Multi-Modal Network Protocols

Rajesh Krishna Balan, Aditya Akella, Srini Seshan
Carnegie Mellon University
{rajesh+,aditya+,srini+}@cs.cmu.edu

Introduction

Many of the basic parameters of a ubiquitous computing environment are highly variable. For example, rate of mobility, quality of wireless communication, density of communicating nodes, power constraints and computational constraints all vary greatly from situation to situation. Unfortunately, existing routing protocols, transport protocols and adaptive applications can only adapt in a limited way. One feature that is missing from many systems is the ability to identify gross operating conditions and take on very different modes of operation. Such adaptability would improve the efficiency of many solutions. We use the term multi-modal protocols to describe systems that adapt in this way. We have explored two such systems: Wireless TCP and Overlay Routing.

Multi-modal Wireless TCP

Our initial experiments concentrate on adapting the behavior of TCP to the power conditions of the end-nodes. An end-node attempts to enable various power conservation modes based on battery conditions and policies. However, the exact mode of operation may depend on the availability of resources in the nearby environment and at the other connection endpoint. These power conservation modes often require tradeoffs (between throughput, power savings, etc.) and must be agreed upon by the two endpoints. For example, with base station support, the end-nodes can request the base stations to buffer packets for a specified period of time. The stack can then sleep the network device during this time and save power. Things become trickier in the absence of a base station (which happens in Ad-Hoc networks).

In situations without a base station, the end-node may be able to opportunistically determine gaps in the TCP transmissions and sleep during them. However, these gaps tend to be pretty random in the presence of cross traffic. Hence we need ways to artificially induce deterministic burstiness into the TCP transmissions. Changing the delayed ACK (delack) threshold of TCP introduces more of the desired regular burstiness into the transmission. Our preliminary results show that changing the delack threshold for TCP can result in power savings whenever the link is unsaturated. For example, this frequently occurs during slow start and when the window of a connection is constrained. However, increasing the delack threshold decreases the throughput of the connection. Hence, the loss in throughput has to be balanced with the savings in power. We also have simulation results that show that it is possible to experience some power savings by opportunistically sleeping during gaps in the transmission window. However, in the presence of cross traffic, these gaps become fairly random. Fig 1. shows the effect of changing the delack threshold from 2 to 8 packets. The burstiness of the data transmission increases while the throughput decreases.

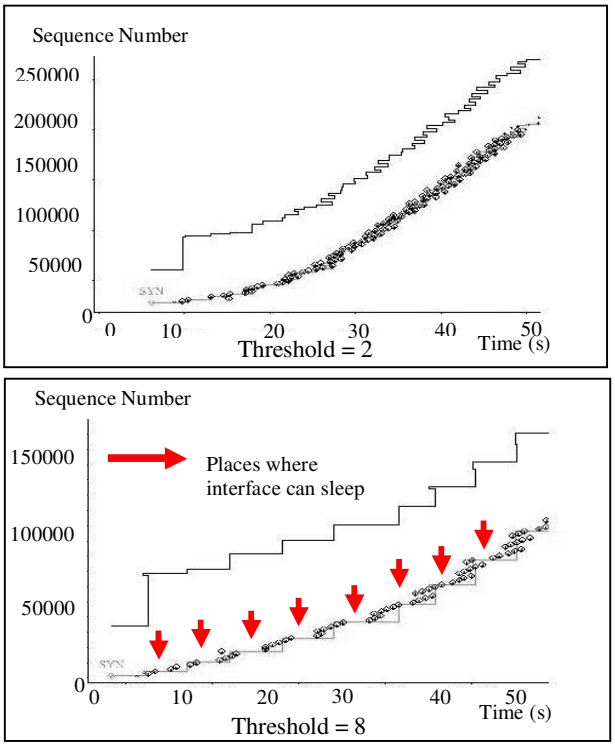

Fig 1: Effect of changing delayed ACK threshold

Multi-modal Overlay Routing

All overlay networking systems incorporate an algorithm to self-organize the participating nodes into a routing infrastructure. This self-organization requires the nodes to probe and discover performance characteristics of the underlying network. These systems often incorporate tradeoffs between the overhead of probing the network and the quality of the resulting overlay network.

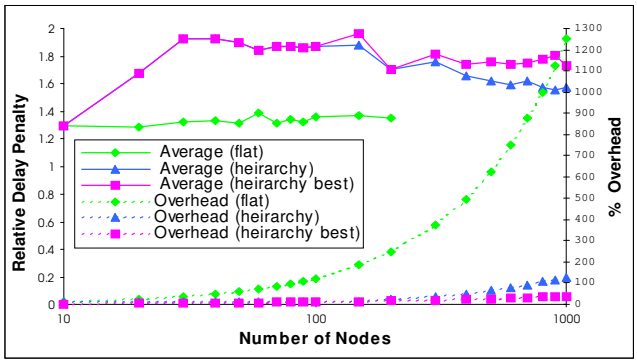

Fig 2: Hierarchical Versus Flat Routing

We have evaluated a system design that would automatically determine which overlay routing algorithm to use based on the current population size, density and availability of infrastructure etc. As Fig 2. shows, there is a clear tradeoff between latency and overhead for various overlay routing algorithms. For example, the latency (between end nodes in the overlay) for flat overlay routing algorithms is smaller compared with hierarchical routing algorithms. However, the network bandwidth probing overhead of the flat algorithms grows linearly with group size while the overhead growth of hierarchical algorithms is much slower. Hence, for small groups, flat routing algorithms make sense while for larger groups, hierarchical algorithms should be used. Having a system which could automatically change the overlay routing algorithm when the environment changed would be extremely useful as this would allow the creation of long-lived, highly variable content distribution networks and it would also allow application writers to write overlay applications without worrying about the underlying overlay routing algorithms.