Project Ideas – 15-744 F04

TCP/TCP Routers

Multi-path routing for a single TCP flow
Contact: Brad Karp (bkarp@cs.cmu.edu)

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?

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
Contact: Brad Karp (bkarp@cs.cmu.edu)

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.

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

XCP: What does it really buy us? A Comparison with CSFQ
XCP is a recent proposal to modify routers and end-point congestion control that has received much attention. To me, this system looks like fair queuing (specifically CSFQ) with flow control (fair share rates sent to the senders to inform them to slow down). There are some important additional differences: 1) XCP informs routers of the RTT of a connection (this allows routers to compute RTT-fair rates instead of max-min fair share rates) and 2) XCP uses a TCP-based AIMD controller instead of CSFQ's fair share estimation.

In this project, you are to determine and explain what properties of XCP give it better performance than CSFQ. For example: Does explicit flow control provide better performance for protocols like TCP (which does congestion control)? Does XCP's fair share estimator work do a better job of tracking available bandwidth than CSFQ's scheme?

Background material:
- Core-Stateless Fair Queueing. Ion Stoica, Scott Shenker and Hui Zhang. ACM SIGCOMM 1998. Available at { http://www-2.cs.cmu.edu/~istoica/csfq/}
- Congestion Control for High Bandwidth-Delay Product Networks. Dina Katabi, Mark Handley, and Chalrie Rohrs. ACM SIGCOMM 2002.Available at { http://ana.lcs.mit.edu/dina/XCP/}

Mobile/Wireless

Mobile Networking Experiments using a Signal Propagation Emulator

contact: Peter Steenkiste (prs@cs.cmu.edu)

Wireless networking research faces a fundamental tension between experimental realism on one hand and repeatability on the other. Experiments with real hardware are realistic, but difficult to conduct and repeat. Simulations are controllable, but may be unrealistic. An attractive middle ground between this two approaches is to use a physical layer emulator that allows wireless experimentation with real hardware in a controllable environment. This wireless emulator allows emulation of effects such as mobility and signal propagation by digitally manipulating the wireless signals. This emulator may serve as the basis for many interesting projects that vary depending on the skills of the project members. The following project is likely to be accessible by most students. This project involves using a prototype wireless emulator to analyze the use of 802.11 as the basis for inter-vehicular and vehicle to access point communication. Most major automobile manufacturers are actively investigating the introduction of 802.11 for this purpose, and it would be interesting to characterize the network performance as well as to look at applications of the technology. This will likely involve the study of both traditional topics such as handoff time between access points as well as less traditional topics such as inter-vehicular communication and applications of this technology.

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. We have a variety of war-driving data sources that may be of help in this project.

Network Measurement

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.

Router Configuration

contacts: Hui Zhang and Dave Maltz(dmaltz@cs.cmu.edu)}

This project will involve analyzing router configuration files of a large ISP (AT&T) to answer one or more interesting questions.

Here's a quick from Jennifer Rexford (ATT Research) about Router configuration data that AT&T is willing to provide for this project: "We could produce a stripped-down version of the configs that are (i) anonymized, (ii) condensed (removing anything not related to routing and addressing), and (iii) perhaps "bug free" (by removing "dangling" routing sessions, etc. for the sake of simplicity). This would provide great anonymity and would also make it easier for the students to understand the data. We could also make the data more "parseable" by putting clear markers and consistent indentation at the end of "sections" of the config".

