The MaDMAN Middleware for Delay Tolerant Networks

Agoston Petz, Christine Julien
{agoston,c.julien}@mail.utexas.edu

1. OVERVIEW

Delay-tolerant networks (DTNs) are quickly gaining traction with industry and research given their applicability to "challenged" networking environments where standard ad-hoc network protocols and solutions are unable to provide connectivity due to mobility, sparse node densities, interference, etc. Such networks require delay-tolerant protocols since an end-to-end path between communicating nodes may never exist. Despite this, packet delivery is often possible given intermediate nodes which can ferry data between communicating parties. Ultimately, DTN protocols have to coexist with standard ad-hoc networking protocols since varying networking conditions dictate that no one protocol is best in every case. We present a middleware for delay-tolerant mobile ad-hoc networks (MaDMAN) which dynamically picks the appropriate protocols to use for a given connection, and can migrate live application sessions between protocols in response to changing networking conditions, we explain the architecture of our middleware, and our experiences building a preliminary implementation of it within the Click Modular Router [1]. To test the middleware, we make use of a real-world mobile ad-hoc delay-tolerant network comprised of autonomous robots with x86 hardware and commodity 802.11 wireless cards.

2. MADMAN MIDDLEWARE

MaDMAN's architecture, shown in Figure 1, is designed around the concept that a combination of traditional mobile ad-hoc protocols and delay tolerant protocols is what is required by the dynamic nature of DTNs (since different communication styles are suited to different underlying networking conditions). Using a combination of protocols allows the middleware to choose the optimal strategy. Thus we have architected the MaDMAN middleware to dynamically reconfigure the network stack even during active application sessions, without dropping the sessions. MaDMAN is capable of composing a potentially unique network stack for each connection. To support the dynamic selection of protocols we have defined a vertical component, the *context aggregator* which collects network context from each layer of the stack (e.g. signal-to-noise ratios, windows sizes, round trip times, throughput, etc.) and using this context, decides which combination of protocols is best suited for each connection. To support the integration of new and improved protocols, we decided to implement the MaDMAN architecture within the Click Modular Router [1], a flexible, modular, and stable framework for developing routers. Click has a large base of available network protocols and is itself a good framework for implementing experimental protocols, which we hope fosters the integration of new protocols from other researchers. Finally to test the performance of our MaDMAN implementation, we are using autonomous robots from the Pervasive Computing Testbed [2], which provide a mobile platform with configurable mobility models and standard x86 hardware (running Linux v.2.6) complete with Intel 802.11bg wireless cards.

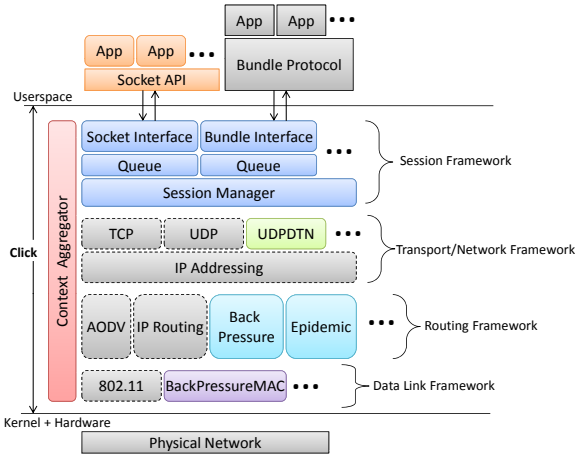

Figure 1: The MaDMAN Architecture

3. REFERENCES

[1] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. Kaashoek. The Click modular router. *ACM Transactions on Computer Systems*, 18(3):263–297, 2000.

[2] http://mpc.ece.utexas.edu/pharos/.