot \text{Current}(T_t)$$
4. if $(\text{Avg}(Q_t) + \text{Avg}(T_t)) > \text{stamped } D$
   
   $$\text{stamped } D = \text{Avg}(Q_t) + \text{Avg}(T_t)$$
5. else
6.   $$D_{\text{projected}} = i \cdot (\text{Current}(Q_t) + \text{Current}(T_t))$$
7.   if $(D_{\text{projected}} > \text{stamped } D)$
8.     $$\text{stamped } D = D_{\text{projected}}$

Receiver
9. Set $\text{Avg}(D) = \text{Current}(D)$
10. send $\text{packet}_\text{feedback}$ to sender with $\text{Avg}(D)$

Sender
11. $\text{packet}_\text{feedback}$ received with $\text{Avg}(D)$
12. compute rate $R = \frac{1}{\text{Avg}(D)}$
13. $\text{send}_\text{rate } S = R$
14. $\text{start}_\text{epoch}_\text{timer}()$
15. $\text{send}_\text{packet}()$

Fig. 7. Pseudocode for quick-start.
ATP protocol (cont.)

- Congestion Control
  - Use a **three-phase** congestion control protocol consisting of increase, decrease, and maintain phases

  - Increase phase: $R > S + \varphi \cdot S$
  - Decrease phase: $R < S$
  - Maintain phase: $S \leq R \leq S + \varphi \cdot S$
ATP protocol (cont.)

Fig. 8. Pseudocode for normal operation.

Normal Operation:
Intermediate node
1 Compute $Q_t + T_t$ for packet
2 if($\text{Avg}(Q_t) + \text{Avg}(T_t) > \epsilon$)
3 \hspace{1cm} $\text{Avg}(Q_t) = \alpha \ast \text{Avg}(Q_t) + (1 - \alpha) \ast \text{Current}(Q_t)$
4 \hspace{1cm} $\text{Avg}(T_t) = \alpha \ast \text{Avg}(T_t) + (1 - \alpha) \ast \text{Current}(T_t)$
5 \hspace{1cm} if ($\text{Avg}(Q_t) + \text{Avg}(T_t) > \text{stamped } D$)
6 \hspace{2cm} stamped $D = \text{Avg}(Q_t) + \text{Avg}(T_t)$

Receiver
7 \hspace{1cm} $\text{Avg}(D) = \beta \ast \text{Avg}(D) + (1 - \beta) \ast \text{Current}(D)$
8 \hspace{1cm} \text{On epoch timer expiry}
9 \hspace{2cm} stamp $\text{Avg}(D)$ on packet$_{feedback}$
10 \hspace{2cm} send packet$_{feedback}$ to sender

Sender
11 \hspace{1cm} packet$_{feedback}$ received with $\text{Avg}(D)$
12 \hspace{1cm} Compute new rate $R = \frac{1}{\text{Avg}(D)}$
13 \hspace{1cm} Rate Adjustment:
14 \hspace{2cm} if send rate $S < R - \phi \ast S$
15 \hspace{3cm} $S = S + 1$
16 \hspace{2cm} else if $S > R$
17 \hspace{3cm} $S = R$
18 \hspace{3cm} else maintain $S$
19 \hspace{1cm} start epoch timer()
20 \hspace{1cm} send packet()
ATP protocol (cont.)

- It is possible that the rate feedback from the receiver is lost
  - Perform a multiplicative decrease of the sending rate for every epoch it does not receive feedback from the receiver

- If it does not receive any feedback at the end of the third epoch
ATP protocol (cont.)

- Reliability

**Reliability Operation:**

**Receiver**
1. *On epoch_timer expiry or receipt of probe pkt*
2. identify all holes encountered
3. *stamp number of SACK blocks (len) on packet_feedback*
4. *stamp SACK blocks starting from first hole on packet_feedback*
5. send *packet_feedback* to sender

**Sender**
6. *On packet_feedback receipt with SACK information*
7. *update scoreboard and identify pkts for retransmission*
8. *On send_timer expiry*
9. pkts marked for retransmission are sent preferentially

Fig. 9. Pseudocode for reliability operation.
Performance evaluation

Fig. 11. Congestion window/rate progression v.s time [one flow].

(a) Default TCP.
(b) TCP-ELFN.
(c) ATP.
Performance evaluation (cont.)

Fig. 12. Congestion window/rate progression v.s time [25 flows].

(a) Default TCP.
(b) TCP-ELFN.
(c) ATP.
Performance evaluation (cont.)

Fig. 13. Instantaneous throughput dynamics [two flows].
(a) ATP rate adaptation. (b) TCP congestion window adaptation.
Performance evaluation (cont.)

Fig. 14. Throughput versus mobility.
(a) One flow.
(b) Five flows.
(c) 25 flows.
Conclusions

- We proposed a new transport protocol called ATP, which addresses the problems that TCP faces over ad hoc networks.

- ATP shows considerable performance improvement over TCP, TCP-ELFN, and ATCP.