MATP ——A New Transport Protocol

Wei Wang 5100309834

Abstract

The objective of this project is to develop a new transport protocol for MANET. The results will be analyzed from implementation and simulations in ns-2. An existing transport protocol , ATP[1] ,is the main inspiration of my design. And my design ,MATP(Mobile Ad-hoc Network Transport Protocol), is different from that of ATP in that a direct link between sender and receiver is added to the whole process.

1. Introduction

In a wired network, TCP is utilized as the transport control protocol that enables the network to transport packets reliably. However, when it comes to mobile network, some critical aspects of wireless network should be taken into consideration which form great influence on the working mode and packet-loss of the network. As is known to us, a wireless node could be moving at certain speed, yet the range of signal at certain SNR is limited, these two aspects make it important that the route be established from time to time. So a reliable transport could not be guaranteed if we use TCP , which is designed under the circumstances of wired network.

Ad-hoc networks are uniquely characterized by the several factors that differentiate them from traditional computer networks:

(i) Lack of a fixed infrastructure: Due to absence of dedicated routers, mobile hosts in ad-hoc networks also serve as peer-to-peer relays for connections in the network.

(ii) Mobility: All hosts in the net-work are mobile, and hence the network topology can be highly dynamic. From the perspective of a single end-to-end connection, not only are the end-hosts mobile, but the intermediate “routers” are mobile too.

(iii) Shared channel: Because of the all-wireless nature of ad-hoc networks, not only do flows in the same vicinity contend with each other, but part of a flow traversing multiple hops can contend with other parts of the same flow in its vicinity.

(iv) Limited bandwidth: While mobile hosts in general can be assumed to possess fewer amounts of resources than their static (wire line) counterparts, the wireless channel bandwidth is also scarce, resulting in multi-hop flows typically enjoying limited bandwidths of at most a few hundred kilobits per second.

At the transport layer, several works have focused on both studying the impact of using transmission control protocol (TCP) as the transport layer protocol, and improving its performance either through lower layer mechanisms that hide the characteristics of ad-hoc networks from TCP, or through appropriate modifications to the mechanisms used by TCP. Given the almost universal use of TCP as the transport layer protocol in the current Internet, such works are clearly warranted. However, several applications of ad-hoc networking, including more promising ones such as military battlefields, disaster relief operations, etc., are environments where a completely revamped protocol stack tailored to the operating conditions is not merely feasible, but also justifiable.[1]

We first address the shortcomings of TCP over ad-hoc networks ,then we discuss about how to solve these problems by modifying the working mechanism .

2. TCP Characteristics

2.1 Window Based Transmissions

TCP is a window based protocol. One of the underlying motivations behind such a design choice is to avoid the maintenance of any fine-grained timers on a per-flow basis. For wire-line environments, where per-flow bandwidths can scale up to several megabits per second, such a design choice is clearly essential. However, the use of a window based transmission mechanism in ad-hoc net-work networks results in the critical problem of burst in packet transmissions.[1]

TCP relies on self-clocking (ACKs arriving to trigger further transmissions) in the absence of timers. Thus, if several ACKs arrive back-to-back at the sender, a burst of data packets will be transmitted by the sender even if it were in the congestion avoidance phase (where one packet will be transmitted for every incoming ACK). Unfortunately, ACK bunching or several ACKs arriving at the same time is a norm in ad-hoc networks because of the short-term unfairness of the CSMA/CA MAC protocol typically used in such networks. provides a good exposition on the short term unfairness properties of CSMA/CA. Such short-term unfairness results in the data stream of a TCP connection assuming control of the channel for a short period, followed by the ACK stream assuming control of the channel for a short period. Interestingly, such a phenomenon will occur even when the ACK stream does not traverse the exact same path as the data stream. This is because evenif the paths were completely disjoint, the vicinity (2-hop region in the case of CSMA/CA) of the TCP sender and the vicinity of the TCP receiver still are common contention areas for the data and ACK streams.[1]

2.2 Slow Start

The slow-start mechanism is used by TCP both during connection initiation and when TCP recovers from what it perceives as heavy congestion in the network. For both cases, the goal of slow-start is to probe for the available bandwidth for the connection. When a connection is in the slow-start phase, TCP responds with two data packet transmissions for every incoming ACK. [1]

2.3 Loss Based Congestion Indication

TCP uses the occurrence of losses (inferred either through receipt of three duplicate ACKs, or occurrence of a timeout) to detect congestion.[1]

3. The MATP Design

in this part , I elaborate the details of the design of MATP, based on TCP and ATP[1] .The intermediate node provide congestion information about the receiver to the sender. The receiver provides both flow control and reliability to the sender via intermediate node or directly to the sender. The sender is responsible for collecting these information and make decisions according to the given information.

3.1 Intermediate Node[1]

When a packet sent from the sender arrives at a intermediate node, the intermediate node will attach a data concerning the time delay in the intermediate node to the packet. And the corresponding formula goes like this:

$$Q_t = \propto * \, Q_t + (1 - \propto) * Q_{sample}$$
$$T_t = \propto * \, T_t + (1 - \propto) * T_{sample}$$

In the above formula, Q_t stands for average queuing delay, T_t stands for average transmission delay. $\propto$is a constant between zero and one, and it stands for the weight of the previous data in the next data. $Q_{sample}$and T_{sample} are the queuing delay and transmission delay experienced by the packet.

In addition ,each packet consists of a rate feedback field D(actually, it denotes the inverse of the data rate). D represents the largest delay of a packet has traversed on the up-streaming nodes(the sender is not included).