This project will essentially involve analyzing such configuration data to answer one or more of the following questions:

- Data mining to look for patterns (as discussed in our Rexford et. al's EDGE paper), to aid in detecting inconsistencies, concisely summarizing the configurations, and designing templates for automated configuration
- Do a reachability analysis (composing routing configuration at different routers to produce a "reachability matrix")
- Routing table size estimation (computing the "worst case" possible routing and/or forwarding table size for a given network, given the configuration of the routing protocols and the filters/policies)
- Route aggregation analysis (automatically identifying missed opportunities for route aggregation)
- Reliability analysis (asking whether the IP network, the iBGP session graph, etc. remains connected under various failure modes)

Network Games

Contact: Ashwin Bharambe

While networked games are becoming more popular, little is known about their behavior/requirements. The goal of this project would be to measure and model real player workloads using Quake2, Quake 3 and other-games? This would require setting up the required modified servers, informing the master servers so that people play as well as gathering and analyzing the statistics. Some possibilities include characterizing gameplay depending on game-type, map-type, etc.

Network Positioning Systems

Network simulators for those who don't care about the network}

Simulators like NS-2 require users to specify the entire topology of a network. Such simulators, accurately reproduce the routing of packets through the network and the sharing of links, bandwidth and queues at each router. In addition, several research efforts are exploring how to developing network topology generators that produce realistic network graphs.

The cost of accurate simulation is scale. The NS-2 simulator has great difficulty in simulating networks with more than about 1000 nodes. However, many experimenters, such as those evaluating P2P protocols, are not so interested in the detailed behavior of the internal parts of the network. Typically, such systems are simulated using very simple tools that do not reproduce any realistic network delays.

In this project, you are to explore the use of GNP (Global Network Positioning) to help create more scalable, somewhat-realistic network simulations. The basic idea is to use GNP to estimate the end-to-end transmission delays for any message instead of actually simulating each hop in the network interior. There are many steps to achieving this goal and there are variety of different projects that are possible. Some possibilities:

- **GNP Coord generator**. Topology generators that produce realistic looking networks (some with link delays) exist for detailed simulators. Develop an equivalent for GNP-based simulators. This could either require data analysis to identify the underlying patterns in GNP coordinates (much like power-laws to network topology), theoretical analysis to map power-laws directly to the equivalent GNP patterns or clever implementation to generate GNP coordinates directly (and efficiently) from a given graph.
- **Construction/evaluation of GNP simulator**. How much more scalable is this approach? What are the costs in NS-2 that make it slow? While GNP simulates only latency, can we employ some simple hybrid simulator that allows bandwidth simulation as well? For example, we could create a hybrid simulator which uses GNP along with a detailed simulation of first-hop bottlenecks.

Network Positioning

contact:Eugene Ng (eugeneng@cs.rice.edu)

Recently, a concept called network positioning has been proposed. The basic idea is to take Internet round-trip delays among hosts as inputs and generate artificial network positions for hosts in some metric space. The basic challenge is to do this efficiently and accurately, so that network positions can be used to compute unknown network delays. Network positions can directly benefit applications, and can potentially benefit other aspects such as network routing. Several studies have confirmed that it is feasible to compute accurate network positions in the real Internet, but many challenges must be overcome before we can fully understand this approach and build a real network positioning system for the Internet.

Note: A code base for basic networking positioning is available as a starting point for these projects.
- In a simple network positioning system design, fixed dedicated special nodes called Landmarks are used to provide stability to the computed positions. In practice, having fixed dedicated Landmarks both makes management of the system a nightmare, and could limit the accuracy of the system. If there are no fixed dedicated Landmarks, it is not clear what are the implications to stability and robustness. In this project, you will explore methods to reduce a network positioning system's reliance on fixed dedicated Landmarks as much as possible while achieving the best accuracy and stability. Ideally, the system will have no dedicated Landmarks, be highly robust to node failures, accurate, and stable.
- In empirical studies, it has been noted that the dimensionality required to accurately embed Internet delays isn't very high (~5 dimensions). However, so far there is little understanding of what property of the Internet contributes to this behavior. In this project, you will study various types of Internet topology models as well as measured delay data to explore the emergence of dimensionality of Internet delay, and potentially answer how dimensionality may change as the Internet evolves.
- Network position is not 100% accurate. In some cases, it disagrees with the actual measured delay by a significant amount. The question is, is there any useful information provided by such mismatches? In this project, you will explore the application of network position in routing (e.g. perhaps in the context of End System Multicast), and study how network position may help assess the optimality of a route and guide the routing decisions.

