The Internet as a Complex System*

Kihong Park
Department of Computer Sciences
Purdue University
West Lafayette, IN 47907
park@cs.purdue.edu

1 Introduction

The Internet, defined as the world-wide collection of IP (Internet Protocol) speaking networks, is a multifaceted system with many components and diverse features. In recent years, it has become an integral part of the socio-economic fabric. Although the Internet is still an evolving system and therefore a moving target, understanding its properties is relevant for both engineering and potentially more fundamental purposes. The Internet is a complex system in the sense of a complicated system exhibiting simple behavior, as opposed to a simple system exhibiting complex behavior. The latter aspect forms the corner stone of studies of complex systems whose rigorous underpinning is provided by dynamical systems theory. A canonical example is the logistic equation from population dynamics which possesses a rich structure including chaotic orbits and fractal non-divergent domains [41]. A somewhat different example of simple systems capable of generating complicated behavior are many-body systems, in particular, interacting particle systems [82]. We would classify such systems, despite the large number of components they admit, as "simple" due to the homogeneity and limited capability of each component—in Ising spin systems each component has two states—and the local interaction allowed of the components. Statistical mechanics studies macroscopic and emergent properties of such systems, with ergodic theory and percolation theory providing part of the mathematical foundation. Cellular automata, discrete-time interacting particle systems, are capable of universal computation which gives rise to undecidability problems and computation theoretic considerations of dynamical systems.

We view the Internet as an instance of a complicated system exhibiting simple behavior due to a compendium of innate architectural features that span a range of scientific disciplines. They include the characteristics of network traffic when viewed as a time series, closed-loop and open-loop control governing the flow of traffic which involves control theory, the connectivity structure of information networks which engages graph theory, the behavior of users and protocols—protocols are formal rules and conventions underlying information exchange—in resource-bounded, competitive environments which involves game theory, and the organizational behavior of ISPs (Internet Service Providers) which influences the network infrastructure, including peering relationships among ISPs, a domain of social sciences. A discussion of these features is the subject matter of this article.

There are few established examples of complex systems exhibiting simple behavior, excluding tautological cases such as the natural phenomena and systems studied in physics and biology.

*Supported in part by NSF grants ANI-9714707, ANI-9875789 (CAREER), ESS-9806741, EIA-9972883, ANI-0082861 (ITR), and grants from AFRL F30602-01-2-0539 (DARPA ATO FTN), Xerox, and ETRI.

The Internet is comparatively more transparent and tractable, perhaps requiring on the order of decades—as opposed to centuries for physics and biology—to uncover and understand its workings. At its physical basis, the Internet is a flow network whose information transmission is governed by communication theory pioneered by Shannon [125]. Quantum communication [14, 15], which uses entanglement in quantum mechanics for efficient communication, is in its early stages, too early to be included in the present discussion of the Internet. As a complex system exhibiting simple behavior, what makes the Internet unique are its two distinguishing characteristics: one, it is a melting pot where the key ingredients represent a confluence of several disciplines, and two, the Internet is perhaps the largest man-made many-body system. Jointly, the interdisciplinary nature of the architectural features and the engineered nature of the many-body system give rise to novel phenomena, modeling challenges, and synergistic opportunities, which have a chance of being scientifically grounded and form another corner stone of complex systems.

An example of the synergistic opportunitity is the enhanced effectiveness of game theory when played out in the context of the Internet. Von Neumann and Morgenstern advanced game theory [138], in part, to establish a mathematical foundation for economics—referred to, then, by many as the dismal science. In some respects, economics, notwithstanding the wealth of beautiful mathematics and modeling work carried out since then, has remained a dismal science for the simple reason that it continues to lack sufficient predictability. Econometric models cannot adequately account for the effect of political upheaval—never mind predicting their occurrence—a consequence of the Achilles' heel of social sciences: limited ability to factor the dynamics of human behavior, collective or singular. Game theory can provide important qualitative insights but ultimately suffers under the same predicament. A competitive game involving several human participants may be modeled by a corresponding noncooperative game involving rational self-optimizing decision processes, however, the outcomes may agree or may not agree—it all depends on what the human players actually decide to do.

The situation is less bleak when game theory is fused with the Internet. The Internet injects a measure of predictability—through behavior codification—effected by protocols that sit between a human user and the communication medium handling the transfer of information. The bulk of Internet traffic is comprised of file transfers that arise from HTTP (Hypertext Transfer Protocol) based Web traffic which, in turn, is governed by TCP (Transmission Control Protocol). TCP is a cooperative protocol that tries to achieve speedy, reliable communication. Cooperative means that TCP behaves gentlemanly upon detecting possible congestion by throttling the traffic submitted to the network. TCP is a protocol whose behavior is standardized by the IETF (Internet Engineering Task Force), the standards body of Internet protocols. Protocols, acting as automated assistants embodying codified behavior, induce a well-behaved environment without constant subjection to the inner workings and whims of human decision making. Questions involving stability and efficiency can be addressed, and quantitative predictions advanced with scientific certainty. That is not to say that humans play no part. The selection and timing of information transfer requests is under the control of human users which puts us in the driver's seat. However, the time scale of information transmission and control in broadband networks is at the millisecond level and below, a regime where most humans would be out-of-place. Thus the total picture is that of a 2-layered playing field separated by time scale where the influence of human decision making is partly delimited and isolated. Whether this crack provides a sufficient opening for reigning in the unwieldy influence of human decision making remains to be seen. The potential is there, making the Internet a fertile playing ground for a more effective game theory.

The preceding example is but one of several that will be discussed in this article. In the

next section, we give a birds-eye-view of five key architectural features of the Internet—self-similar traffic, scalable traffic control, power-law connectivity, game theory and quality of service (QoS), and organizational behavior—which are related to the Internet as a complex system theme. This is followed by sections discussing the known aspects, consequences, and challenges. The selected features meet two criteria: networking relevance and technical novelty. Networking relevance means that the complex system trait has direct bearing on Internet traffic engineering and is not a mere curiosity. Technical novelty means that understanding the complex system trait involves more than straightforward application of known ideas and techniques.

2 Interdisciplinary Features of the Internet

2.1 A packet's journey

We motivate the selection of the interdisciplinary architectural features by describing a typical—and on the surface mundane—sequence of events that transpire on the Internet. We will use this example as a skeleton to attach the five features which will, then, be given concrete meaning. The description, although oversimplified, may help give the general reader a logical glimpse of network mechanisms and their intricacies, in addition to introducing needed terminology.

Suppose a user runs a Web browser at an end system—PC, laptop, or handheld device—and clicks on a link containing the location information of an object such as a HTML (Hypertext Markup Language) document or some other file (e.g., executable binary, audio or video data) that is to be accessed using HTTP. This triggers a HTTP request message that is passed down to TCP in the protocol stack—protocols in the operating sytem (OS) of an end system are organized in a partial order represented as a protocol graph—which encapsulates the HTTP request by treating it as payload. This is akin to an already sealed letter going into a FedEx envelop. TCP memorizes the packet information in the event it needs to resend it: the Internet is "leaky." TCP's packet is handed down to IP, a protocol responsible for routing. IP determines where to forward the packet to so that it can come closer to reaching its final destination. IP performs its own encapsulation and hands the resultant packet to the link layer. A popular link layer is Ethernet—standardized by IEEE (Institute of Electrical and Electronics Engineers) under IEEE 802.3 for wired and IEEE 802.11 for wireless media. The link layer has access to the physical address of the next hop's IP address—every network device has a unique physical address—encapsulates the IP packet with an envelop containing physical addresses, and hands it down to the physical layer. The physical layer oversees the transmission of information containing the link layer packet over its communication medium.

The physical layer at the receiving end decodes the transmission and does a hand-off to the appropriate link layer protocol above. Assuming the receiver is an IP speaking device, the link layer protocol decapsulates and hands off to the IP layer which determines whether additional forwarding is required to reach the final destination. If so, the packet is encapsulated and passed down the protocol stack. This process is repeated at every IP-enabled device—called router—on the forwarding path until the destination IP device is reached. At that point, the IP layer passes its payload up to TCP which, in turn, hands off its payload to HTTP, and HTTP to its application. In this example, a Web server that processes the HTTP request. This prompts a HTTP response, which is passed down the protocol stack at the destination IP device and returned to the original sender, the client. Several things can go wrong on an IP packet's journey. The packet, during physical transmission, may get corrupted due to noise or interference—especially severe in wireless

segments of the Internet—which results in dropping of the corrupted packet when so detected. The packet, upon arriving at a router, may find the router busy processing other waiting packets, and even worse, find no room or buffer space which causes the packet to be discarded. Less frequently, a router may fail, erasing the transiting packet residing in its memory. IP has no provision for dealing with packet loss, which means that TCP running at the sender with the assistance of its counterpart running at the receiver is the earliest point at which recovery can be attempted. A dumb network core-intelligent network edge is characteristic of the Internet's design, referred to as the end-to-end paradigm [33].

2.2 Complex systems features

We revisit the packet journey example and point out several innate, interdisciplinary features that reside with the skeleton of packet forwarding and information transmission mechanics.

Self-similar traffic. An important engineering consideration is the load or traffic—measured in bits per unit time—impinging on a bottleneck router as excessive traffic can result in congestion and information loss. The same is true of end systems, in particular, servers. If traffic, viewed as a time series, is undulating with severe peaks and valleys, the capacity allocated to handle demand must be correspondingly bursty to match the time-varying demand. This is to avoid losses—translated to delay when sufficient buffer space is available to hold excess traffic—and reduce resource wastage stemming from overprovisioning which carries an economic cost. Flat traffic is desirable because it is predictable and obviates the need to frequently reshuffle resources which, in many instances, is difficult to do. Whatever our wishes, Internet traffic may follow its own set course, and understanding its properties is fundamental to effective traffic engineering. Two obvious factors that influence traffic demand are the arrival pattern, in time, of user requests—e.g., clicking on a Web link in the packet journey example—and the size of the file or information object requested. Other things being equal, the more frequent the user requests and the larger the requested files, the higher the average load experienced by a network system. Our interest concerns the shape of the resultant traffic.

