Client/server messaging protocols in serverless environments

Justin Deanb, Andrew Harrisonc, Robert N. Lass a, Joe Mackerb, David Millara, Ian Taylorc, *

a Department of Computer Science, Drexel University, USA
b Naval Research Lab, Washington DC, USA
c School of Computer Science, Cardiff University, Cardiff, UK

A R T I C L E  I N F O

Article history:
Received 14 October 2010
Received in revised form
21 January 2011
Accepted 10 March 2011
Available online 17 March 2011

Keywords:
XMPP
Serverless chat
WS-Notification
SLP
JmDNS
Peer-to-Peer
Multicast
NORM

A B S T R A C T

In this paper we discuss the adaptation of TCP transport-oriented client–server messaging protocols to
many-to-many peer-to-peer networking environments more suitable for deployment in dynamic
wireless networks capable of multicast forwarding. We describe four main issues in adapting such
protocols: exposing a network server for receiving TCP session data; the creation of server-side
semantic proxies to process the messages and adapt to a serverless environment; service discovery
to enable the discovery of necessary services on the network and to maintain the network state;
and finally support for multicast interfaces for the transportation of messages amongst peers. We show that
our system, called GUMP, can be used to support such protocol adaptations and to illustrate we use
GUMP to implement an XMPP proxy allowing existing off-the-shelf XMPP client software to
dynamically create and operate multi-user chat sessions in a serverless network environment. We
then present two sets of results that show how appropriate discovery systems and transport protocols
can dramatically increase the success of protocols, such as XMPP, within a mobile wireless networked
environment. Specifically, we first demonstrate that a GUMP supported discovery system, INDI, can
significantly increase the success rates and decrease latency of discovering services through profiles,
caching and retrying schemes. Second, we show that success rates for XMPP transmission of messages
can be vastly improved through the use of multicast as opposed to TCP within the mobile environment.
These two factors provide strong empirical support for the justification of GUMP in its ability to adapt
between a client–server and serverless world.

1. Introduction

XML-based standardized messaging protocols (e.g., XMPP, Saint-Andre, 2004a,b, WS-Notification, Graham et al., 2006 and
WS-Eventing, Box et al., 2004) typically assume dependence on an
underlying TCP transport stack for achieving a reasonable level of
reliability across wide area networks. It follows therefore that
application clients (e.g., for XMPP, Pidgin1 and Spark2) that use
these protocols are also tied into TCP, often needing to make a
client–server connection to a gateway or management server in
order to join and use the network. This dependency does not work
well when the group application moves into a more dynamic
environment, such as a wireless mobile ad-hoc network (MANET,
Corson and Macker, 1999), where server location and availability
is in flux and where TCP transport effectiveness may be reduced
and even render the application unusable. One issue is that TCP
congestion control can often overreact to temporal disruptions
(e.g., wireless link errors, collisions, routing dynamics) not indi-
cative of queue congestion and reduce the rate or stall transport
connections.

UDP and transport enhancements (e.g., reliability) built above
UDP, on the other hand, have historically had a more restricted
portfolio of conventional internet usages, such as real-time
applications requiring low latency, real-time delivery, e.g., video
or audio streaming. However, given that design and performance
issues can be quite different in highly dynamic and disruptive
network environments, such as MANET, UDP-based algorithms
can often provide more effective results. Dynamic, wireless
networking environments are also often less suitable for centra-
lized server deployment due to potential server disruptions and
mobility. Another point is that multicast or group-oriented net-
work transport and delivery is often more suitable in wireless
environments and therefore collaborative applications may often
employ forms of UDP-based transport protocols or transport
Performance enhancing proxies (PEPs), rather than pure end-
to-end TCP.

The Extensible Messaging and Presence Protocol (XMPP) is a
set of open XML-based standardized messaging technologies for
presence and real-time communication developed by the Jabber

1 http://www.pidgin.im/
2 http://www.igniterealtime.org/projects/spark/index.jsp