Preliminary Technical Program

Session 1: Multicast and Anycast
Wednesday, Aug 30, 10.45-12.15

1-1 A Framework for Scalable Global IP-Anycast (GIA). *Dina Katabi*, *John Wroclawski* (Massachusetts Institute of Technology)

1-2 pgmcc: A TCP-friendly Single-Rate Multicast Congestion Control Scheme. *Luigi Rizzo* (Universita di Pisa)

1-3 Fault Isolation in Multicast Trees. *Anoop Reddy*, *Ramesh Govindan*, *Deborah Estrin* (USC/Information Sciences Institute)

Session 2: Control Mechanisms
Wednesday, Aug 30, 13.30-15.00

2-1 Equation-Based Congestion Control for Unicast Applications. *Sally Floyd*, *Mark Handley* (AT&T Center for Internet Research at ICSI), *Jitendra Padhye* (University of Massachusetts, Amherst), *Joerg Widmer* (AT&T Center for Internet Research at ICSI)

2-2 Endpoint Admission Control: Architectural Issues and Performance. *Lee Breslau* (AT&T Research), *Edward Knightly* (Rice University), *Scott Shenker* (AT&T Center for Internet Research at ICSI), *Ion Stoica*, *Hui Zhang* (CMU)

2-3 Decoupling QoS Control from Core Routers: A Novel Bandwidth Broker Architecture for Scalable Support of Guaranteed Services. *Zhi-Li Zhang*, *Zhenhai Duan* (University of Minnesota), *Lixin Gao* (Smith College), *Yiwei Thomas Hou* (Fujitsu Labs)

Session 3: World Wide Web
Wednesday, Aug 30, 15.30-17.00

3-1 A Content-based Technique for Eliminating Redundant Web Traffic. *Neil T. Spring*, *David Wetherall* (University of Washington)

3-2 On Network-Aware Clustering of Web Clients. *Balachander Krishnamurthy* (AT&T Research), *Jia Wang* (Cornell University)

3-3 The Content and Access Dynamics of a Busy Web Site: Findings and Implications. *Venkata N. Padmanabhan* (Microsoft Research), *Lili Qiu* (Cornell University)

Session 4: Performance Analysis and Modeling
Thursday, Aug 31, 9.00-10.30

4-1 Critical Path Analysis of TCP Transactions. *Paul Barford*, *Mark Crovella* (Boston University)

4-2 Tuning RED for Web Traffic. *Mikkel Christiansen*, *Kevin Jeffay*, *David Ott*, *F. Donelson Smith* (University of North Carolina at Chapel Hill)

4-3 A Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED. *Vishal Misra*, *Wei-Bo Gong*, *Don Towsley* (University of Massachusetts, Amherst)

Session 5: Routing Stability and Convergence
Thursday, Aug 31, 11.00-12.00

5-1 Routing Stability in Congested Networks: Experimentation and Analysis. *Aman Shaikh* (University of California, Santa Cruz), *Lampros Kalampoukas*, *Rohit Dube* (Bell Labs, Lucent Technologies), *Anujan Varma* (University of California, Santa Cruz)

5-2 An Experimental Study of Delayed Internet Routing Convergence. *Craig Labovitz* (Microsoft Research), *Abha Ahuja*, *Abhijit Abose*, *Farnam Jahanian* (University of Michigan)

Session 6: Routing and Bridging
Thursday, Aug 31, 13.30-14.30

6-1 FIRE: Flexible Intra-AS Routing Environment. *Craig Partridge* (BBN Technologies), *Alex C. Snoeren* (BBN Technologies & Massachusetts Institute of Technology), *Tim Strayer*, *Beverly Schwartz*, *Matthew Condell*, *Isidro Castineyra* (BBN Technologies)

6-2 SmartBridge: A Scalable Bridge Architecture. *Thomas L. Rodeheffer*, *Chandramohan A. Thekkath* (Compaq Systems Research Center), *Darrell Anderson* (Duke University)

