Project Ideas

15-744 F02

TCP

Multi-path routing for a single TCP flow

Using RR-TCP, a single flow's packets can be spread over multiple paths without the severe throughput degradation normally caused by packet reordering.

But there are yet problems to consider:

- To maximize throughput, you'd like to choose paths with disjoint bottlenecks on which to send. Can you devise a system that allows a sender to identify paths with disjoint bottlenecks?
- What happens when two paths have different loss rates? If you naively keep a single window size at the sender, and have no knowledge of the different loss characteristics of the two paths (e.g., say you just alternate packets between the two paths), what behavior results? Is the sender unfair (too aggressive) on either path under any circumstances? Compare the throughput the sender achieves with that achieved by two individual flows, each routed separately over one path.
- Devise a system for improving upon the naive behavior you investigated above. One possibility:
- Suppose your TCP sender is directly informed of the number of paths to be used, and the loss rate on each path. (A system like RON might be able to provide this information.) Use this information to "color" packets appropriately; marking them for which paths they should take. How close to the sum of the actual available bandwidths on each path can you achieve, without sending too aggressively on any one path?

Contact: Brad Karp (bkarp@cs.cmu.edu)
Reading: RR-TCP (Reordering-Robust TCP) paper (Zhang, Karp, Floyd, Peterson; under submission to ToN) available at http://www.icir.org/bkarp/

Receiver-oriented RR-TCP

RR-TCP keeps a non-trivial amount of state at the sender: it keeps a histogram of all reorderings a flow experiences, and extra scoreboard information to detect when packets were retransmitted when they were only reordered, not lost. This design concentrates state at the server. Since servers typically have a great many open connections, there are concerns about memory requirements on the server.

Implement a receiver-oriented version of RR-TCP. That is, have the *receiver* measure the reordering a flow's packets experience, and explicitly inform the sender of the dupthresh value it should use.
Contact: Brad Karp (bkarp@cs.cmu.edu)
Reading: RR-TCP (Reordering-Robust TCP) paper (Zhang, Karp, Floyd, Peterson; under submission to ToN) available at http://www.icir.org/bkarp/

Mobile/Wireless

Mapping the geographic boundaries of a network

In a network where every node knows its own position, devise efficient algorithms for mapping the network's geographic boundaries. The context for this work is the GHT system, which needs to know the boundaries of the deployment area so that it can hash only to locations "inside" the network. You probably don't need to map a very detailed outline of the network; a simple polygon, like a square, is probably both easier to map and easier to use when constraining the hash function.

Tradeoffs to examine: How low can you keep the communication cost, vs. how accurately can you map the network boundary?, How quickly can you find the correct network boundary when the boundary changes?
Contact: Brad Karp (bkarp@cs.cmu.edu)
Reading: GPSR paper (Karp, Kung; Mobicom 2000) available at http://www.icir.org/bkarp/

Geographic routing with imperfect positioning

Investigate the behavior of GPSR when nodes don't know their own positions with perfect accuracy. For example, nodes using GPS devices may choose to power off their GPS receivers intermittently to save power. During this period, a node's knowledge of its own position (and others' knowledge of its position) will become stale, and interfere with correct routing. Investigate the tradeoffs between radio range, motion rate, and GPS sleep interval. Figure out how to make the routing system more robust to position inaccuracy. Example: doing 2-hop distance-vector routing, rather than 1-hop beaconing, will allow hosts within two hops to be reached, even if their correct positions aren't known. (On the other hand, increasing to 2-hop DV from 1-hop DV increases network load and power consumption...) Are there heuristics for "noticing" from passing traffic that one's position information is out-of-date, and that the GPS should be powered up now?
Contact: Brad Karp (bkarp@cs.cmu.edu)
Reading: GPSR paper (Karp, Kung; Mobicom 2000) available at http://www.icir.org/bkarp/