From a server's perspective or a router's perspective that lies on frequented paths to popular servers, request arrivals from different users may be construed to be independent, at least at time scales on the order of minutes and below. If indeed so, the law of large numbers (LLN), assuming users are many, induces statistical regularity conducive to flatness on two fronts: one, the number of user requests occurring during a time window will concentrate around a mean, and two, the number of user requests across two disjoint time windows will become uncorrelated. To a first approximation, the arrival interval between successive requests has been observed to be independent, with an exponential distribution and resultant total load that is Poisson. The action of LLN across space and time has been the central property targeted and harnessed by traffic engineering tools in telephony and data communication. To the surprise of many, traffic measurement collected in the late 1980s at Bellcore on Ethernet showed that traffic was bursty at large time scales [81], inconsistent with the flatness predicted by independent or weakly correlated arrivals. In fact, Bellcore's Ethernet data exhibited self-similarity in the sense of variability, as captured by correlation, remaining invariant across a wide range of time scales: from milliseconds to seconds to tens of minutes. This phenomenon has been confirmed in other contexts since then and shown to be the norm rather than the exception. One factor in the packet journey example we ignored is the size of requested files. It turns out that size does matter, and the peculiar

distribution of file sizes—most files are small but a few are very large[1], also referred to as "mice and elephants"—effects the fractal characteristic of Internet traffic. Traffic self-similarity is an emergent, macroscopic trait of the Internet whose seed can be found in microscopic properties of its components. The fractal dynamics of Internet traffic makes it a representative example of the Internet as a Complex System metaphor.

Scalable traffic control. The self-similar nature of Internet traffic has served to draw attention to the importance of empirical measurement, almost to the extent of treating the Internet as a "natural" phenomenon in its own right, whose properties, even to the designers, may remain initially hidden. The Internet, notwithstanding its phenomenological richness, is an artificial, engineered system, a key distinguishing feature compared to complex systems arising in nature. The Internet is a controlled system where the bulk of daily traffic is regulated by TCP, a feedback control designed to affect speedy, reliable data transfer while avoiding congestion. TCP implements nonlinear control, is subject to the effect of feedback latency, and is instantiated in an environment containing thousands of other TCP flows competing for shared network resources—bandwidth and buffer space—at bottleneck routers. For example, the HTTP session in the packet journey example is but one of many traversing a bottleneck link. Coupled TCP flows, when conditions are ripe, may synchronize [145]—a phenomenon abundant in nature such as the synchronized flashing of fireflies [112]—leading to periodic underutilization and overutilization. Even without oscillatory synchronization, TCP is subject to stability problems due to feedback latency and congestion avoidance. Fairness—in the sense of equal share—may be violated when one TCP flow sharing a bottleneck link with another has to traverse many more hops, which puts the longer flow at a comparative disadvantage: due to increased feedback latency news arrives more slowly and consequently is less useful in control actions. TCP, in some instances, can contribute to self-similarity of network traffic as its nonlinear control, when reducing the sending rate during persistent packet loss, injects exponentially increasing wait periods that can translate to prolonged idleness, a form of correlation. However, it is a secondary factor compared to the dominant role played by file size distribution. Congestion control, of which TCP is an instance, has the potential to induce complicated dynamics, representing another aspect of complex systems.

Related observations hold for routing, the second of the two major Internet traffic controls. Synchronization of routing updates can affect cyclic busy periods during which part of a router's capacity is turned away from packet forwarding [54]. Routing tables may take a long time to stabilize, creating time windows during which packets are misrouted [80]. Quality of service (QoS) is another important pillar of traffic control. Its present impact, however, is limited compared to congestion control and routing due to almost non-existent deployment. That is not to say that routers on the Internet do not possess QoS capabilities. They, in fact, are endowed with a slew of IETF standardized mechanisms, but these features are not activated outside of isolated testbeds. Several of these are candidate building blocks in a future QoS-enabled Internet, still a technical as well as socio-economic challenge.

Power-law connectivity. In the late 1990s, two separate, but intimately related, phenomenological features of the Internet were discovered [4, 50]. The first concerns the connectivity structure of the World Wide Web (WWW), where two Web pages, viewed as nodes in a graph, are defined to be joined by an edge if there is a link from one to the other. The second concerns the business-to-business relationship between domains or autonomous systems that represent organizational units. Autonomous system (AS) is a technical term and part of the Internet standard. The Internet AS

[1]The technical definition—heavy-tailedness—will be discussed in Section 3.

topology is defined as a graph where the nodes are ASes, and an edge exists between two nodes if the ASes—more precisely, border routers belonging to the ASes that affect the transfer of packets between domains—are connected by one or more physical links representing customer-provider relationships. What these measurement graphs showed is that their connectivity structure is far from random, exhibiting power-law decay—as opposed to exponential decay—in the size of the neighborhood of a node. Random means that a node is connected to any other node with some fixed, independent probability. In the case when the probability is 1/2, this is tantamount to saying the set of all graphs with a given number of nodes, when viewed as a discrete sample space, has uniform probability. The measurements showed that not all nodes are created equal, and not all graphs are created equally likely.

A characteristic trait of random graphs is that most nodes look alike, possessing about the same number of links that are strewn all over the place. The probability that a node has a neighborhood size deviating from the mean is exponentially small, a consequence of LLN. In WWW and Internet domain graphs, the decay was observed to be polynomially small—a power-law, and thus the name power-law graph—admitting nodes with very many neighbors. With 20/20 hindsight, it is perhaps not unusual that WWW and Internet domain graphs should exhibit power-law connectivity. During long flights one may have browsed through an airline magazine finding the carrier's route map in the back pages. Power-law graphs, visually, resemble airline route maps where nodes corresponding to major hubs have many links that connect smaller regional airports. Some regional nodes connect to more than one hub, and hubs are connected to each other through a backbone. Two important aspects of power-law graphs are captured by the expressions: "the rich get richer and the poor get poorer," and "a few are connected to many, many are connected to a few." The first represents a dynamic viewpoint, whereas the second conveys a static structural aspect. A number of studies in the 1950s [146, 126, 87] showed power-law (also called Zipf law in special cases) skews in social phenomena which were attributed to the intuition that already popular entities were likely to become even more popular—in part, through a bandwagon effect—whereas unpopular entities faced the opposite predicament. In Internet domain graphs, a major service provider acting as a conduit for other providers and stub customers—a stub AS is a domain that does not provide transit service to other domains—is likely to attract more customers compared to smaller providers. New customers may perceive an advantage in connecting to an established provider with an existing, large customer base that is accessible by a single hop. This is in addition to the perceived reliability associated with established organizations and brand names. An extreme form of a few nodes being connected to many and many being connected to a few is the star topology: there is a single central node from which all other nodes emanate. When there are multiple high degree nodes—the degree of a node is the number of links incident on the node—they resemble locally star-like subgraphs that are connected to each other through a backbone network. There are a number of consequences of power-law connectivity for network security and performance, some relating to percolation and phase transition. They will be discussed in Section 4.

Game theory and QoS. As indicated in the Introduction, game theoretic considerations arise naturally in congestion control where multiple sessions share a common bottleneck link through which traffic must be scheduled. In the packet journey example, two servers transmitting information to clients across a common bottleneck may distinguish themselves by employing clever congestion controls that outperform the competition—inclusive throughput gains obtained at the expense of others—that translate to faster response times and commercial advantage. Supposing throughput is the performance metric that selfish users and their protocols aim to maximize, the noncooperative congestion control game that pits selfish congestion control protocols against each

other is an instance of Tucker's Prisoner's Dilemma game. In the 2-player setting, at the onset of congestion, the protocols can act cooperatively and throttle their sending rate, alleviating congestion and attaining an equitable share of the total achievable throughput. This corresponds to neither prisoner, when interrogated in separate rooms, ratting out the other. Each prisoner receives a two-year sentence with possibility of parole. If one protocol acts cooperatively but the other does not—the cooperative protocol backs off when congestion arises while the noncooperative one fills the slack—the game leads to a state where bandwidth is monopolized by the noncooperative protocol. This corresponds to the cooperative prisoner receiving a 20-year sentence for staying mum, while the selfish prisoner who betrayed his colleague gets off scott free. When both protocols behave selfishly, congestion persists with each achieving a small throughput.

The Prisoner's Dilemma illustrates that noncooperative games, in general, lead to equilibria—if they exist—that are less desirable than those reachable by corresponding games where players are cooperative. Indeed, only under special circumstances does Adam Smith's invisible hand lead to an efficient orchestration of shared resources. A somewhat different example of the adverse influence of selfishness is Braess' paradox [20]. It describes a situation where adding resources to a network can lead to a deterioration of performance when selfish, shortest-path routing is employed by individual user flows. This paradox—"how can things be worse when resources are more plentiful?"—cannot arise when routing actions are cooperative. Although the causes underlying Braess' paradox are well-understood, the extent to which the paradox manifests itself in Internet routing is unknown. Selfish routing can also affect instability in the form of a ping-pong effect. Users routing traffic over a congested link, upon discovering a newly uncongested link, may switch over in tandem in an attempt to improve individual performance causing the uncongested link to be swamped. This results in a state where the previously congested link becomes uncongested, prompting a reverse migration and consequent oscillatory behavior. In Section 5 we will discuss noncooperative network games where players share multiple classes of bounded resources—the congestion control game being a special case where there is a single shared resource—and consider the game structure of noncooperative multi-class QoS provisioning.

Organizational behavior. An important variable of Internet traffic engineering is the organizational behavior of autonomous systems which intimately impacts routing, QoS, topology, and deployment of new traffic management solutions. Consider the packet forwarding process at IP routers in the packet journey example. When a user downloads a file from a Web server, packets carrying the content of the file are routed by two separate subsystems: inter-domain routing and intra-domain routing. Supposing the client and server machines reside on different domains—generally the case although caching tries to place frequently accessed content close to where the demand is—the path undertaken by packets at the granularity of domains is determined by BGP (Border Gateway Protocol), an inter-domain routing protocol that governs packet forwarding across ASes. The path chosen by a packet as it traverses IP routers within a domain is determined by intra-domain routing protocols. Two examples are OSPF (Open Shortest Path First) and RIP (Routing Information Protocol). OSPF implements Dijkstra's algorithm [42] for computing shortest paths—a centralized method—and RIP implements Bellman-Ford's algorithm [13, 55] which is decentralized. Whereas intra-domain routing follows the shortest-path principle, inter-domain routing is policy based, meaning that an organization can inject economic, political, and other criteria it deems relevant, including shortest-path, when making routing decisions. This can lead to scenarios where a company, when sending an e-mail to another company across the street, has its message routed to a different continent before eventually reaching the destination just a holler away. Policy influences peering relations between ASes—who is connected to whom—the substrate upon

which inter-domain routing operates. Organizational behavior implicitly gives rise to power-law connectivity of Internet AS topology.