Session 7: TCP Analysis
Thursday, Aug 31, 15.30-17.00

7-1 TCP is Max-Plus Linear. *Francois Baccelli*, *Dohy Hong* (ENS-INRIA)

7-2 A Stochastic Model of TCP/IP with Stationary Random Losses. *Eitan Altman*, *Kostia Avrachenkov*, *Chadi Barakat* (INRIA)

7-3 On the Propagation of Long-range Dependency in the Internet. *Andras Veres* (Ericsson), *Zsolt Kenesi*, *Sandor Molnar* (Technical University of Budapest), *Gabor Vattay* (Eotvos University, Budapest)

Session 8: Tracing and Measurement
Friday, Sep 1, 9.00-11.00

8-1 Deriving Traffic Demands for Operational IP Networks: Methodology and Experience. *Anja Feldmann* (University of Saarbruecken), *Albert Greenberg*, *Carsten Lund*, *Nick Reingold*, *Jennifer Rexford*, *Fred True* (AT&T Research)

8-2 Trajectory Sampling for Traffic Measurement. *Nick Duffield*, *Matthias Grossglauser* (AT&T Research)

8-3 Measuring Link Bandwidths Using a Deterministic Model of Packet Delay. *Kevin Lai*, *Mary Baker* (Stanford University)

8-4 Practical Network Support For IP Traceback. *Stefan Savage*, *David Wetherall*, *Anna Karlin*, *Tom Anderson* (University of Washington)

Session 9: Header Processing
Friday, Sep 1, 11.30-13.00

9-1 When the CRC and TCP Checksum Disagree. *Jonathan Stone* (Stanford University), *Craig Partridge* (BBN Technologies)

9-2 Packet Types: Abstract Specification of Network Protocol Messages. *Peter J. McCann*, *Satish Chandra* (Bell Labs, Lucent Technologies)

9-3 Terabit IP Lookups. *Sandeep Sikka* (Inktomi), *George Varghese* (UCSD)

Abstracts

1-1 A Framework for Scalable Global IP-Anycast (GIA) (Dina Katabi, John Wroclawski): This paper proposes GIA, a scalable architecture for global IP-anycast. Existing designs for providing IP-anycast must either globally distribute routes to individual anycast groups, or confine each anycast group to a preconfigured topological region. The first approach does not scale because of excessive growth in the routing tables, whereas the second one severely limits the utility of the service. Our design scales by dividing inter-domain anycast routing into two components. The first component builds inexpensive default anycast routes that consume no bandwidth or storage space beyond that used by the underlying unicast routing. The second component, controlled by the edge domains, generates enhanced anycast routes that are customized according to the beneficiary domain's interests. We evaluate the performance of our design using simulation, and prove its practicality by implementing it in the Multithreaded Routing Toolkit.

1-2 pgmcc: A TCP-friendly Single-Rate Multicast Congestion Control Scheme (Luigi Rizzo): We present a single rate multicast congestion control scheme which is TCP-friendly and achieves scalability, stability and fast response to variations in network conditions. Our scheme is suitable for both non-reliable and reliable data transfers, and uses a window-based TCP-like controller based on positive ACKs and run between the sender and a group's representative, the *acker*. The innovative part of our scheme is a fast and low-overhead procedure to select (and track changes of) the acker, which permits us to consider the acker as a *moving* receiver rather than a *changing* one. As such, the scheme is robust to measurement errors, and support fast response to changes in the receiver set and/or network conditions. The scheme has been implemented in the PGM protocol, and the paper presents a number of experimental results on its performance.