So ,if D of a packet is smaller than the $Q_t + T_t$ of the current node ,then D is updated to $Q_t + T_t$.

3.2 ATP Receiver[1]

3.2.1 Rate Feedback

For every incoming packet belonging to a flow, the receiver performs an exponential averaging of the D value specified in the packet:

$$D_{avg} = \beta * D_{avg} + (1 - \beta) * D$$

3.2.2 Reliability Feedback

The MATP (just like ATP does) uses selective ACKs(SACKs) for providing information about losses in the data stream received.

3. 3 ATP Sender[1]

3.3.1 Quick Start

Initial Rate estimation:

Sender

Send probe packet

Intermediate Node

Compute $Q_t + T_t$ for Packet:

$if\left(Avg(Q_t) + Avg(T_t)\right) > \epsilon)$

$Avg(Q_t) = \propto \times Avg(Q_t) + (1 - \propto) \times current(Q_t)$

$Avg(T_t) = \propto \times Avg(T_t) + (1 - \propto) \times current(T_t)$

If$\left(\left(Avg(Q_t) + Avg(T_t)\right) > stamped(D)\right)$

$stamped(D) = Avg(Q_t) + Avg(T_t\ )$

else

$D_{projected} = i * (current(Q_t) + current(T_t))$

if ($D_{projected} > stamped(D)$)

$\mathrm{stamped(D) = projected(D)}$

Receiver

Set $\mathrm{avg(D) = current(D)}$

Send $packet_{feedback}$ to sender with $\mathrm{avg(D)}$

Sender

$packet_{feedback}$ received with $\mathrm{avg(D)}$

Compute rate $\mathrm{R} = \frac{1}{avg(D)}$

Send rate S = R

3.3.2 Congestion Control

Normal operation

Intermediate node

Compute $Q_t + T_t$ for Packet

$if\left(Avg(Q_t) + Avg(T_t)\right) > \epsilon)$

$Avg(Q_t) = \propto \times Avg(Q_t) + (1 - \propto) \times current(Q_t)$

$Avg(T_t) = \propto \times Avg(T_t) + (1 - \propto) \times current(T_t)$

If$\left(\left(Avg(Q_t) + Avg(T_t)\right) > stamped(D)\right)$

$$stamped(D) = Avg(Q_t) + Avg(T_t)$$

else

$D_{projected} = i * (current(Q_t) + current(T_t))$

if ($D_{projected} > stamped(D)$)

$\mathrm{stamped(D)} = \mathrm{projected(D)}$

Receiver

Set $\mathrm{avg(D)} = \mathrm{current(D)}$

Send $packet_{feedback}$ to sender with $\mathrm{avg(D)}$

Sender

$packet_{feedback}$ received with $\mathrm{avg(D)}$

Compute rate $\mathrm{R} = \frac{1}{avg(D)}$

Rate adjustment :

if sendrate $\mathrm{S} < \mathrm{R} - \emptyset * \mathrm{S}$

$\mathrm{S} = \mathrm{S} + \frac{R-S}{k}$

else if $\mathrm{S} > \mathrm{R}$

$\mathrm{S} = \mathrm{R}$

($\emptyset$ and k) are constants

3.4 No Intermediate Case

When there is no intermediate between sender and receiver, although this might be rare in today's wireless communication networks , there is still something need to be taken into consideration in the designing of a protocol.

One bit of flag INnum(0 or 1) will be added to the packet, INnum will be set 0 when initialization. If there is no intermediate ,it would be kept the same. If there is one or more intermediate nodes, it will be set to 1.

The whole process will be like this:

3.4.1 Quick Start

Initial Rate estimation:

Sender

Send probe packet

Intermediate Node(Non-existent)

Receiver

Check INnum = 0

Set $\mathrm{avg(D)} = T_0$(a constant representing the time delay from sender to receiver)

Send $packet_{feedback}$ to sender with $\mathrm{avg(D)}$

Sender

$packet_{feedback}$ received with $\mathrm{avg(D)}$

Compute rate $\mathrm{R} = \frac{1}{avg(D)}$

Send rate S = R

3.4.2 Congestion Control

Normal operation

Receiver

Set $\mathrm{avg(D)} = \mathrm{T_0}$

Send $packet_{feedback}$ to sender with $\mathrm{avg(D)}$

Sender

$packet_{feedback}$ received with $\mathrm{avg(D)}$

Compute rate $\mathrm{R} = \frac{1}{avg(D)}$

Rate adjustment :

$\text{if sendrate } \mathrm{S} < \mathrm{R} - \emptyset * \mathrm{S}$

$\mathrm{S} = \mathrm{S} + \frac{R-S}{k}$

$\text{else if } \mathrm{S} > \mathrm{R}$

$\mathrm{S} = \mathrm{R}$

($\emptyset$ and k) are constants

4 Conclusion

As my research goes ,the biggest challenge lies in the fully understanding the tcp.c in NS-2, which generates really great obstacle in my design. Up till now ,I still haven't fully understand the whole working process of TCP or ATP, and my design was just a theoretical imagination with the idea from ATP.

But the idea still works its way out in the designing of a new protocol of transport protocol for mobile ad-hoc networks.

5 References

[1] Karthikeyan Sundaresan,Vaidyanathan Anantharaman,Hung-Yun Hsieh, and Raghupathy Sivakumar,"ATP :A Reliable Transport Protocol for Ad –hoc Networks" ,2003