Organizational behavior affects the viability of new Internet technology deployment, especially as they pertain to quality of service. The service quality experienced by a user is end-to-end, encompassing end systems and their resources—e.g., CPU (Central Processing Unit) speed and access bandwidth—and the state of intermediate hops that packets must traverse to reach their destination. End-to-end QoS is only as good as the weakest link in the resource chain whose components may span multiple autonomous systems under the governance of different organizations. To achieve guaranteed QoS, say, in the form of a dedicated 128 Kbps (kilo-bit-per-second) communication channel for CD quality real-time audio, all ASes on an end-to-end path must participate and reserve the required resources. Although the technology for performing the necessary coordination and signalling is there, such services presently do not exist on the Internet outside of limited settings such as leasing of lines from a single service provider and bandwidth commodity markets that trade raw bandwidth without the nuts-and-bolts needed to achieve on-demand QoS across domains. In telephony, an international call will traverse multiple carriers, but agreements exist that allow a dedicated end-to-end channel to be set up on the fly. In the airline industry, codesharing allows multiple carriers to route passengers across their collective routes, reserving a seat on each leg of an end-to-end journey. Of course, we are ignoring overbooking, flight delays, and other factors that contribute to the end-to-end flying experience. On the Internet, policy barriers between administrative domains prevent QoS solutions to be realized, prompting a revival of application layer methods that package services relying only on IP's reachability functionality. Giving up on the ability to exercise direct resource control facilitates deployability, however, at the cost of performance.

3 Self-Similar Traffic

3.1 What is self-similar traffic?

Suppose we instrument a router so that we can monitor and record the number of bytes (or packets), per unit time, that exit a link. The time unit, for example, may be 10 milliseconds which implies that every logged measurement represents the total number of bytes that have left through the link during a 10 msec interval. This results in a time series $X(t)$ with t discrete. Let $X^{(10)}(t)$ denote the time series arising from the same measurement process with the difference that the time unit is 100 msec, i.e., 10-fold coarser. Similarly, $X^{(100)}(t)$ denotes measurement at the granularity of 1 sec and, in general, we define the aggregated time series at aggregation level m as $X^{(m)}(i) = \sum_{t=m(i-1)+1}^{mi} X(t)/m$ which is composed of averaged, non-overlapping m-blocks.

Figure 1(a) shows TCP/IP traffic where the top plot shows measurements at the 100 sec granularity over a 10,000 second measurement period. Hence the plot contains 100 data points or samples. The second plot from the top shows traffic logs at the 10 second granularity where the measurement data are taken from the first 1,000-second interval as indicated by the rectangular time window. In a similar vein, the third and fourth plots are obtained from their preceding time series by zooming in on the first 10 data points and magnifying the detail 10-fold. Figure 1(b) shows corresponding multi-scale plots for Poisson traffic that is more representative of telephone traffic. We observe a stark difference between Figure 1(a) and Figure 1(b): for network traffic variability or burstiness is preserved across four orders of time scale whereas for telephony-like traffic the aggregated plots become rapidly flat with increasing time scale.

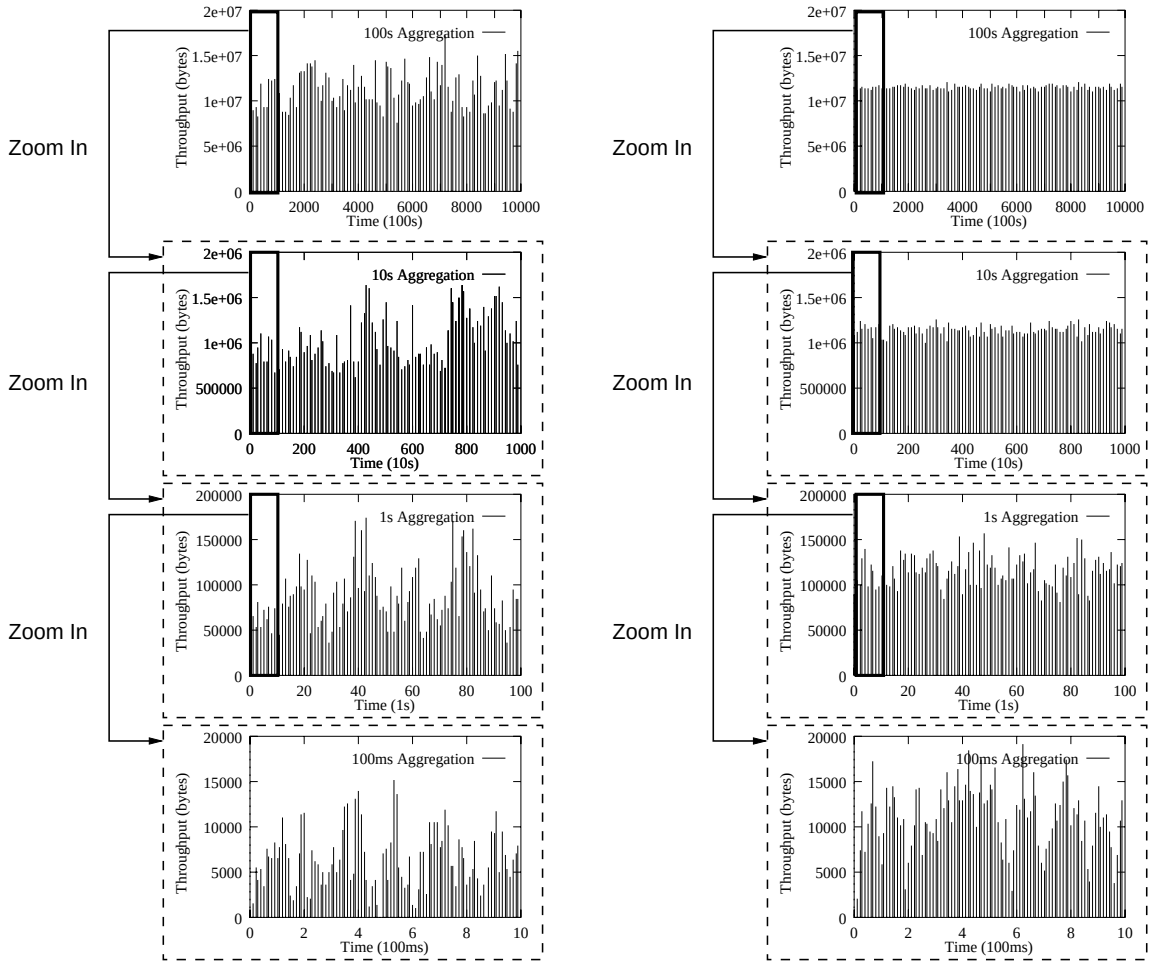


Figure 1: Self-similar burstiness. (a) Long-range dependent network traffic. (b) Short-range dependent Poisson traffic.

If $X(t)$ were independent, in particular, i.i.d. (independent identically distributed) with finite variance σ^2, $X^{(m)}$ can be viewed as a sample mean and its variance is given by $\mathrm{Var}(X^{(m)}(i)) = \sigma^2/m$. Thus, as a function of the sample size, i.e., aggregation level m, the rate of decay is polynomial in m with exponent -1. We estimate the shape of $\mathrm{Var}(X^{(m)})$ for the traffic series in Figure 1(b) and find that $\mathrm{Var}(X^{(m)}) \propto m^{-\beta}$ with β close to 1. If we do the same for Figure 1(a), however, we find that $\mathrm{Var}(X^{(m)}) \propto m^{-\beta}$ where β is closer to 0. Thus for network traffic exhibiting self-similar burstiness, the variance at larger time scales diminishes, albeit with a rate that is slower than that of independent traffic. This implies that the traffic series in Figure 1(a) is correlated—the closer β is to zero the stronger the correlation—indicative of long-range dependence. By convention, the exponent β is denoted as $\beta = 2 - 2H$ where H is called the Hurst parameter after the hydrologist Hurst who studied reservoir capacity planning using Nile River data [66]. H close to 1/2 is associated with short-range correlation, and H close to 1 is indicative of long-range correlation. If we

estimate the autocorrelation function $r^{(m)}(k)$ for the time series in Figure 1(a) where k is the time lag, we find that

$$r^{(m)}(k) \approx r(k) \propto k^{2H-2} \tag{1}$$

with H close to 1. It is in the sense of (1)—the correlation structure across time scales is preserved—that long-range dependent traffic is self-similar. In mathematical modeling of self-similar traffic, we may consider second-order stationary $X(t)$ whose autocovariance satisfies

$$\gamma^{(m)}(k) = \frac{\sigma^2}{2}\left((k+1)^{2H} - 2k^{2H} + (k-1)^{2H}\right) \tag{2}$$

for all k and m. $X(t)$ is called exactly second-order self-similar with Hurst parameter $1/2 < H < 1$. (2) implies (1) where the autocorrelation structure is exactly preserved. When (2) holds asymptotically in m, $X(t)$ is called asymptotically second-order self-similar. Property (1) also holds only asymptotically, i.e., $r^{(m)}(k) \sim r(k)$. We refer the reader to [17, 108] for a more comprehensive discussion.

3.2 What causes traffic self-similarity?

3.2.1 Structural traffic models

If we compare the 10 second aggregation plots of Figure 1(a) and Figure 1(b), we find that their average is about the same. However, the left plot possesses significant undulations whereas the right plot is fairly even. In the left plot there are 10 second intervals where traffic demand significantly exceeds the average and time intervals where demand significantly falls below. Matching supply, i.e., resources, to demand requires correspondingly variable resources that can be adjusted to the time-varying demand if loss or waiting during peak periods is to be avoided. Frequent reshuffling of resources can be difficult and, in some cases, such as installing physical lines and routing equipment, outright infeasible. If resources are allocated at the peak rate then customer satisfaction is assured, but at the cost of wasted or underutilized resources during lull periods which may translate to increased resource/service prices. Utilization can be improved by lowering the allocated bandwidth, albeit at the expense of quality of service. Bandwidth and buffer space—the two principal network resources—are not storable commodities. That is, when a 100 Mbps link is utilized 50% during an hour, 50 Mbps of unused bandwidth is not saved and available for consumption during the next hour for a total bandwidth budget of 150 Mbps. The use-it-or-lose-it nature of network resources—similarities exist with perishable goods—complicates resource provisioning when traffic demand is bursty across a wide range of time scales. In the case of flat traffic, it is possible to "have the cake and eat it too": both high QoS and high utilization are attainable.