1-3 Fault Isolation in Multicast Trees (Anoop Reddy, Ramesh Govindan, Deborah Estrin): Fault isolation has received little attention in the Internet research literature. We take a first step towards addressing this deficiency, exploring robust and scalable mechanisms by which multicast receivers can (in some cases, approximately) locate the on-tree router responsible for a route change, or the link responsible for significant packet loss. These mechanisms rely on receivers with overlapped paths to the source sharing the responsibility of monitoring their overlapped path segments. Our mechanisms assume no additional path monitoring capability other than that provided by *multicast traceroute* (mtrace). We explore the tradeoff between monitoring overhead and fault isolation error for two classes of mechanisms: those that assume some kind of router assist for selectively multicasting the responses to mtrace requests, and those that do not. In the former category fall schemes that use *subcast* or *di-*

rected multicast. In the latter are schemes that use *scoping*, or a limited number of multicasts. The latter two approaches are deployable in today's multicast infrastructure. Our evaluations reveal that while some deployable alternatives have acceptable overhead, schemes that employ router assist have very desirable scaling characteristics.

2-1 Equation-Based Congestion Control for Unicast Applications (Sally Floyd, Mark Handley, Jitendra Padhye, Joerg Widmer): This paper proposes a mechanism for equation-based congestion control for unicast traffic in the Internet. Most best-effort traffic in the current Internet is well-served by the dominant transport protocol TCP. However, some unicast traffic could find use for a TCP-friendly congestion control mechanism that refrains from reducing the sending rate in half in response to a single packet drop. Instead of decreasing the sending rate in response to each packet drop, with our mechanism the sender explicitly adjusts its sending rate as a function of the measured packet drop rate. We use both simulations and experiments over the Internet to explore performance. Equation-based congestion control is also a promising avenue of development for congestion control of multicast traffic, and so an additional reason for this work is to lay a sound basis for the later development of multicast congestion control. In this paper we briefly discuss the lessons of this work for multicast congestion control.

2-2 Endpoint Admission Control: Architectural Issues and Performance (Lee Breslau, Edward Knightly, Scott Shenker, Ion Stoica, Hui Zhang): The traditional approach to implementing admission control, as exemplified by the Integrated Services proposals in the IETF, uses a signaling protocol to establish reservations at all routers along the path. While providing excellent quality-of-service, this approach has limited scalability because it requires routers to keep per-flow state and to process per-flow reservation messages. In an attempt to implement admission control without these scalability problems, several recent papers have proposed various forms of endpoint admission control. In these designs, the hosts (the endpoints) probe the network to detect the level of congestion; the host admits the flow only if the level of congestion is sufficiently low. This paper is devoted to the study of such algorithms. We first consider several architectural issues that guide (and constrain) the design of such systems. We then use simulations to evaluate the performance of such designs in various settings.

2-3 Decoupling QoS Control from Core Routers: A Novel Bandwidth Broker Architecture for Scalable Support of Guaranteed Services (Zhi-Li Zhang, Zhenhai Duan, Lixin Gao, Yiwei Thomas Hou): We present a novel bandwidth broker architecture for scalable support of guaranteed services that decouples the QoS control plane from the packet forwarding plane. More specifically, under this architecture, *core routers do not maintain any QoS reservation states, whether per-flow or aggregate*. Instead, the QoS reservation states are stored at and managed by a bandwidth broker. There are several advantages of such a bandwidth broker architecture. Among others, it avoids the problem of inconsistent QoS states faced by the conventional hop-by-hop, distributed admission control approach. Furthermore, it allows us to design efficient admission control algorithms without incurring any overhead at core routers. The proposed bandwidth broker architecture is designed based on a *core stateless* virtual time reference system we developed earlier. This virtual time reference system provides a unifying framework to characterize, in terms of their abilities to support delay guarantees, both the *per-hop behaviors* of core routers and the *end-to-end properties* of their concatenation. In this paper we focus on the design of efficient admission control algorithms under the proposed bandwidth broker architecture. We consider both *per-flow* end-to-end guaranteed delay services and *class-based* guaranteed delay services with flow aggregation. Using our bandwidth broker architecture, we demonstrate how admission control can be done on an entire *path* basis, instead of on a "hop-by-hop" basis. Such an approach may significantly reduce the complexity of the admission control algorithms. In designing class-based admission control algorithms, we investigate the problem of flow aggregation in providing guaranteed delay services, and devise a new apparatus to effectively circumvent this problem. We conduct extensive analysis to provide theoretical underpinning for our schemes as well as to establish their correctness. Simulations are also performed to demonstrate the efficacy of our schemes.