Some references:

- T. S. Eugene Ng and Hui Zhang, "Predicting Internet Network Distance with Coordinates-Based Approaches", INFOCOM'02, New York, NY, June 2002
- Russ Cox, Frank Dabek, Frans Kaashoek, Jinyang Li, Robert Morris, "Practical, Distributed Network Coordinates", HotNets-II, Boston, MA, Oct 2003

Game/Graph Theory

Macroscopic Internet Design: Scaling of Congestion in the Network Core
contact: Aditya Akella (aditya@cs.cmu.edu)

As the Internet grows in size, it becomes crucial to understand how the speeds of links in the network must improve in order to sustain the pressure of new end-nodes being added each day. Although the speeds of links in the core and at the edges improve roughly according to Moore's law, this improvement alone might not be enough. Indeed, the structure of the Internet graph and routing in the network might necessitate much faster improvements in the speeds of key links in the network.

Past work by Akella et. al. has shown that the worst congestion in the Internet in fact scales poorly with the network size ($n^{1+Omega(1)}$, where n is the number of nodes), when shortest-path routing is used. This paper also shows that policy-based routing does not exacerbate the maximum congestion when compared to shortest-path routing. The second part of the paper is devoted to identifying ways to alleviate this congestion to avoid some links from being perpetually congested. However, this paper only considers one such mechanism: introducing moderate amounts of redundancy in the graph in terms of the edges between pairs of nodes. Also, the graph generation model considered in the paper only holds at an AS level and abstracts potentially many parallel peering links between adjacent ASes by a single link.

The project is aimed at extending this work on the scaling of congestion by addressing the following issues:
- How much parallelism actually exists in the Internet today? In other words, what is hidden "under" the abstraction of peering links into a single edge, described above?
- What other ways of modifying the Internet graph can alleviate congestion from the core of the network? Candidate mechanisms to test out may include adding random links vs adding links in an informed manner.

The first question involves measurement in the wide-area Internet. Aditya Akella (aditya@cs.cmu.edu) will be happy to provide guidance on the data collection (possibly in collaboration with Bruce Maggs and Akamai Technologies).

The second question will involve simulations over and existing code base (simulator written by Arvind Kannan and Aditya Akella). If students are interested, there is a fair bit of theory involved too (some mild graph theory and combinatorics may be involved. Read references below for info). Students involved in the project are expected to address at least one of these two questions.

Here are some useful reading and background pointers:
- Scaling Properties of the Internet Graph} Aditya Akella, Shuchi Chawla, Arvind Kannan and Srinivasan Seshan. ACM Principles of Distributed Computing (PODC) 2003. Available at { http://www-2.cs.cmu.edu/~aditya/publications.html}
- Conductance and Congestion in Power Law Graphs, C. Gkantsidis, M. Mihail and A. Saberi, ACM SIGMETRICS 2003. Available at {http://www.cc.gatech.edu/fac/Milena.Mihail}

Realistic Models for Selfish Routing in the Internet
contact: Shuchi Chawla and Aditya Akella

The Internet of today is a large collection of independently administered autonomous entities. These entities typically take unilateral decisions, such as selecting a path to route their packets, in order to maximize their own utility. Our goal is to study the effect of selfish routing on the efficient performance of the Internet.

Several studies have tried to understand the impact of entities in the Internet routing traffic in a selfish manner, by casting this question as a game between selfish agents and have tried to quantify what is called the price of anarchy – the ratio of performance of the network at the Nash Equilibrium of the game, to the optimal performance. In a series of papers, Roughgarden et al study the cost of Nash equilibrium, when selfish flows in a network pick routes so as to minimize their latencies. They show that the cost of a Nash equilibrium can, in general, be arbitrarily larger than the cost of the optimal routing, except when the latencies on links in the network depend linearly on the total flow on them.