Traffic self-similarity is not a mere curiosity but has potentially serious ramifications for network engineering. Thus understanding its causes is important. Let us consider an abstraction of a user's traffic flow, called the on-off model, where a user alternates between on (i.e., active) and off (i.e., inactive) states. During an on-period, data transmission is assumed underway—by default, at constant rate—also referred to as a packet train [70]. During an off-period, the session is assumed to be idle with no traffic emitted. When viewed as a stochastic process, the on-off model can be described by a set of random variables $\tau_{\text{on}}(1), \tau_{\text{off}}(1), \tau_{\text{on}}(2), \tau_{\text{off}}(2), \ldots$ denoting the lengths of the first on-period, the first off-period, the second on-period, and so forth. The on-period random variables are assumed to be i.i.d. and independent of the off-period random variables, and similarly for off-period random variables. An instance of an on-off traffic process is defined by the distributions of τ_{on} and τ_{off}. The collective traffic impinging at a router's bottleneck link, which

is comprised of several user flows, may be modeled as a superposition of multiple on-off processes. This is depicted in Figure 2.

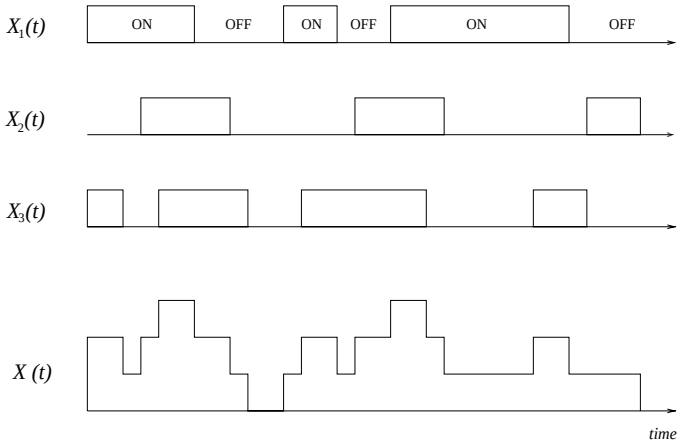


Figure 2: Superposition of three on-off processes.

Traffic measurements have shown that on-periods tend to be heavy-tailed while off-periods are light-tailed [142]. A probability distribution is heavy-tailed with index $0 < \alpha < 2$ if the tail of the distribution follows a power-law

$$\Pr\{Z > x\} \sim c\,x^{-\alpha} \tag{3}$$

for large x. The precise mathematical counterpart of heavy-tailed random variables are regularly varying random variables [90] whose definition involves the use of slowly varying functions, a technical detail we will ignore here. A canonical heavy-tailed distribution is the Pareto distribution whose distribution function is given by $\Pr\{Z \leq x\} = 1 - (b/x)^{\alpha}$, $b \leq x$, where $0 < \alpha < 2$ is the shape parameter (or tail index) and b is called the location parameter. The Pareto distribution has a power-law tail for all $x \geq b$. Heavy-tailed random variables possess infinite variance, and if $0 < \alpha \leq 1$, they also have an unbounded mean. For traffic modeling purposes, we are interested in the regime $1 < \alpha < 2$. Practically, an on-off process with heavy-tailed on-periods and light-tailed—by default, exponential—off-periods gives rise to many short traffic bursts mixed in with a few very long transmissions. This captures the empirical fact that most TCP sessions are short-lived whereas a few are long-lived. Superposition of independent on-off processes with heavy-tailed on periods leads to asymptotic second-order self-similarity [131], in particular, fractional Gaussian noise [86], a generalization of Gaussian noise ($H = 1/2$). An equivalence relationship holds where the superposition of on-off processes is long-range dependent ($H > 1/2$) if and only if the on-periods—or the off-periods—are heavy-tailed ($1 < \alpha < 2$). For this reason, in the second-order self-similarity context, long-range dependence—technically defined as having a non-summable autocorrelation function—and self-similarity are used interchangeably.

An intimately related, perhaps even more succinct traffic model leading to second-order self-similarity is the $M/G/\infty$ traffic model. We think of traffic as being generated by Poisson arrivals—the inter-arrival time between successive sessions is exponentially distributed—where the lifetime of sessions is heavy-tailed. A session is viewed as a single on-period. This is illustrated in Figure 3. Customers arrive randomly, and most customers are smallish ("mice") whereas a few are very big ("elephants"). $M/G/\infty$ denotes a queueing system, so its interpretation as a traffic model needs some explanation [35, 110]. $M/G/\infty$ describes a queueing system where customers arrive randomly—i.e., their inter-arrival time is exponentially distributed (the "M" in Kendall's notation

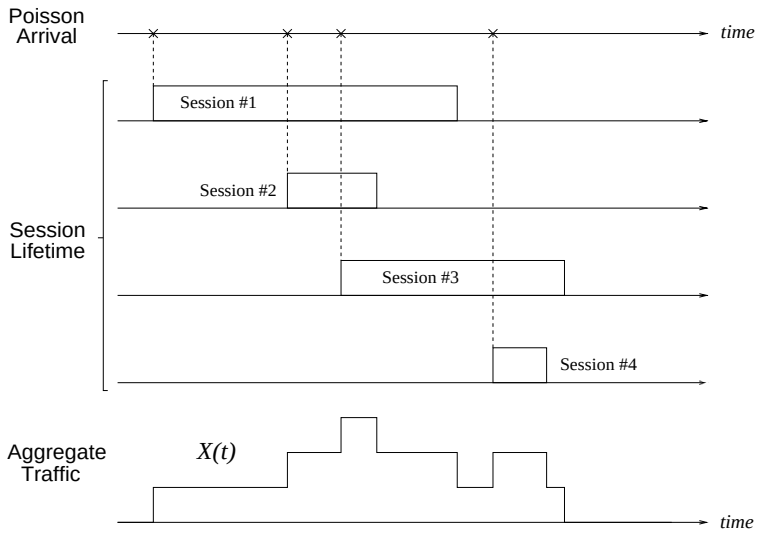
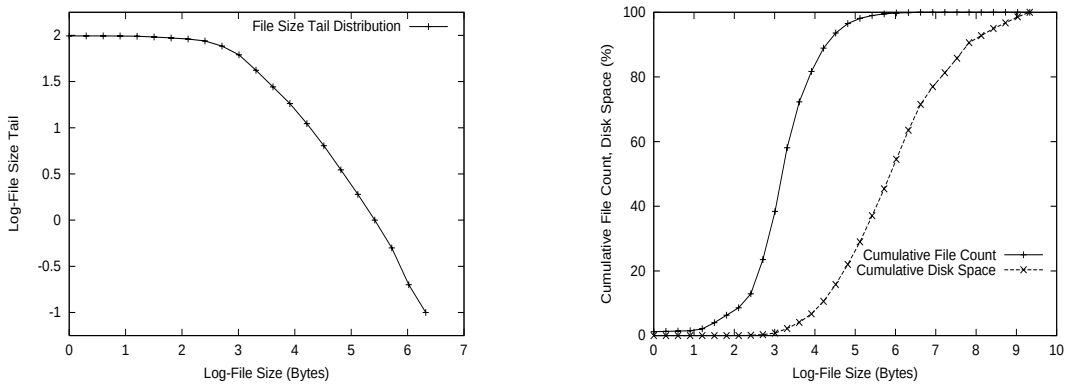


Figure 3: $M/G/\infty$ traffic model.

which standards for Markovian)—there are an infinite number of servers (the "∞"), and a newly arriving customer is assigned to one of the idle servers. At any given time only a finite number of servers are busy. The time to service a customer can follow any distribution (the "G" which stands for general). We are specifically interested in heavy-tailed service times. The performance variable of interest in $M/G/\infty$ is the counting process $X(t)$ which at time t tallies how many servers are busy. It is readily checked that the busy server process $X(t)$ of an $M/G/\infty$ queue with heavy-tailed service times corresponds to the traffic model where session arrivals are Poisson and session lifetimes are heavy-tailed. Again, the essential ingredient that renders the $M/G/\infty$ traffic model long-range dependent is the heavy-tailedness in the service time.

3.2.2 Causality: Heavy-tailed file sizes

Since empirical measurement data pointed to heavy-tailed session lifetimes [110, 142], a question remained as to why this is the case. This was addressed in [102] where UNIX file systems research carried out in the 1980s was examined and shown to indicate skews in file sizes consistent with heavy-tailedness. In [39] file access by Web clients at Boston University during Nov. 1994–Feb. 1995 were shown to be heavy-tailed. The role of UNIX file systems for explicating the causality of self-similar network traffic is relevant for two reasons. First, the original Bellcore data, the basis for Leland et al.'s seminal study [81], was collected during 1989–1992 when the World Wide Web did not exist yet: the Web could be a facilitator but not the root cause of traffic self-similarity. Second, empirical evidence from the 1980s in the file system research community predating the self-similar traffic discovery provided independent support that heavy-tailedness of file sizes may be a phenomenon that is not necessarily specific to the Internet. Figure 4(a) shows the log-log tail distribution of file sizes from a 1993 survey of UNIX file systems [68]. For large file sizes—the detailed structure of small files has negligible influence on long-range dependence—we find a straight-line fit consistent with heavy-tailedness. Figure 4(b) shows the cumulative distribution of the percentage of files of a certain size and the corresponding cumulative disk space consumed. File size is shown in log-scale with base 10. We observe that close to 90% of files are of size less than 10,000 bytes. Collectively they consume less than 10% of the disk space. The minority of files that



(a) (b)

Figure 4: Heavy-tailed UNIX file size distribution. (a) Log-log plot of file size tail. (b) Cumulative distribution of file size count and disk space.

are bigger than 10,000 bytes—some are very large—consume the bulk of the disk space. Together they invoke the metaphore "many mice and a few elephants." We will encounter power-law skews again when discussing Internet connectivity in Section 4. The relevance of Cunha et al.'s Web access measurements [39] (and their further analysis in [36]) derives from the fact that file access is not synonymous with file size distribution: a file server may be heavy-tailed but if users only access small files, then the file access distribution—not file size distribution—becomes the variable of interest. The results in [39] showed that user access behavior resembled random sampling from heavy-tailed file size distribution.