3-1 A Content-based Technique for Eliminating Redundant Web Traffic (Neil T. Spring, David Wetherall): We present a new technique for identifying repetitive information transfers and use it to analyze the redundancy of Web traffic. Our insight is that dynamic content, streaming media and other traffic that is not cached by today's Web caches is likely to derive from similar information. We have therefore adapted existing similarity detection techniques to the problem of building a system that eliminates redundant transfers. The result generalizes other approaches such as delta-coding and duplicate suppression. Our technique is lightweight enough to run at better than T3 rates (45 Mbps), and trace analysis predicts that a system based on it can be highly effective in practice. In our traces, after Web proxy caching, an additional 40% of Web traffic is found to be redundant. Moreover, since our system makes no assumptions about HTTP protocol syntax or caching semantics, it provides immediate benefits for other types of content, such as streaming media, FTP traffic, news and mail.

3-2 On Network-Aware Clustering of Web Clients (Balachander Krishnamurthy, Jia Wang): Being able to identify the groups of clients that are responsible for a significant portion of a Web site's requests can be helpful to both the Web site and the clients. In a Web application, it is beneficial to move content closer to groups of clients that are responsible for large subsets of requests to an origin server. We introduce *clusters* - a grouping of clients that are close together topologically and likely to be under common administrative control. We identify clusters using a "network-aware" method, based on information available from BGP routing table snapshots. Experimental results show that our entirely automated approach is able to identify clusters for 99.9% of the clients in a wide variety of Web server logs. Sampled validation results show that the identified clusters meet the proposed validation tests in over 90% of the cases. An efficient self-corrective mechanism increases the applicability and accuracy of our initial approach and makes it adaptive to network dynamics. In addition to being able to detect unusual access patterns made by spiders and (suspected) proxies, our proposed method is useful for content distribution and proxy positioning, and applicable to other problems such as server replication and network management.

3-3 The Content and Access Dynamics of a Busy Web Site: Findings and Implications (Venkata N. Padmanabhan, Lili Qiu): In this paper, we study the dynamics of one of the busiest Web sites in the Internet today. Unlike many other efforts that have analyzed client accesses as seen by proxies, we focus on the server end. We analyze the dynamics of both the server content and client accesses made to the server. The former considers the content creation and modification process while the latter considers page popularity and locality in client accesses. Some of our key results are: (a) files tend to change little when they are modified, (b) a small set of files tends to get modified repeatedly, (c) file popularity follows a Zipf-like distribution with a parameter alpha that is much larger than reported in previous, proxy-based studies, and (d) there is significant temporal stability in file popularity but not much stability in the domains from which clients access the popular content. We discuss the implications of these findings for techniques such as Web caching (including cache consistency algorithms) and prefetching or server-based "push" of Web content.

4-1 Critical Path Analysis of TCP Transactions (Paul Barford, Mark Crovella): Improving performance in the Internet requires a detailed understanding of when and how delays in data transfers are introduced. Latency in data transfers between hosts can be caused by load on hosts themselves, the protocols used in the transfer of data and by the network links which connect the hosts. We describe a method for pinpointing where delays are introduced in TCP transactions which utilizes tcpdump traces taken simultaneously at both end points of a TCP transaction. This method extracts acknowledgement packets and the data packets which liberated them from the traces and constructs the critical path for a TCP transaction. The critical path enables us to assign latency to either client, server or network. We have implemented our technique in a tool called tcpeval which automates critical path analysis for Web transactions. We show that our analysis method is robust enough to analyze traces taken for two different TCP implementations (Linux and FreeBSD) and we argue that it can easily be extended to analyze data from other applications (such as FTP). We present results of critical path analysis for a set of Web transactions taken over a 15 day period under a variety of server and network conditions. These results show that for small and medium sized files, transfer latency is primarily determined by server load and for large files, transfer latency is primarily determined by network conditions.