Directed diffusion vs. GHT bakeoff

Both directed diffusion and GHT have been proposed as data dissemination methods for sensor networks. Devise a few application scenarios and compare these two systems' performance (energy consumption, data traffic concentration, robustness, &c.). One aspect of the comparison that may be interesting to think about: directed diffusion essentially combines a sort of distance-vector routing with the application-level functionality of querying the sensor network. Does this integration offer advantages as the authors of the work claim? Or can a more layered system (such as GHT, which is an application write/query layer atop a geographic routing system) achieve similar energy savings?
Contact: Brad Karp (bkarp@cs.cmu.edu)
Reading: GHT paper (Ratnasamy, Karp, et al.; WSNA 2002) & Data-Centric Storage paper (Shenker, Ratnasamy, Karp, et al.; HotNets 2002) available at http://www.icir.org/bkarp/

802.11 power management

802.11 Power Management uses a simple periodic negotiation between a base station and mobile node. The mobile host and base station agree to a beacon period (in multiples of 100ms). The base station beacon buffers packets for the mobile host and uses the beacon to inform the mobile host of packets awaiting delivery. An interesting project would to evaluate how well this works (power savings and transfer performance impact) with typical 802.11 interface cards and typical workloads. The project could also evaluate alternate techniques that may work better. For example, a node could use past network patterns to predict future traffic more accurately.

Ad hoc w/ HW link evolution

Because of the nature of the devices involved and the design of radio systems, it is likely that many different radio technologies will need to co-exist in this environment. Current routing techniques focus on mobility and geographic range. However, routing in this new environment will need to optimize for bridging, spectrum usage and power consumption.

Network Measurement

Tool to determine access bandwidth of link

A key issue with emerging peer-to-peer applications is that a large number of peers use bandwidth constrained technologies such as DSL, cable modem, etc. to access the Internet. The last mile limitation has implications for p2p applications. For example, in Kazaa-like peer selection, a client must try to avoid picking a peer behind a DSL link. In End System Multicast, a client must avoid choosing a host behind a DSL machine as parent. Further, a host behind DSL cannot achieve a throughput from the source better than its access link capacity.

The question then is can we build a light-weight low-overhead tool that can automatically determine the upstream and downstream capacity of the access link?

In order to build the tool, we suggest taking the following steps:

1. Are existing bottleneck bandwidth measurement tools good enough? What are the accuracy and the overhead of existing tools? Evaluate existing bottleneck bandwidth measurement tools, such as Nettimer and Sprobe, by collecting measurements to Internet "peers."
Nettimer: http://mosquitonet.stanford.edu/~laik/projects/nettimer/index.html
Sprobe: http://sprobe.cs.washington.edu/

2. What is the significance of the "access link" bandwidth? From your measurements, what is the role of the bandwidth on the "access link" in determining the bottleneck bandwidth of a path? For example, if most Internet "peers" have DSL access, then it is likely that the access link is the bottleneck of the path.

3. How should one measure "access link" bandwidth? Are existing bottleneck bandwidth tools are too expensive, and can we build a light-weight low-overhead tool? Perhaps simple heuristics will work.

4. What is the effect of asymmetric paths (or asymmetric bandwidth) on the accuracy of the measurements?

Contact: Sanjay Rao, (sanjay@cs.cmu.edu), Eugene Ng (eugeneng@cs.cmu.edu), Yang-hua Chu (yhchu@cs.cmu.edu) and Kay Sripanidkulchai (kunwadee+@cs.cmu.edu)

Link capacity & buffering

Build a tool to measure the capacity & buffering of a bottleneck link. Use this tool to evaluate a number of deployed links with the network. Unlike the access bandwidth project, the objective of this tool is for offline evaluation. Therefore, the network can be much more heavyweight. However, this tool should not be limited to access links. You may want to look at tools like Sting to help provide endpoints for such probing.

Topology of networks

The early efforts have primarily concentrated on AS-AS connectivity. One possibility is that there are a multiple "types" of networks. E.g. physically constrained networks - networks where physical issues such as distance or physical fanout constrain the possible connections; logically constrained networks - connections can be made to arbitrary peers however, connections are often limited by "friendships"/knowledge of peers. An interesting project would search for different types of networks and try to identify key characteristics.