3.2.3 Influence of traffic control

To complete the reductionist reasoning and causal chain—(i) traffic is self-similar because multiplexing of sessions with heavy-tailed lifetimes leads to self-similarity, (ii) session lifetimes are heavy-tailed because most traffic is TCP file transfer traffic and file sizes are heavy-tailed, ergo (iii) traffic is self-similar because file sizes are heavy-tailed—required showing that TCP does not significantly interfere with the transfer process of heavy-tailedness of file size distribution into heavy-tailedness of session lifetime. In [102] it was shown that TCP approximately preserved this transfer relationship, and coupling of TCP sessions at shared bottleneck links did not break the chain. On the other hand, UDP (User Datagram Protocol) traffic that is not feedback controlled—UDP is a protocol operating at the same layer as TCP whose essential function is to identify the application layer process that a packet is destined to—resulted in reduced self-similarity during contention periods. This can be understood by noting that TCP conserves information due to its retransmission based reliability mechanism whereas UDP—unless application layer protocols running above it implement their own reliability mechanism—suffers information loss which diminishes the impact of the file size tail. During high contention accompanied by persistent packet loss, TCP stretches its transmission into an on-average thin stream where the reduced signal strength stemming from thinning is compensated by the lengthened file transfer completion time. Presence of duplicate retransmissions may amplify the total number of bytes transmitted by TCP. In the case of plain UDP, this "conservation law" need not hold.

An on-average flatish TCP session contains internal structure at the time scale of the round-trip time (RTT), i.e., network latency, at which feedback control actions are undertaken. The output behavior of a typical long-lived TCP session follows the shape of a saw-tooth interspersed with varying lull periods. Linear ascend in the saw-tooth is affected by TCP's feedback control which additively increases traffic submitted to the network when unused bandwidth is deemed available. Sharp descend in the saw-tooth pattern stems from TCP's multiplicative back-off which is

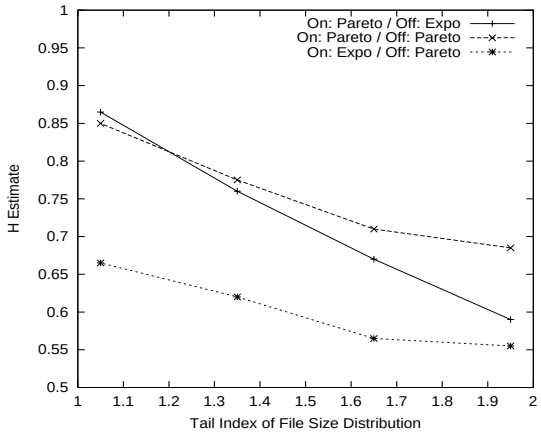


Figure 5: Hurst parameter of on/off TCP sessions: Pareto file size/exponential idle period, Pareto file size/Pareto idle period, and exponential file size/Pareto idle period.

instituted at times of perceived packet loss. The lull periods are introduced by TCP's retransmission mechanism which injects exponentially increasing pause periods during successive retransmission attempts. Small time scale TCP dynamics can lead to chaotic trajectories and contribute to longer time scale correlation [136, 137]. In particular, both exponentially increasing idle periods—extended inactivity is a form of correlation—and linear growth of throughput that elongates transmission duration contribute to long-range correlation. The magnitude of this contribution, however, is small when compared to the impact of heavy-tailedness in inducing long-range dependence of network traffic. This can be seen in Figure 5 (based on results from [102]) where the Hurst parameter of aggregate traffic stemming from 32 concurrent TCP connections that share a bottleneck link is shown for three different cases: (i) file sizes are drawn from a Pareto distribution with the specified tail index and idle periods between successive file transfer sessions within a connection are set exponential; (ii) both file size distribution and session inter-arrival times are set to be Pareto; (iii) file sizes are set to be exponential but idle times are set to be Pareto. Case (i) corresponds to the canonical configuration of practical interest reflecting Internet workload measurements. In case (ii), Pareto inter-session arrival times within a TCP connection inject an additional measure of long-range dependence that for lighter tails (e.g., $\alpha = 1.65$ and 1.95) exact a noticeable effect. For heavier tails (e.g., $\alpha = 1.05$ and 1.35), however, the effect is negligible. The most relevant scenario is case (iii) where exponential file sizes remove the heavy-tailed file size factor of long-range dependence, while Pareto idle times inject long-range correlation stronger than those induced by TCP's exponential backoff. As the tail index approaches 1, we observe an increase in the Hurst parameter. However, its magnitude (around 0.65) is significantly smaller than that of case (i) (around 0.85) representative of Internet traffic. It is for this reason that heavy-tailed file size distribution constitutes the dominant cause of long-range dependence in Internet traffic, with TCP dynamics playing a tertiary role.

3.3 Performance implications of self-similar burstiness

3.3.1 Queueing: How long must one wait

A key concern of network performance evaluation is how long packets arriving at routers have to wait before they are processed and sent on their way. The same is true of client requests at servers. From everyday experience at toll booths, restaurants, and fast food pick-ups, we know that the waiting time is influenced by the bursty nature of arrivals—sometimes attributed to "luck" or the lack thereof—in addition to time-of-day and overall traffic intensity. Such phenomena are studied in queueing theory [76] where the biggest challenge lies in understanding the sometimes complicated structure of bursty arrivals and their impact on waiting. To understand the influence of heavy-tailedness in the input on queueing, we will consider a simplified queueing system with traffic input $X(t)$, storage occupancy $Q(t)$, service rate μ, and storage capacity S, where packets are treated as fluid and $S = \infty$. The motion of the system is governed by

$$Q(t + \Delta t) = \max\{Q(t) + X(t, t + \Delta t) - \mu\Delta t,\ 0\} \tag{4}$$

where $X(t, t + \Delta t)$ denotes the input during the specified time interval and the queue drains at constant rate μ. Thus storage occupancy at Δt in the future is determined by adding the net influx, $X(t, t + \Delta t) - \mu\Delta t$, to the current occupancy level. Equation (4) defines a random walk with a reflecting barrier at zero. To prevent the queue from growing out of bound, we require that there be a negative drift, $\lambda < \mu$, where λ is the average traffic rate of input $X(t)$. Assuming the system is well-behaved, we seek to find the equilibrium distribution of $Q(t)$, $Q(\infty)$, the main performance variable. When the distribution of $Q(\infty)$ has an exponential tail, i.e., $\Pr\{Q(\infty) > x\} \propto e^{-ax}$, buffer dimensioning becomes cost-effective since an additional unit of buffer capacity multiplicatively decreases the probability of overcrowding. The bulk of Markovian or short-range dependent traffic lead to $Q(\infty)$ with light tails. For long-range dependent (LRD) traffic, the picture is starkly different: heavy-tailedness in the input is preserved and translated to heavy-tailedness in the queue length distribution. Thus the probability of overcrowding for large x is significantly amplified rendering buffer dimensioning a less cost-effective resource provisioning strategy for accommodating crowds. We will illustrate this transfer relation using a single on-off process with heavy-tailed on-periods and light-tailed off-periods.

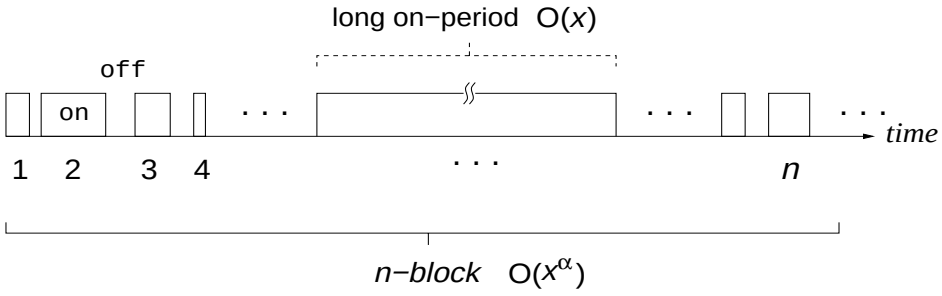


Figure 6: Block of n alternating on- and off-periods at time scale $n = O(x^\alpha)$.

Let the on-period be Pareto with tail index α and location parameter b, and let the off-period be exponential with parameter λ_{off}. Let λ_{on} denote the traffic rate during an on-period. Then $\lambda = \lambda_{\text{on}} E\{\tau_{\text{on}}\}/(E\{\tau_{\text{on}}\} + E\{\tau_{\text{off}}\})$, and $\lambda < \mu < \lambda_{\text{on}}$ is the regime where queueing behavior is nondegenerate: neither unbounded nor empty. Suppose we want to lower bound the tail probability

$\Pr\{Q(\infty) > x\}$ for large x. Let $\tau_{\rm on}(1)$, $\tau_{\rm off}(1)$, $\tau_{\rm on}(2)$, $\tau_{\rm off}(2)$, ..., $\tau_{\rm on}(n)$, $\tau_{\rm off}(n)$ denote a block of n alternating on- and off-periods where $n = y^\alpha/b^\alpha$ and $y = x/(\lambda_{\rm on} - \mu)$. Since $\Pr\{\tau_{\rm on} > y\} = (y/b)^{-\alpha}$, the expected number of on-periods in the n-block that exceed y is one. Figure 6 depicts the n-block configuration. Let L denote the length of the "long" on-period. Its expectation is given by

$$E\{L\} = E\{\tau_{\rm on} \,|\, \tau_{\rm on} > y\} = \int_0^\infty (s+y)\frac{\alpha y^\alpha}{(s+y)^{\alpha+1}} ds = \frac{\alpha}{\alpha - 1} y$$

where $\alpha y^\alpha/(s+y)^{\alpha+1}$ is the conditional probability. y is set to $x/(\lambda_{\rm on} - \mu)$ so that the queue is assured to build up to x. Since $\lambda_{\rm on} > \mu$ the queue remains above x for at least $E\{L\} - x/(\lambda_{\rm on} - \mu)$ during the long on-period. Let B denote the length of the n-block. We have

$$E\{B\} = n\left(\frac{1}{\lambda_{\rm off}} + \frac{k\alpha}{\alpha - 1}\right) = \Theta(x^\alpha)$$

where $E\{\tau_{\rm on}\} = k\alpha/(\alpha - 1)$. For large x, we estimate a lower bound using

$$\Pr\{Q(\infty) > x\} \gtrsim \frac{E\{L\} - y}{E\{B\}} = \Theta(x^{1-\alpha})$$

The queue tail decays polynomially and heavy-tailedness is preserved. The preceding is not a proof but an intuitive illustration using elementary arguments to show why the queue would find itself overcrowded for $\Theta(x^{1-\alpha})$ fraction during a $\Theta(x^\alpha)$-length period. Queueing with long-range dependent input is discussed in [19, 83, 85].