4-2 Tuning RED for Web Traffic (Mikkel Christiansen, Kevin Jeffay, David Ott, F. Donelson Smith): The IRTF is promoting deployment of RED in Internet routers. We study the effects of RED on the performance of Web requests with a novel aspect of our work being a concentration on a user-centric measure of performance %97 response time for HTTP request-response transactions. We empirically evaluate RED across a range of control parameter settings and offered loads. Our results show that: (1) contrary to expectations, compared to a (properly configured) FIFO queue, RED has a minimal effect on HTTP response times for offered loads up to 90% of link capacity, (2) response times at loads in this range are not substantially effected by RED control parameters, (3) between 90% and 100% load, RED can be carefully tuned to yield performance somewhat superior to FIFO, however, response times are quite sensitive to the actual RED parameter values selected, and (4) in such congested networks, RED parameters that provide the best link utilization produce poorer response times. We conclude that for links dominated by web traffic, RED appears to provide no clear advantage over FIFO for end-user response times.

4-3 A Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED (Vishal Misra, Wei-Bo Gong, Don Towsley): In this paper we use jump process driven Stochastic Differential Equations to model the interactions of a set of TCP flows and RED routers in a network setting. When expected behavior as a function of time is of interest, e.g., average queue length, loss rate, we show how the SDEs can be transformed into a set of ODEs which can be easily solved numerically. Our results show excellent agreement with those of similar networks simulated using the well known ns-simulator. Our model enables us to get an in-depth understanding of the RED algorithm. Using the tools developed in this paper, we present a critical analysis of the RED algorithm. We explain the role played by the RED configuration parameters in the behavior of the algorithm in a network and present guidelines for choosing those parameters. Our technique has straightforward extensions to other active queue management algorithms. We believe this modeling/solution methodolgy has a great potential in analyzing and understanding various network congestion control algorithms.

5-1 Routing Stability in Congested Networks: Experimentation and Analysis (Aman Shaikh, Lampros Kalampoukas, Rohit Dube, Anujan Varma): Loss of the routing protocol messages due to network congestion can cause peering session failures in routers, leading to route flapping and routing instabilities. We study the effects of traffic overload on routing protocols by quantifying the stability and robustness properties of two common Internet routing protocols, OSPF and BGP, when the routing control traffic is not isolated from data traffic. We develop analytical models to quantify the effect of congestion on the robustness of OSPF and BGP as a function of the traffic overload factor, queueing delays, and packet sizes. We perform extensive measurements in an experimental network of routers to validate the analytical results. We subsequently use the analytical framework to investigate the effect of factors that are difficult to incorporate into an experimental setup, such as a wide range of link propagation delays and packet dropping policies. Our results show that increased queuing and propagation delays adversely affect BGP's resillience to congestion, in spite of its use of a reliable transport protocol. Our findings demonstrate the importance of selective treatment of routing protocol messages from other traffic, by scheduling and buffer management policies in the routers, to achieve stable and robust network operation.

5-2 An Experimental Study of Delayed Internet Routing Convergence (Craig Labovitz, Abha Ahuja, Abhijit Abose, Farnam Jahanian): This paper examines the latency in Internet path failure, fail-over and repair due to the convergence properties of inter-domain routing. Unlike switches in the public telephony network which exhibit fail-over on the order of milliseconds, we show inter-domain routers in the packet switched Internet may take several minutes to reach a consistent view of the network topology after a fault. This delay stems from the independent computation and route selection of the BGP path vector algorithm on each backbone router. During these periods of delayed convergence, end-to-end Internet paths will experience intermittent loss of connectivity, as well as increased packet loss and latency. We present a two-year study of Internet routing convergence based on the experimental instrumentation of key portions of the Internet infrastructure, including both passive data collection and fault-injection machines at major Internet exchange points. Based on data from the injection and measurement of several hundred thousand inter-domain routing faults, we describe several unexpected properties of convergence and show that the measured upper bound on Internet