While seminal in their contributions, the aforementioned studies have a common drawback: they are too simplistic in their modeling of the behavior of flows on the Internet. As a result the applicability of these results to the current Internet is approximate at best.

This project will involve extending these models in several ways and bring them closer to reality. Here are some proposed extensions:

- TCP is the predominant model of transport in today's Internet used by over 80% of the bytes carried by the network. TCP uses a congestion control algorithm to help adapt its rate of flow to the bandwidth available on its path. Is it possible to model selfish TCP flows? How well does the network perform at Nash Equilibrium in this new model?

- In the wide area Internet, realistic selfish entities may not have arbitrary control in the end to end paths they choose, mainly because of policies between ISPs and the lack of support for source routing in the Internet today. How do the past results for the Nash equilibria of the routing game change in a new setting where the routes of the flows are somehow restricted?

- What effect does the topological structure of the Internet (heavy-tailed degree distribution) have on the nature of these equilibria?

- This project will involve basic game theoretic analysis and, mostly, simulation based work. The simulations will likely use the same code base as with the project above (Scaling of congestion). The basic approach would be to use simulations and existing simulations and LP-solvers to compute the Nash Equilibria for at least one of the above questions. Again, interested students are welcome to analyze the problem and the simulation results from a game-theoretic perspective.

Some related past work and useful background material:

- How bad is selfish routing? T. Roughgarden and E. Tardos. FOCS 2001. Available at {http://www.cs.cornell.edu/timr/papers/routing.pdf}
- On Selfish Routing in Internet-Like Environments Lili Qiu, Yang Richard Yang, Yin Zhang and Scott Shenker. ACM SIGCOMM 2003. Available at {http://www.acm.org/sigcomm/sigcomm2003/papers.html#p151-qiu}
- Realistic models for selfish routing in the Internet, Aditya Akella and Shuchi Chawla. Preprint.Available at {http://www.cs.cmu.edu/~shuchi/papers/selfishrouting.ps}

More Selfishness: Analyzing Selfish Congestion Control}
contact: Aditya Akella (aditya@cs.cmu.edu)}

For years, the conventional wisdom has been that the continued stability of the Internet depends on the widespread deployment of "socially responsible" congestion control. In a past paper, Akella et. al. try to answer the following question: If network end-points behaved in a selfish manner, trying to maximize their observed throughput, would the stability of the Internet be endangered? They analyze what is called the "TCP Game" where each flow attempts to maximize the throughput it achieves by modifying its congestion control behavior (contrast this with the previous project where selfish flows could modify their routing behavior). The paper shows the following key results, on a simple network model with single bottlenecks and equal RTT flows: In more traditional environments – where end-points use TCP Reno-style loss recovery and routers use drop-tail queues – the Nash Equilibria of the TCP game are reasonably efficient. However, when endpoints use more recent variations of TCP (e.g., SACK) and routers employ either RED or drop-tail queues, the Nash equilibria are very inefficient.

The aim of this project would be to extend these results in the following ways:

- What happens when only a small fraction of TCP flows behave in a greedy manner and the rest behave in the socially optimal manner? Does this improve the efficiency of the network at the Nash Equilibrium?
- What happens when selfish flows employ different variants of TCP when trying to behave in a greedy manner?

Evaluate these two questions in a model where flows can have very different RTTs and where they may face multiple, sometime unshared, bandwidth bottlenecks along their respective paths.

The project will primarily involve simulations in NS-2 (over some existing code written by Aditya Akella). This work may also involve some game theoretic analysis of the problem, for interested students. Students involved in this project are expected to address at least two of the above questions.

Here are the papers you should read for background:

- Selfish Behavior and Stability of the Internet: A Game-Theoretic Analysis of TCP, Aditya Akella, Richard Karp, Christos Papadimitriou, Srinivasan Seshan and Scott Shenker. ACM SIGCOMM 2002. Available at {http://www-2.cs.cmu.edu/~aditya/publications.html}

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?