An upper bound of $\Pr\{Q(\infty) > x\}$ may be gleamed from the following closure property satisfied by heavy-tailed random variables: for large x,

$$\Pr\{\tau_{\rm on}(1) + \cdots + \tau_{\rm on}(n) > x\} \sim n \Pr\{\tau_{\rm on} > x\}. \tag{5}$$

The most likely time scale at which n heavy-tailed on-periods in a n-block would collude to cause the queue to fill beyond x is $n \sim cx^\alpha$. Moreover, (5) implies $\Pr\{\tau_{\rm on}(1) + \cdots + \tau_{\rm on}(n) > x\} \sim \Pr\{\max\{\tau_{\rm on}(1), \ldots, \tau_{\rm on}(n)\} > x\}$ which indicates that overcrowding occurs "suddenly": at time scale $O(x^\alpha)$ the typical picture is that of a single, long on-period dominating the queue dynamics with respect to storage level x. Hence, $\Pr\{Q(\infty) > x\}$ is upper bounded by a function that is $O(x^{1-\alpha})$ for large x. In the case of exponential on-period $\Pr\{\tau_{\rm on} \le x\} = 1 - e^{\kappa x}$, an analogous lower bound argument with $n = e^{\kappa x}$ yields a $\Theta(e^{-\kappa x})$ bound since the conditional expectation of the exponential distribution is constant. For the upper bound, $\Pr\{\tau_{\rm on}(1) + \cdots + \tau_{\rm on}(n) > x\}$ is exponentially small with a corresponding exponential upper bound on the queue tail.

3.3.2 Finite time scale effects

The arguments used for bounding $\Pr\{Q(\infty) > x\}$ required that x be large, i.e., $x \to \infty$. Queueing analyses with LRD input that provide rigorous estimates on equilibrium tail probability are asymptotic in nature, a handicap when it comes to resource dimensioning and computing buffer overflow for practical systems with small x. Imposing finite storage capacity, $S < \infty$, complicates matters—the resulting queue process has two reflecting barriers—and drawing definitive conclusions on queueing delay and packet loss rate in finitary systems is difficult. Long-range dependent traffic, when compared with short-range dependent traffic, may or may not affect a higher packet loss rate: it depends on the specifics of the finitary storage system. That a heavy-tailed queue

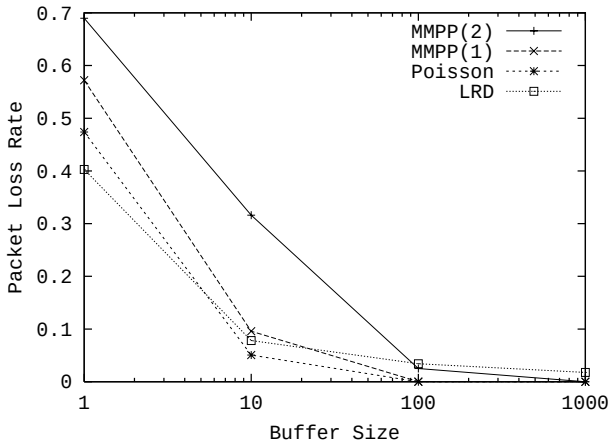


Figure 7: Packet loss rate of finite buffer queue with Poisson, LRD, and MMPP input.

tail is not synonymous with higher packet loss can be seen from Figure 7, which shows packet loss rate as a function of buffer size for a simulated queueing system with three types of traffic input: Poisson, LRD, and MMPP. A 2-state MMPP (Markov Modulated Poisson Process) is a stochastic process generated by a 2-state Markov chain—when the system finds itself in the first state, it behaves as a Poisson process with rate λ_1; when the system is in the second state, it produces Poisson traffic with rate λ_2. By appropriately tuning the transition probabilities along with λ_1 and λ_2, varying degrees of short-range dependent burstiness can be introduced. MMPP(2) denotes MMPP traffic whose parameters have been set to make it significantly burstier than MMPP(1). LRD denotes an aggregated on-off process (cf. Section 3.2.1, Figure 2) composed of 32 independent on-off sources each with tail index $\alpha = 1.1$. All four traffic processes—Poisson, LRD, MMPP(1), and MMPP(2)—possess the same traffic rate. Figure 7 shows that when buffer capacity is 1, LRD traffic achieves the lowest packet loss rate, followed by Poisson, MMPP(1), and MMPP(2). When buffer size is increased to 10, Poisson traffic has the lowest loss rate with LRD second, followed by MMPP(1) and MMPP(2). As buffer capacity increases to 100, another inversion takes place where LRD overtakes both MMPP(1) and MMPP(2) resulting in the highest packet loss rate. At buffer size 1,000, only LRD traffic suffers nonnegligible loss. Thus we observe a transition in the packet loss order

$$\text{LRD} \preccurlyeq \text{Poisson} \preccurlyeq \text{MMPP} \quad \longmapsto \quad \text{Poisson} \preccurlyeq \text{MMPP} \preccurlyeq \text{LRD}$$

as buffer size is increased from small to large. The reason for this is: when buffer capacity is small, variability and correlation nascent in short-range dependent traffic is sufficient to dominate queue dynamics with respect to buffer overflow; only when buffer capacity is large does long-range dependence have an advantage at overflowing the queue and exerting a deciding influence on packet loss rate via-à-vis short-range dependence. In the latter, the required collusion to cause overcrowding is exponentially rare and all-too-brief when it happens.

3.3.3 Self-similar burstiness and jitter

The queueing discussion showed that long-range dependence in self-similar traffic does not necessarily lead to amplified packet loss rate. Although packet loss rate, a first-order performance measure, is a primary yardstick in Internet performance evaluation, another important criterion is

"jitter"—a generic term for second-order performance statistics—that captures variability in the packet loss or packet delay process. In multimedia communication, it is not only needed that the average packet loss be small but that the loss pattern be dispersed so that their effect may be masked by the human perceptual system or through traffic control. Masking by traffic control can be affected through forward error correction (FEC), i.e., channel coding, where redundant information is transmitted to offset damage that may occur during travel. At the granularity of documents or video frames where a single object is split into several packets, encoder and decoder functions exist [113, 115] that satisfy the k-out-of-N property: k information packets are transformed into $N = k + h$ encoded packets—h represents the degree of redundancy—and a receiver is able to reconstruct the original content as long as no more than h packets, whichever they may be, are lost in the network. FEC is especially relevant in real-time applications over long distances where retransmission based error correction may not be an option. For a FEC-protected traffic stream, successful recovery is impeded by jitter in the form of bursty or clustered packet loss as this can lead to too many losses within a block.

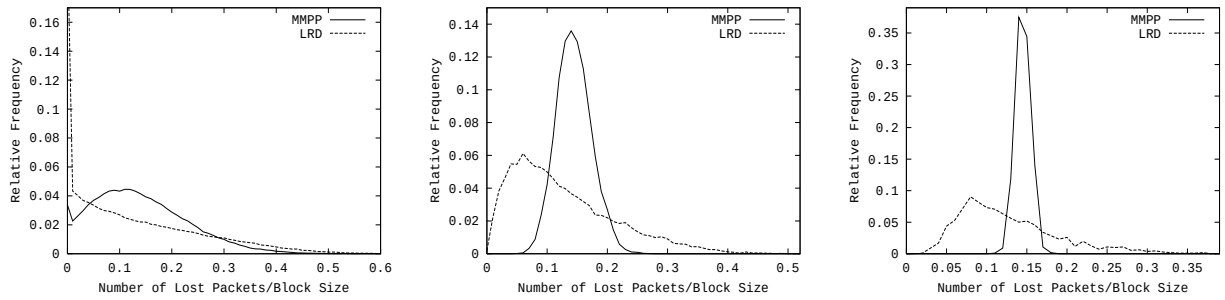


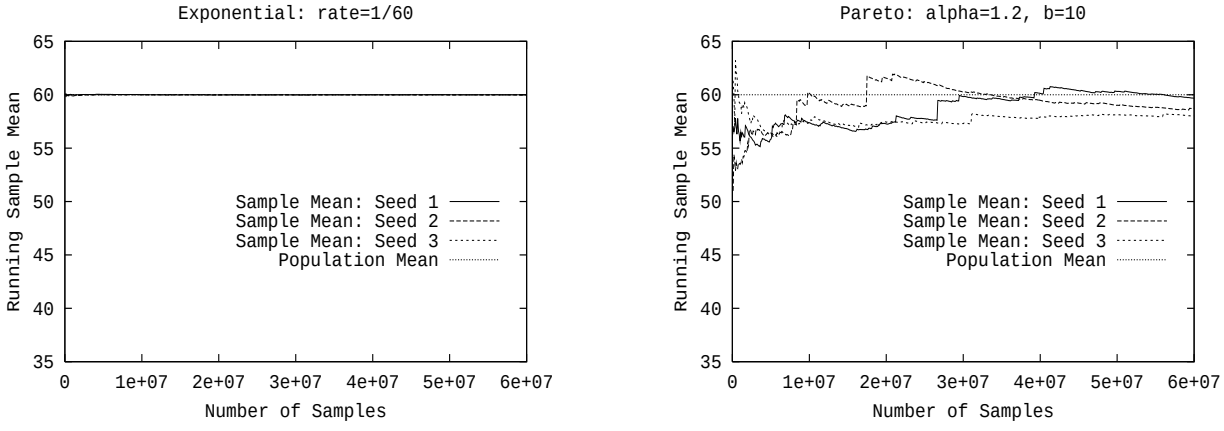
Figure 8: Second-order performance: Block loss probability comparison between LRD and MMPP traffic. (a) Block size $N = 100$. (b) $N = 1,000$. (c) $N = 10,000$.

