ATP: Autonomous Transport Protocol

Tamer Elsayed, Mohamed Hussein, Moustafa Youssef, Tamer Nadeem, Adel Youssef and Liviu Iftode
Department of Computer Science
University of Maryland, College Park, MD 20742
Email: {telsayed,mhussein,moustafa,nadeem,adel,iftode}@cs.umd.edu

I. Introduction

As ubiquitous computing emerges; the users, not the end hosts, should become the focus of the communication. To achieve this goal, connections should be carried between users, independent from the host on which the user is located.

In this paper, we introduce the Autonomous Transport Protocol (ATP) which allows dynamic endpoint relocation without disrupting the transport connection between them. The ATP has the following features:

- It does not enforce any naming scheme on the user application. The application is responsible for uniquely identifying the endpoints.
- The endpoints of a transport connection are defined as contents in the content based network, which is an overlay communication platform over which end-point entities (called *contents*) communicate independently from their physical locations. This decouples the connection from the physical host where the user endpoint is located, and hence ensures autonomy.
- Mobility of the endpoints is handled by dynamically changing the mapping between the endpoints and the hosts using the new *instant-based network (IBN)* layer in which we enhanced the content-based network to provide additional functionalities [1]. The ATP layer is responsible for moving data segments to the destination and the acknowledgments to the source regardless of their current location in the network.
- Data is transferred by a combination of *active* and *passive* operations, where the ATP layer of a node can decide whether to actively push the data to the destination or to passively wait for the destination endpoint to pull the data. The decision to whether to use the active or passive modes can be taken by a local policy on the node running the ATP network.

The ATP protocol stack consists of four layers: the underlying-network, the IBN, the ATP, and the application layer [1].

II. Protocol Overview

The main goal of ATP protocol is to decouple the operation of the endpoints and maintain a reliable communication between them.

A connection in the ATP is established between two endpoints which are identified by their *content ID*'s. Endpoints could migrate or temporarily disappear from the network, and the data *segments* and *acknowledgments* should continue to flow between them.

Assuming a source endpoint *Src* establishes a connection with a destination endpoint *Dst*. The connection goes through the ATP layers on both nodes that *Src* and *Dst* are attached to and the intermediate nodes. In case of neither *Src* nor *Dst* migrates to different nodes, this mode of operation is similar to the operation of the standard TCP protocol over the IP networks. When the source endpoint migrates, the ATP layer on the old node spawns an agent which takes care of sending any data in the buffer and receives acknowledgments. The ATP layers on the new and old nodes cooperate to make the migration transparent to the destination endpoint. A similar process is applied for the *Dst* migration. The ATP layers on the new and old nodes cooperate to make the migration transparent to the source endpoint. The migration step can be performed multiple times and there can be multiple agents working for the same endpoint at any time.

The ATP agent can take the decision to participate actively in the connection or to wait passively for the destination to pull the data. In the former case, the node publishes (registers) itself in the network as *AS (Active Source)*, while in the latter the node publishes itself as *PS (Passive Source)*. The default mode of the agents acting on behalf of the source is the active mode. Agents acting on behalf of the destination should buffer the received data until the destination appears on a new node. Therefore, these agents stay in the passive mode until the destination reappears and *pulls* the buffered data.

Each agent has a unique name composed of the original content ID plus the starting sequence number of the data it is responsible for. For example, *AS:4* denotes an agent for the source endpoint in the active mode responsible for the segments starting with sequence number 4. This naming of the agents is supported by the underlying IBN where different instances (agents) of the same content are identified by the sequence number of the starting segment the agent is responsible for. A segment with destination address *D:j* is routed to the agent whose instance-ID is *D:j* if such agent exists. Otherwise, it is routed to the agent responsible for the maximum sequence number lower than j. For example, if agents *AS:0*, *AS:5*, and *AS:9* are handling a connection, then a segment with a destination address of *AS:8* is routed to the agent *AS:5*. Such addressing mechanism helps in achieving the protocol autonomy [1].

References

[1] T. Elsayed, M. Hussein, M. Youssef, T. Nadeem, A. Youssef, and L. Iftode, “ATP: Autonomous Transport Protocol,” Tech. Rep. UMIACS-TR-2003-52 and CS-TR-4483, University of Maryland, May 2003.