6-1 FIRE: Flexible Intra-AS Routing Environment (Craig Partridge, Alex C. Snoeren, Tim Strayer, Beverly Schwartz, Matthew Condell, Isidro Castineyra): Current routing protocols are monolithic. They specify the algorithm used to construct forwarding tables, the metric used by the algorithm (generally some form of hop-count), and the protocol used to distribute these metrics (e.g., link-state or distance vector) as an integrated package. The Flexible Intra-AS Routing Environment (FIRE) is a link-state, intra-domain routing protocol that decouples these three components. FIRE supports run-time programmable algorithms and metrics over a secure link-state distribution protocol. By allowing the network operator to dynamically reprogram in Java both the metrics and routing algorithm used to construct forwarding tables, FIRE supports the development and deployment of novel routing algorithms without the need for a new protocol to distribute state. FIRE supports multiple concurrent routing algorithms and metrics, each constructing separate forwarding tables. Through the use of operator-specified packet filters, separate classes of traffic may be routed using completely different routing algorithms, all supported by a single routing protocol. This paper presents an overview of FIRE, focusing particularly on the novel aspects of FIRE with respect to traditional routing protocols. We also briefly describe the Java programming interface and discuss our implementation experience.

6-2 SmartBridge: A Scalable Bridge Architecture (Thomas L. Rodeheffer, Chandramohan A. Thekkath, Darrell Anderson): As the number of hosts attached to a network increases beyond what can be connected by a single local area network (LAN), forwarding packets between hosts on different LANs becomes an issue. Two common solutions to the forwarding problem are IP routing and spanning-tree bridging. IP routing scales well, but imposes the administrative burden of managing subnets and assigning addresses. Spanning-tree bridging, in contrast, requires no administration, but often does not perform well in a large network, because too much traffic must funnel through the root of the spanning tree, creating a bottleneck. This paper introduces a new architecture, called SmartBridge, that combines the good features of IP routing and spanning-tree bridging. We have implemented the SmartBridge design for 10 Mb/s and 100 Mb/s Ethernet LANs, using standard PC hardware with off-the-shelf network interface cards and running our algorithms in software. Our 100 Mb/s system runs at full link bandwidth.

7-1 TCP is Max-Plus Linear (Francois Baccelli, Dohy Hong): We give an exact representation of the packet-level dynamical behavior of the Reno and Tahoe variants of TCP over a single end-to-end connection. This representation allows one to consider the case when the connection involves a network made of several, possibly heterogeneous, deterministic or random routers in series. It is shown that all key features of the protocol and of the network can be expressed via a linear dynamical system in the so called max-plus algebra. This opens new ways of both analytical evaluation and fast simulation based on products of matrices in this algebra. This also leads to closed form formulas for the throughput allowed by TCP and for the detailed dynamical behavior of the routers; these new formulas are shown to refine those obtained from earlier models which either assume that the network could be reduced to a single bottleneck router and/or approximate the packets by a fluid.

7-2 A Stochastic Model of TCP/IP with Stationary Random Losses (Eitan Altman, Kostia Avrachenkov, Chadi Barakat): We consider here a flow control mechanism in which the rate at which data is sent increases linearly in time until a loss occurs. At that point the transmission rate decreases by a multiplicative factor. This mechanism is a good approximation to TCP/IP, the congestion control in the Internet. Losses are generated by some exogenous random process, which is assumed to be stationary ergodic; no Markovian assumptions are made. We obtain an explicit expression for the average transmission rate and bounds in the case that there is a limit on the maximum rate. A set of experiments are conducted over the Internet to validate the analytical results as well as the motivations behind the work.