Given traffic $X(t)$ and block size N, the block loss process induced by feeding $X(t)$ into a finite buffer queue is a sequence of random variables $L_N(i)$, $i \in \mathbb{Z}_+$, where $L_N(i) \in \{0, 1, \ldots, N\}$ denotes the number of packet losses suffered in the i'th block. $L_N(\infty)/N \in [0, 1]$ denotes normalized block loss $L_N(i)$ in steady-state as $i \to \infty$. Figure 8 shows normalized block loss distribution when LRD and MMPP traffic are fed into a queue of buffer size 10. The packet loss rate experienced by MMPP is higher than that of LRD, and we interprete buffer overflow with respect to three different block sizes $N = 100, 1,000, 10,000$. Figure 8(b) shows normalized block loss distribution for $N = 1,000$. Even though MMPP's loss rate is higher, LRD's block loss distribution has a wider spread and variance, incurring up to 40% packet loss within some blocks. MMPP's block loss, on the other hand, is bounded by 23%. Assuming the input is already FEC-encoded with redundancy h, $\Pr\{L_N(\infty)/N > h/N\}$ is the probability that the k-out-of-N property is violated and resultant decoding unsuccessful. Thus, LRD's heavier block loss tail, $\Pr\{L_N(\infty)/N > x\}$, $x \in [0, 1]$, implies that FEC performance significantly degrades for LRD traffic when compared to MMPP traffic despite the former's smaller loss rate. The impact of self-similar burstiness on second-order performance is distinct from that of first-order performance. It is affected by block size N which exerts a similar influence as buffer capacity on block loss performance. Figures 8(a)

and 8(c) show that the relative difference in block loss variance between LRD and MMPP becomes pronounced for $N = 10,000$, whereas for $N = 100$ the difference becomes dampened. For smaller N, the block loss distributions of LRD and MMPP grow close with their tails eventually stretching to 100%. For $N = 1$, $\Pr\{L_1(\infty) = 1\} = 1 - \Pr\{L_1(\infty) = 0\}$ is the loss rate. At the opposite extreme, for large N the distribution of $L_N(\infty)/N$ becomes concentrated around the loss rate.

3.3.4 Sampling and slow convergence

Given the instrumental role played by heavy-tailedness in self-similar network traffic, sampling from heavy-tailed distributions is relevant for network simulation including artificial workload generation. A canonical example is the comparison of queueing performance between short-range dependent and long-range dependent traffic where the latter may involve on-off or $M/G/\infty$ traffic. A key requirement, for comparability, is that the average traffic intensity of the input be the same so that observed differences in loss performance can be attributed to the correlation structure of the input, not sampling induced discrepancy and bias in the actual traffic rate. Figure 9(a) shows the running sample mean for 60×10^6 samples drawn from an exponential distribution with rate



(a) (b)

Figure 9: Sampling and convergence. (a) Running sample mean for exponential distribution with rate 1/60. (b) Running sample mean for Pareto distribution with $\alpha = 1.2$ and $b = 10$.

1/60 for three different random seeds. Figure 9(b) shows corresponding sample means for a Pareto distribution with parameters $\alpha = 1.2$ and $b = 10$. Both distributions have the same population mean 60, but their convergence properties are markedly different. Whereas the sample mean of the exponential distribution converges rapidly and hugs the population mean after 6,000 samples, the sample mean of the Pareto distribution requires at least 30 million samples to approximate the population mean with 5% accuracy. The polynomial tail amplifies the contribution of large values to the expectation, and its realization requires polynomially many observations with respect to the inverse of the accuracy parameter but greater than exponentially many as a function of the tail index.

To see this, let Z be Pareto with tail index α and location parameter b. The probability density function of Z is given by $f(x) = \alpha b^\alpha x^{-(1+\alpha)}$. Recall that Z has finite mean but infinite variance

for $1 < \alpha < 2$. We split the expectation, $E\{Z\} = b\alpha/(\alpha - 1)$, into two parts separated by y, $E\{Z\} = \int_b^y xf(x)dx + \int_y^\infty xf(x)dx$, and consider the relative contribution of the tail in achieving an accuracy level ε,

$$\frac{\int_y^\infty xf(x)dx}{E\{Z\}} = \frac{\alpha - 1}{b\alpha} \int_b^y x\alpha b^\alpha x^{-(1+\alpha)} dx = \left(\frac{b}{y}\right)^{\alpha-1} < \varepsilon.$$

We have $y_0 = b(1/\varepsilon)^{1/(\alpha-1)} < y$, and since $\Pr\{Z > y_0\} = \left(\frac{b}{y_0}\right)^\alpha = \varepsilon^{\frac{\alpha}{\alpha-1}}$, we require $N_0 = \varepsilon^{-\frac{\alpha}{\alpha-1}}$ samples to expect a value greater than y_0. The number of samples is polynomial in the inverse of the accuracy parameter, $1/\varepsilon$, but grows faster than exponentially in α as it approaches 1. For $\alpha = 1.2$ and $\varepsilon = 0.05$, N_0 yields 64 million samples. In the case of the exponential distribution, the contribution of the tail in the expectation is exponentially small in y, which implies that y need only be logarithmically large in $1/\varepsilon$. However, due to the exponential smallness of the underlying probability distribution, the number of samples required remains a polynomial function of $1/\varepsilon$, albeit with a small exponent that does not depend on the rate parameter.

Returning to Figure 9(b), we observe sudden jumps in the sample mean at 8 million, 17 million, and 26 million samples which illustrates that the sum of heavy-tailed random variables is dominated by the maximum "outlier," a consequence of property (5) (cf. Section 3.3.1). The slow convergence associated with heavy-tailed random variables exacts a heavy toll on sampling, and speed-up methods that admit shortcuts—the subject of rare event simulation [63]—are needed. For example, importance sampling, a technique that modifies an underlying probability measure to pump up the likelihood of rare events, is aimed at achieving both small variance and relative error. Importance sampling with an exponential change of measure has been applied, with some success, to light-tailed distributions, but encounters difficulties when extending to heavy-tailed distributions. On the other hand, a change of measure that selects distributions with heavier tails shows initial promise [6]. Additional discussion on sampling and fast simulation with heavy-tailed and long-range dependent input can be found in [38, 57].

3.4 Traffic control: What can be done about it?

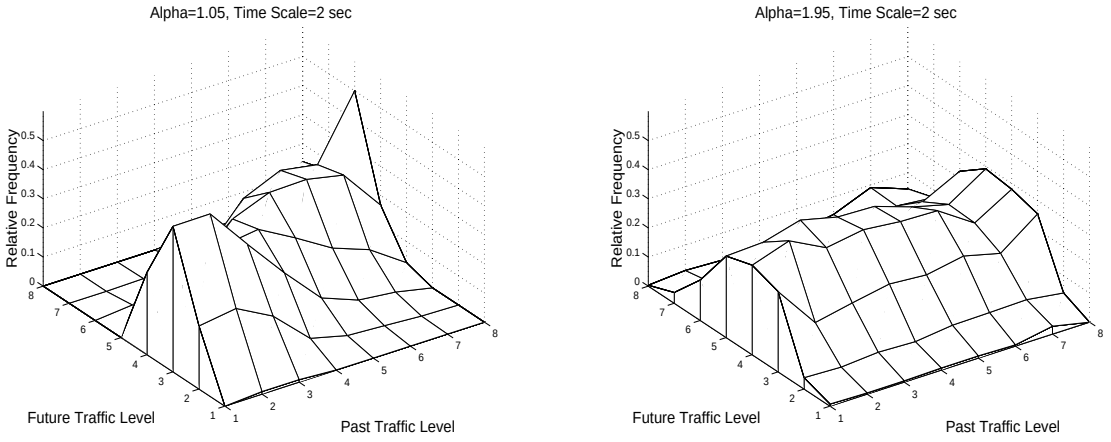
3.4.1 Large time scale predictability

Given that long-range dependence in self-similar traffic is caused by heavy-tailed session durations which, in turn, arise from heavy-tailed file sizes, the predictability inherent in heavy-tailed distributions—manifested as long-term correlation in network traffic—may be harnessed for traffic control purposes. To see why heavy-tailedness implies predictability, consider a heavy-tailed random variable Z with index α. We are interested in the conditional probability $\Pr\{Z > x+y \,|\, Z > y\}$ which captures the likelihood of predicting the "future" value of Z given its "past" value y. For example, if Z were the session duration of an IP flow that is being tracked at a router, $\Pr\{Z > x + y \,|\, Z > y\}$ would give the probability that a session that has lasted for y seconds will continue for at least another x seconds. We have

$$\Pr\{Z > x + y \,|\, Z > y\} = \frac{\Pr\{Z > x + y\}}{\Pr\{Z > y\}} \sim \left(\frac{y}{y + x}\right)^\alpha. \tag{6}$$

$\Pr\{Z > x+y \,|\, Z > y\} \to 1$ as $y \to \infty$, which implies that conditioning the future on an ever longer past brings about certitude. In contrast, for the memoryless exponential distribution, conditioning on the past has no bearing on the future. The expected future duration is given by the conditional

expectation $E\{Z \mid Z > y\} = y\alpha/(\alpha - 1)$ (cf. Section 3.3.1). Thus persistence into the future is proportional to persistence in the past, with α close to 1 amplifying the predictable time horizon. The conditional variance, $\mathrm{Var}\{Z \mid Z > y\}$, is unbounded for $0 < \alpha < 2$. In the case of aggregate traffic, for example $M/G/\infty$ input, aggregate traffic at time scale $O(t^\alpha)$ is dominated by a single long session of duration $O(t)$. Relative signal strength, $t^{1-\alpha}$, decays slowly which renders long-term positive correlation generated by heavy-tailedness detectable in the presence of short-term fluctuations.



(a) (b)

Figure 10: Long-range correlation and predictability. (a) Conditional probability for $\alpha = 1.05$ traffic. (b) Conditional probability for $\alpha = 1.95$ traffic.

Figure 10 shows predictability in aggregate network traffic generated by multiple TCP connections that share a common bottleneck link. We consider two scenarios: one, files transported by TCP is Pareto with $\alpha = 1.05$, and two, file size distribution is Pareto with $\alpha = 1.95$. A time series representing traffic measurement is partitioned into 2-second time windows. The traffic volume within a time window is quantized into 8 levels where 1 is low and 8 is high. Let S_L and S_R denote random variables representing the quantized traffic volume at two consecutive time windows. Figure 10(a) shows the estimated conditional probability $\Pr\{S_R \mid S_L = h\}$ as a function of traffic level $h \in \{1, 2, \ldots, 8\}$ for TCP traffic with $\alpha = 1.05$. At each traffic level, we observe a skewed distribution: when S_L is high (low), it is likely that S_R is high (low). Positive correlation between S_L and S_R leads to a diagonally shifting distribution peak in the 3-dimensional plot. Figure 10(b) shows the corresponding results for $\alpha = 1.95$. We observe a stark difference: the shape of $\Pr\{S_R \mid S_L = h\}$ does not depend on S_L, i.e., $\Pr\{S_R \mid S_L = h\} \approx \Pr\{S_R\}$ for all h. Short-range dependent traffic possess a conditional probability profile similar to Figure 10(b) at the 2-second time scale. At time scales 1–2 orders of magnitude smaller, however, short-range dependent traffic exhibit conditional probability plots similar to that of Figure 10(a). That is, at sufficiently small time scales, short-range dependent traffic is predictable.