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?

ESM Monitor
Contact: Aditya Ganjam

End System Multicast (ESM) is a large distributed system for efficiently distributing multicast content (i.e. video streams) from a source to a set of receivers. In ESM, receivers interested in the content forward data to other receivers. The system handles constant churn in group membership and changes in network conditions.

One of the difficulties in ESM is monitoring the performance of the system as a whole and performance of each individual receiver during the event. An ESM monitor should be able to gather information from a large set of receivers, determine statistics of the current environment (group membership over time, stay time of receivers, NAT/firewall properties of receivers, available overall resources in the system), characterize the behavior of the protocol (current distribution tree, resource usage (efficiency) of the tree) and determine receivers' performance (received bandwidth, loss, delay).

The result of this project will be a scalable monitoring system that will give the user high-level information about the running event, as well as, low-level information when needed. In addition, some of this information can be fed back into the system and can be used to invoke hosts that can contribute bandwidth when needed.

Part I:
While collecting information from all receivers in one centralized location is straightforward, this solution will not scale well with the number of receivers. The first part of this project is to design a distributed data collection infrastructure.

Part II:
The monitoring system must analyze the collected data to generate the statistics and performance information described above in a distributed fashion.

Part III:
Even with all the information described above, it is not straightforward to answer some of the most basic questions when running an event. This part of the project entails picking some of the following questions (or new ones that come up) and developing robust heuristics to answer them.
- Are receivers getting acceptable performance?
- What is the problem for receivers that are not getting acceptable performance? Are they physically constrained? Is there not enough upstream bandwidth? Is there too much churn is the group membership? Is network congestion bad?

- Is there a particular region of the world where receivers get poor performance?
- Is the delivery tree efficient? Could it be better? Where are the inefficiencies?
- Can extra hosts that can contribute more upstream bandwidth help the system?

ESM Security
Contact: Jibin Zhang

End System Multicast (ESM) is a peer-to-peer based live (audio/video) streaming application. Comparing to the traditional streaming server based solutions, P2P based systems can utilize end users' resources, such as bandwidth, hence are very attractive especially for high bandwidth video streaming. But they also face some unique challenges. One of the big challenges is security.

The big goal of this project is try to establish some framework of how to address security related issues for P2P based applications in general, or just P2P based video/audio streaming applications in particular. The framework can provide the basic guidelines of what are the things the P2P based applications need to do so that P2P based applications can provide services as trustful as those centralized servers provide. We can first see whether we can define a security model for P2P based systems in general, or P2P based streaming applications in particular, so that we can compare the security properties of P2P based systems with traditional central servers based solutions. This may be similar to what WEP is for wireless networks comparing to wire based networks. With that model, we might be able show to users how and why the P2P based system may be secure enough for some of their applications. As an example, it is not hard for us to show that we can guarantee content authenticity in P2P based system and we can have a relative simple model to argue that the content end users received (through other users) is actually what the source sends out with very high confidence. Can we establish similar property for: access control, digital right management, privacy protection, billing, dos attack protection, reputation (or any other sort of ranking) of the peers, etc? You may have some other areas, and you can just focus one or a few of those areas. For example, just access control and billing, and focus on some practical implementations, such as interaction with directory services and payment systems, etc.

Note that you can make some reasonable assumptions about the systems if this may help simplify the problem. For example, you can assume the code of ESM applications can not be tempered easily, (but they may be able to change any dynamically linked libraries used by ESM), the control protocols used in ESM are not known and the control traffic are encrypted (what are the reasonable assumptions and approaches to do that?). But the assumptions should be reasonable and justified, for example you can not assume people will not be able to identify the network ports used by the system, you can not assume people don't have root access, or they can not change the kernel of the operating systems, etc.

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?