7-3 On the Propagation of Long-range Dependency in the Internet (Andras Veres, Zsolt Kenesi, Sandor Molnar, Gabor Vattay): In this paper we show that TCP congestion control can ”conduct” self-similarity between distant areas of the Internet. This property of TCP is due to its congestion avoidance algorithm which can adapt to self-similar fluctuations on several timescales. We analyze the behavior and limitations of this conductivity and demonstrate that TCP can conduct self-similarity above a characteristic timescale depending on the end-to-end path properties. Our analysis reveals that a TCP path with multiple self-similar bottlenecks is also self-similar and is characterized by the largest Hurst exponent. Furthermore, we show that self-similarity of one TCP stream is passed on to all other TCP streams that it is multiplexed with. These mechanisms significantly contribute to widespread scaling reported in a number of recent papers. We support our arguments with a combination of analytic techniques, simulations and real Internet traffic measurements.

8-1 Deriving Traffic Demands for Operational IP Networks: Methodology and Experience (Anja Feldmann, Albert Greenberg, Carsten Lund, Nick Reingold, Jennifer Rexford, Fred True): Engineering a large IP backbone network is impossible without an accurate view of traffic demands. Shifts in user behavior, changes in routing policies, and failures of network elements can result in significant (and sudden) fluctuations in the flow of traffic through the backbone. In this paper, we first provide a model of traffic demands, carefully constructed for traffic engineering and performance debugging of large operational IP networks. In the Internet, a large fraction of the traffic is interdomain. We represent a demand as a volume of load from an ingress link to a set of egress links to capture how routing affects the flow of traffic between domains. Second, we provide a measurement methodology for computing traffic demands, combining flow-level measurements collected at all ingress links with reachability information about all egress links. Third, we extend our methodology to cope with practical limitations in operational networks on the amount of measurement data and the number of collection locations. Specifically, we show how to estimate traffic demands using measurements collected at a smaller number of edge links – the peering links connecting to neighboring domains. For traffic entering the network on other links, we show how to infer the ingress link by combining the egress measurements at the peering links with a routing model that determines which path(s) the flow could have followed across the backbone. The traffic demand computations involve collecting, validating and joining a very large and diverse set of usage, configuration and routing data from multiple locations in the network, over the same time frame. We report on our experience in carrying out these computations. Finally, we analyze the dynamics of the traffic demands and discuss the implications on traffic engineering.

8-2 Trajectory Sampling for Traffic Measurement (Nick Duffield, Matthias Grossglauser): Traffic measurement is a critical component for the operation and management of communication networks. First, traffic measurement plays an important role in traffic engineering, as part of the network control feedback loop. Second, traffic measurement is important as a verification and trouble-shooting tool. Traffic measurement can uncover anomalies and problems that are due to faults or misconfigurations, or unforeseen consequences of policy decisions. We propose a method that allows direct observation traffic flows through a domain by observing the trajectories of a subset of all packets traversing the network. The key advantages of the method are (i) it does not rely on knowledge of the routing subsystem, (ii) it does not require router state, (iii) the reduction in loads associated with measurement, both in the router making measurements, and the network reporting them.

8-3 Measuring Link Bandwidths Using a Deterministic Model of Packet Delay (Kevin Lai, Mary Baker): Based on a new deterministic model of packet delay, we develop a novel technique, called packet tailgating, for measuring link bandwidths along a path through an internet. Packet tailgating is faster and consumes less network bandwidth than previous techniques. Unlike its predecessors, packet tailgating can detect multi-channel links, can be run on multicast trees, does not rely on consistent behavior of routers handling ICMP packets, and does not rely on timely delivery of acknowledgments. Using our implementation of packet tailgating, we demonstrate its utility on real traffic in the Internet. Our measurements indicate that for the same amount of traffic, packet tailgating is at least as accurate as other current measurement tools. We also use our deterministic model of packet delay to prove the correctness of the packet pair [Bol93] technique for measuring the bandwidth of the lowest-bandwidth link along a path (the bottleneck bandwidth). Packet pair has previously been explained intuitively but not analytically.