Characteristics other than connectivity

In order to use topologies to drive protocol design, other characteristics (such as link bandwidth and latency) must be characterized as well. It seems likely that these characteristics may have interesting properties as well (e.g. nodes with high fanout may have high bandwidth as well). An interesting project would be to measure these characteristics and identifying any interesting properties.

Peer-to-Peer

Siena publish-subscribe system

There have been two basic approaches to distributed publish-subscribe. Systems like Scribe and Herald basically use distributed hash tables to create rendezvous points for subjects. Siena uses a more routing style approach that allows a richer query language but relies on similarity of interests to help scalability. However, it is unclear how scalable Siena is on realistic network topologies with realistic workloads.

Abstract path model

In order to develop a robust distributed application, developers need effective testing and simulation tools. The current approach to improving simulation tools has been to more accurately reproduce the network. Current work has concentrated on providing realistic network topologies and scaling simulation to large numbers of nodes. Future work is likely to examine other aspects of network topology (BW and latency) as well as network traffic loads. An alternate approach to improving realism may be to create the equivalent of "Impulse Response" for a network path. Based on some basic measurements of the path it may be possible to reproduce the response of the path to arbitrary cross traffic.

Congestion control in overlay multicast

What support for congestion control should the shared infrastructure provide? Congestion control is tightly linked to monitoring losses on the Internet. Monitoring losses in multicast groups results in the problem of ACK implosion. Therefore, having the infrastructure deal with some of the congestion control may provide a more scalable solution.

Performance of Content Distribution Networks (CDNs)

Content distribution networks (CDNs) such as Akamai are becoming increasingly popular and it seems likely that most popular sites will end up being transmitted over CDNs. Your task will be to come up with benchmarks and evaluate the performance of Akamai's network. In addition to understanding the details of how their Web CDN works, you should be able to analyze its performance under different workloads. The best way to start would be to get several accounts at different ISPs (ASes) and go from there. There's one study from Sightpath by Johnson et al. on this, which appeared in the WWW caching workshop. You should become familiar with this work.

Miscellaneous

Hardware reconfiguration
CAMs and ASICs have had a significant impact on the design/architecture of modern routers. Can we effectively use of reconfigurable hardware in routers for interconnect or route lookup? Are the current forms of reconfigurable hardware well suited to this task?

Service discovery with dynamic attributes
One of the desirable features of service discovery is to be able to do things like print to the printer with the shortest queue. Service attributes for the printer such as location and memory do not change over time. However, attributes such as queue length are dynamic. Service descriptions tend to very simple in most systems. What if we made them into code segments that would be evaluated by either the service discovery server or client? This would enable service discovery servers to pull information about service attributes on demand. Pulling information would make more sense for something like a compute server where the load attribute changes frequently but few queries either match the server's other attributes. It may also be useful to support attributes such as price which may vary based on a variety of parameters including client characteristics. Issues to resolve include: How is the pull code defined (Java maybe)?, What are the security restrictions on this code, are these pulled values cached?, Is the caching defined by the code? If so, what type of persistent state is allowed across invocations?

Incentives in non-administrative hierarchies
Current service discovery hierarchies primarily only supports administratively scoped queries. It needs to support geographically scoped queries for discovering resources in mobile environments and support network performance based queries for discovering rentable network resources. However, nodes in a service discovery system must have also had the incentive to provide accurate responses to queries. In an administratively organized system, this incentive structure is simple. However, in alternate organizations, this incentive is unclear. For example, suppose the Mellon Bank ran the server for the CMU area, a search for ATM's (a service) probably will not return any non-Mellon ATM's. How do we ensure that service location servers return truthful results?

Protocol evolution
End-to-end protocols are deployed rapidly today (e.g. SACK was effectively deployed in 3 years) because end-hosts are frequently replaced or upgraded. In a future environment with many more embedded devices, it is less likely that devices will be updated. The architecture of end-node protocols stacks have to be modified to enable automated upgrading and evolution. Unfortunately, because end nodes may be extremely simple the solution of active networks may not be as applicable (not every node will be able to support the active programming environment). Perhaps some negotiation techniques for features may provide some middle ground?

802.11 PCF
It is unclear how widely deployed the 802.11 PCF mode is deployed. Would its use significantly change the behavior of networks compared to the DCF that is widely used.

