UDP Transport and Router Mini-SRS

1. INTRODUCTION

UDP Transport Enhancements and Router Optimization (UTERO) is project to provide the ability to transport AllJoyn messages over IP/UDP and extensions to the AllJoyn Routing Node to quickly route UTERO datagrams.

This document is an informal document containing the high-level system requirements of the UTERO project. This document is intended to be a vehicle to facilitate discussion and to communicate our high-level design informally. It can also be used as a guide during implementation. It is not intended to be a complete software requirements document in the formal sense.

1.1. VERSION

Version	Description	Author	Date
0.01	Capture High-Level Requirements	Craig Dowell	February 27, 2014

2. REQUIREMENTS

From a very high level perspective, UTERO must be lighter, faster and thinner than existing transports. Existing AllJoyn transports are very resource-hungry, requiring many sockets, many file descriptors, threads and very complicated routing code in the Routing Node. The existing UDP Transport, called the Packet Engine is designed along the lines of a standard AllJoyn transport, but is designed to operate over the Internet. As such, it must deal with wildly variable environments and must respect the requirements of the greater Internet to avoid congestion collapse. This results in a very heavyweight and complicated protocol that actually performs worse than TCP in the proximal case. UTERO should treat sockets and threads, and OS resources in general, as a precious resource and focus on a lightweight implementation.

> *Requirement: Use fewer resources than Existing AllJoyn Transports. There should be one instance of a UDP socket per Node that is shared among all endpoint equivalents. There should be a minimal number (goal one) thread/timer used to manage the protocol endpoint equivalents.*

TCP-based transports, since they are reliable transports, tend to do exactly the wrong thing in the presence of slow networks or responders. UTERO must be capable of providing reasonable latency bounding and be able to discard messages that have expired according to message TTL at any point in the message path accessible to user-level code.

> *Requirement: Latency bounding and intrinsic message TTL support. At the transport level, be able to examine message datagrams and extract TTL information. Based on TTL, discard expired messages. Real time latency guarantees are not a requirement.*

The TCP protocol is designed to operate over a wide variety of network types and to avoid congestion in the wider Internet. The resulting design avoids keep-alive messages at basically all cost. The cost to AllJoyn is difficulty in detecting the loss of a link in a reasonable time. Typically the idea of reasonable is defined by the application.

> *Requirement: Quick link loss detection. At the endpoint-equivalent level, UTERO must be able to quickly detect link loss conditions. Quickly should be defined by the endpoint-equivalent.*

In current AllJoyn transports, there is no concept of quality of service. This manifests as a problem when data flows congest a link and control flows are unable to move across the link. This often results in livelocks as control messages fight for room on the link. This can also result in deadlocks when unresponsive nodes clog links completely and prevent control messages from flowing through the system.

Requirement: Quality of Service. UTERO must be able to provide two levels of service: control plane and data plane. The control-plane-level QOS corresponds to AllJoyn system messages that are required for the correct operation of the overall system. Data-plane level messages are user data (method call, property, signal) messages. Control plane messaging must flow.

Since a variety of underlying network mechanisms are contemplated, there must be some mechanism to prevent high speed links from overwhelming lower speed links and either consume unlimited buffering resources or be unnecessarily dropped (for buffer occupancy instead of latency reasons).

Requirement: Throttling. UTERO must consider path link capacity and provide some mechanism to provide backpressure to an endpoint-equivalent in order to avoid unbounded buffering requirements for intermediate routing nodes. This mechanism must support enable low-power, low-speed (i.e.802.14.4) links.

There is an almost universal agreement that sessions as they exist today are a problem. Since UDP is connectionless, we can think about how we could optimize them out but retain compatibility. i.e. generate signaling at the endpoints.

Possible Requirement: Desessionization. UTERO should consider how to make sessions optional in the connectionless data path.

The use of UDP as an underlying message transport immediately raises the possibility of multicast addressing. While this is a desirable feature, we do not contemplate providing a reliable multicast mechanism for the initial release of UTERO.

Non-Requirement: Reliable multicast.

While reliable multicast per se is a non-requirement we do have the opportunity to greatly improve the latency of one-to-many message transactions by initially multicasting one copy of the message and ensuring reliability by unicasting duplicate copies to all destinations. While this requires one multicast and N unicasts, it allows for low latency (with error recovery) in cases where messages must be sent to many destinations (for example the lighting service controller).

Requirement: Fast point-to-multipoint messaging

UTERO is designed for proximal environments. The conventional meaning of proximal is awareness of and addressing to devices residing on the directly connected subnetwork. Since this intentionally excludes the greater Internet, messaging traffic need not implement the IETF/IANA requirements for congestion control mechanisms.

Non-Requirement: Congestion control.

Caveat: If organizations choose to route AllJoyn messages across their private systems using organization-local scope advertisements, they may do so. UTERO will not dynamically detect the *presence of multiple subnets and dynamically adjust by applying congestion control strategies. This may result in instability in highly loaded situations.*

3. ARCHITECTURE

See the “UDP Transport and Router Architecture” document.