8-4 Practical Network Support For IP Traceback (Stefan Savage, David Wetherall, Anna Karlin, Tom Anderson): This paper describes a technique for tracing anonymous attacks back to their source. This work is motivated by the increased frequency and sophistication of denial-of-service attacks and by the difficulty in tracing packets with incorrect, or “spoofed”, source addresses. In this paper we describe a general purpose traceback mechanism based on probabilistic packet marking in the network. Our approach allows a victim to identify the network path(s) traversed by an attacker without requiring interactive operational support from Internet Service Providers (ISPs). Moreover, this traceback can be performed “post-mortem” – after an attack has completed. We present an implementation of this technology that is incrementally deployable, (mostly) backwards compatible and can be efficiently implemented using conventional technology.

9-1 When the CRC and TCP Checksum Disagree (Jonathan Stone, Craig Partridge): Traces of Internet packets over the last two years show that 1 in 30,000 packets fails the TCP checksum, even on links where link-level CRCs should catch all but 1 in 4 billion errors. Analysis of 100,000 packets which fail the TCP checksum shows the Internet has a wide variety of error sources which are not detected by link-level checksums. Analysis tools have identified nearly 100 different patterns, or likely cause, of the observed errors. These categories explain half our observe errors, and show that errors occur at all levels of the network. From the observed rate of incorrect checksums, and excluding errors that will always be caught, we conclude that the TCP checksum will fail to detect an error for rougly 1 in 200 milllion packets. From our analysis of the cause of erros, we propose simple changes to several protocols which can decrease the rate of undetected error.

9-2 Packet Types: Abstract Specification of Network Protocol Messages (Peter J. McCann, Satish Chandra): In writing networking code, one is often faced with the task of interpreting a raw buffer according to a standardized packet format. This is needed, for example, when monitoring network traffic for specific kinds of packets, or when unmarshaling an incoming packet for protocol processing. In such cases, a programmer typically writes C code that understands the grammar of a packet and that also performs any necessary byte-order and alignment adjustments. Because of the complexity of certain protocol formats, and because of the low-level of programming involved, writing such code is usually a cumbersome and error-prone process. Furthermore, code written in this style loses the domain-specific information, viz. the packet format, in its details, making it difficult to maintain. We propose to use the idea of *types* to eliminate the need for writing such low-level code manually. Unfortunately, types in programming languages, such as C, are not well-suited for the purpose of describing packet formats. Therefore, we have designed a small *packet specification language* that serves as a type system for packet formats. Our language conveniently expresses features commonly found in protocol formats, including layering of protocols by encapsulation, variable-sized fields, and optional fields. A compiler for this language generates efficient code for type checking a packet, i.e., matching a packet against a type. In this paper, we describe the design, implementation, and some uses of this language.

9-3 Terabit IP Lookups (Sandeep Sikka, George Varghese): Routers must do a best matching prefix lookup for every packet; solutions for Gigabit speeds are well known. As Internet link speeds move to OC-192 (10 Gbps) and higher, IP lookups must complete in tens of nanoseconds, requiring the use of on-chip or off-chip SRAM, which is limited by either expense or manufacturing process. In this paper, we propose an IP lookup scheme that can scale with memory speeds and yet provide worst-case guarantees. We show that doing so requires new algorithms and the breaking down of traditional abstraction boundaries between hardware and software. A particular focus of this paper is to have a lookup chip provide guarantees on the number of IP prefixes it can support. To do so we introduce new memory allocators that have provable worst-case memory utilization guarantees that can reach 100 %; this is contrast to all standard allocators that can only guarantee 20 % utilization when (for example) the requests can come in the range 1..32. An optimal version of our algorithm requires a new (but feasible) SRAM memory design that allows shifted access in addition to normal word access. This small extra feature in the memory design can double the guaranteed number of prefixes the chip can support. Our techniques generalize to other state lookups besdes prefix lookup.