Community networks
The deployment of network connections and 802.11 access points is probably relatively ubiquitous in many communities. Can we effectively use this to provide seamless connectivity throughout the community?

Assume that you're building a community network from scratch. Some number of nodes will have links to the Internet (call these gateways), and all the other nodes will need to connect with the gateways for wide-area connectivity. Ideally, I'd like to be able to just plug in a box at one of the client nodes, and have the network configure itself. (Note that, depending on the network topology, this might not be a purely local change. It might be that the addition of the new client provides a better path for existing clients.)

The question is: how do we do this, while meeting the goals of providing a fat pipe to all users, and making the connectivity reliable? Specific questions:

- How do you handle channel allocation?
- What kind of antennas should you use? Some anecdotal evidence suggests that in some cases you want to use directional antennas to reduce the amount of noise you hear from other sources.

- How do you provide multihop connectivity? The OpenAP project is using the spanning tree protocol to provide a single virtual Ethernet, and may run an ad-hoc routing protcol on top of that.
- Is this a good idea? Spanning Tree generates a somewhat arbitrary tree in that it chooses between multiple links by node ID. Would it be better to build a tree that is geographically aware (thus minimizing interference)?

It might also be possible to improve the performance of the ad-hoc routing protocols. Most of them assume a fairly dynamic network (due to mobility). The topology of a community network should be much less dynamic.

One possibility would be to evaluate existing proposals using realistic node geography and workload models. In addition to OpenAP, someone interested in pursuing this might look at the Seattle Wireless project site for good information. There's also a map server around somewhere with data that people have contributed fromt their war-driving trips. That might provide some useful data for geography models.

Web

Dynamic page content

Currently, there is very little information publicly available about the composition of webpages of a really dynamic site (say, amazon.com). It is hypothesized that dynamically generated pages, and personalized web-pages are not all that dynamic. They consist of components, that can be re-used across versions of the same page. Thus, component caching can be used to remarkably improve cache hit ratios. But, a client (or, a proxy) cannot easily identify these components, because it can only see the final output, and not the source code. This project is about inferring how much spatial (are there common components between any two pages on a single site) and temporal reuse (how frequently components on a URL change) exists in present-day dynamic web-sites by only analyzing web-pages available to different clients. A good starting point would be http://www.cs.nyu.edu/~weisong/papers/globecom02.pdf

Where does the time go?

Many researchers believe that for dynamic web-sites, most of their time is spent querying the database (input/output), and very little time is spent performing actual computation, thus, strengthening the case for a centralized 3-tier web-service architecture. Your task would be to carry out an extensive profiling, and present a detailed breakdown of where the time is spent in a traditional 3-tier architecture. Besides, how does the network bandwidth, disk bandwidth and processing requirement scale with the number of users? Since it is difficult to obtain the scripts and code for a dynamic web site, a good starting point would be TPC-W, a benchmark that models the behavior of users purchasing from an online book-seller.

Browsing behavior

How do users typically behave when browsing a dynamically generated web-site? TPC-W is a benchmark that tries to model the behavior of users purchasing from an online book-shop. A good project would be to devise a benchmark similar to TPC-W that models some other common web-activity (for example, reading mails).

Last-resort department

If none of the above projects interests you and you aren't able to come up with one on your own, here are a couple of suggestions to get you started in finding a project that does.

Find some problem/area in networking that you like (or better still, that truly excites you, and/or that seems of immediate and important relevance to you). Then address the following questions:

- Has a solution been posed to this problem in the past and, if so, what do you think of it?
- If a solution has been proposed, has it been simulated properly?
- If the system has been simulated, does anyone understand its mathematical properties?
- If the system has been simulated and theoretically modeled, has it been implemented and its performance studied under realistic conditions?
- Is the solution scalable? Is it secure? How well does it hold up in the face of failures? Will it work in heterogeneous environments? Will it stand up to coming advances in technology?
- If several solutions have been proposed, has anyone performed a comparative study of them? Why do some schemes work better than others? Can we characterize the conditions under which some schemes work better than others?