3.4.2 Delay-bandwidth product and reactive penalty

In multiple time scale traffic control [105, 132], information present at multiple time scales is engaged to affect traffic control. Feedback traffic control, of which TCP is an instance, acts at the time scale of feedback latency, also called round-trip time (RTT). Network state information from RTT time units in the past is used in the present to enact control actions aimed at achieving improved performance. In dynamic network environments, the larger the RTT the more outdated the information, and the less effective the control action ("the train has left the station"). The reactive cost is especially pronounced in high-speed wide-area networks (WANs) where many packets are simultaneously in transit, and damage ensuing from delayed reaction can be significant. For example, in coast-to-coast transmissions, RTT is in the tens of milliseconds and individual broadband access speed can exceed 1 Mbps (million-bits-per-second). In satellite networks, bandwidth is relatively small but two-way latency reaches 500 msec.

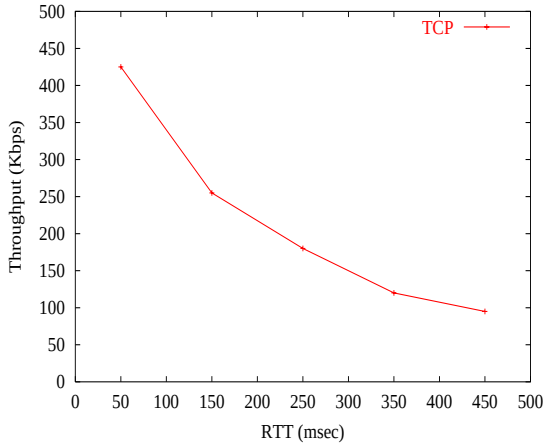


Figure 11: TCP performance as a function of RTT.

Figure 11 shows the diminishing throughput of TCP as RTT increases. Exposure of feedback traffic control to the performance limitation imposed by large delay-bandwidth product networks is an inherent problem. In open-loop traffic control, resources are reserved in advance to protect a traffic flow from uncertainties of future network state in a FCFS (first-come-first-served) shared network environment. When performance guarantees are required, open-loop control is unavoidable. However, it carries its own cost of having to know the characteristics of the traffic flow—not always easy to do—and potential resource underutilization stemming from per-flow reservation near the peak, as opposed to average, data rate. In multiple time scale traffic control, the goal is to exploit long-range predictability in closed-loop traffic control to mitigate the outdatedness of feedback information at the time scale of RTT.

3.4.3 Workload-sensitive traffic control

Two issues need to be addressed when engaging long-range correlation for traffic control: prediction of future traffic and utilization of this information for traffic control. At time t, the average future

traffic level at time horizon $t^* > t$, given by

$$\bar{X}(t, t^*) = \frac{1}{t^* - t} \sum_{s=t}^{t^*} X(s),$$

may be predicted using past observations $X(t-1), X(t-2), \ldots, X(t-t_*)$ extending t_* time units into the past. The prediction error $E\{(\bar{X}(t,t^*) - \hat{X}(t,t^*))^2 \mid X(t-1), \ldots, X(t-t_*)\}$ of the best linear unbiased estimator (BLUE) is known in the asymptotic case (i.e., as $t_* \to \infty$) [16]; the finitary case, however, is more difficult [17]. A simpler, but suboptimal, prediction method uses average past traffic level $\bar{X}(t_*, t-1)$ to predict the future: $E\{\bar{X}(t,t^*) \mid \bar{X}(t_*, t-1)\}$. A quantized variant of the conditional expectation predictor has been employed in [105, 132, 133]. Since the aggregate traffic level at a bottleneck router is not directly observable by the sender, active or passive probing must be used—unless routers are enabled to convey their state information directly, which has its own problems—to estimate contention level. In active probing, probe packets are transmitted to ascertain network state from the packets' delay and loss characteristics observed at the receiver. In passive probing, the characteristics of transmitted application traffic is used to infer network state. This eliminates additional messaging overhead, albeit at the cost of reduced accuracy. The passive probing method used in [105, 132] utilizes the output behavior of TCP, observable at the sender, to infer the contention level on a bottleneck path. This method relies on the tracking ability of feedback congestion control, inclusive TCP, whose output behavior can be shown to be negatively correlated to the contention level.

Assuming the average future traffic level is known, how should this information be utilized to effect improved traffic control? Long-term state information, by definition, is slowly varying—short-term fluctuations may be missed—and the predicted information is probabilistic: with some likelihood predictions are wrong. Hence, at a minimum, a large time scale control action should not do more harm than good. In proactive congestion control, long-term prediction is used to modulate bandwidth consumption behavior at the time scale of RTT: aggressive when the outlook is good and conservative if the outlook is bad. The net effect is improved throughput, achieved by desensitizing control actions against short-term fluctuations while increasing awareness of persistent state changes. By reducing futile reactions to transient events that, by definition, are fleeting and outdated by the time control actions take effect, throughput degradation may be alleviated.

In the context of TCP congestion control [69], long-term prediction may be utilized in two complementary ways. When network conditions are favourable—i.e., available bandwidth is high—TCP should aggressively soak up unused bandwidth as the opportunity cost for not doing so is commensurately high. This may be accomplished by increasing the linear rate at which TCP opens up its throttle. Eventually TCP reaches its maximum throttle or the increased sending rate causes losses at bottleneck routers. In the latter, upon detecting potential packet loss—the TCP sender uses lack of timely acknowledgement from the TCP receiver as an indicator of loss—TCP clamps down on the throttle by a multiplicative factor of 1/2. Consecutive multiplicative clamp-down leads to exponential back-off which, in general, is needed to achieve stability. During periods when available bandwidth is plentiful, however, back-off need not be as drastic. Conversely, when the overall contention level is high, a more conservative bandwidth consumption behavior may be undertaken. A specific form that modulates the slope during linear increase in illustrated in Figure 12. Stability of multiple time scale TCP holds as long as TCP's feedback congestion control is stable and the time scale at which long-term control parameter modulation is undertaken significantly exceeds the time scale of RTT. If both conditions are satisfied, time scale separation assures that TCP feedback control reaches equilibrium during a successive pseudo-stationary time window at which

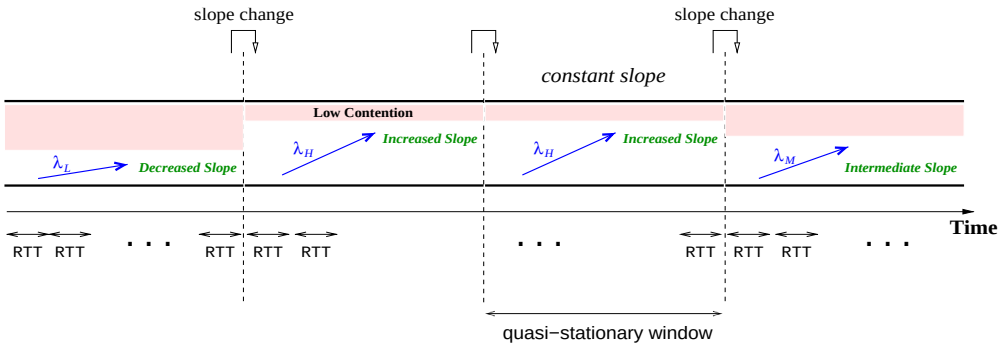


Figure 12: Selective slope control: TCP's linear increase slope is modulated as a function of long-term network state.

control parameters are held constant. In the long run, TCP moves from equilibrium to equilibrium across successive pseudo-stationary time windows.

Figure 13 shows the performance gain achieved by TCP when it is augmented to utilize long-term predicted contention to modulate its linear increase-exponential decrease behavior. We observe that the multiple time scale version of TCP, TCP-MT [105], improves on the throughput of TCP, where throughput gain (%) amplifies with RTT. The quantitative gain depends on the underlying TCP flavor—e.g., TCP Reno, NewReno, Vegas—and specific network conditions. Performance gains up to 60% have been observed in prototype systems. Similar results hold for rate-based congestion control which are employed in UDP based traffic streaming [132]. Feedback congestion control modulation using long-term information applies to long-lived flows. For short-lived flows, feedback is relevant for error and packet loss recovery. Predictability from heavy-tailedness admits on-line classification of short-lived and long-lived flows so that long-term control actions can be affected to long-lived sessions where they matter. The same holds for traffic shaping and admission control: the few elephants that consume much of the Internet bandwidth must be reigned in to make a difference. The mice, collectively, exert only a small impact on system performance.

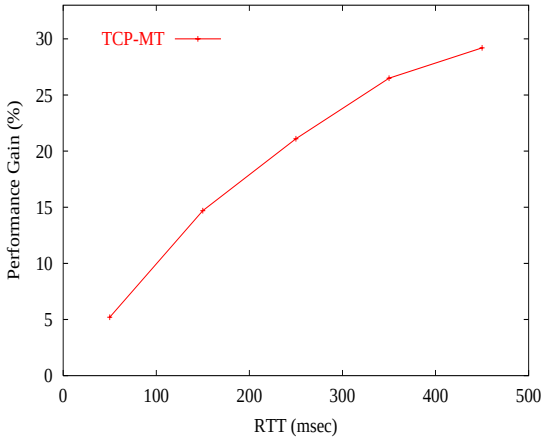


Figure 13: Performance gain of multiple time scale TCP: TCP-MT.

Workload-sensitive traffic control has been applied for real-time video/audio transport using adaptive forward error correction (AFEC), where redundancy is adaptively injected to protect

against packet loss without necessitating retransmission [106, 107]. Multiple time scale AFEC, AFEC-MT, shields AFEC against transient fluctuations, thereby increasing the recovery rate—i.e., correct decoding of video/audio frames—at the receiver [133]. In [100] router-aided rate control for self-similar traffic is explored. Sampling based prediction and scheduling is studied in [144]. Heavy-tailedness has been exploited for dynamic load balancing of UNIX processes where process lifetimes are observed to be heavy-tailed [61], enhancing routing stability through long-lived and short-lived IP flow separation where routing updates are desensitized against short-lived flows [124], and process scheduling where preference is given to short-lived processes [37].

3.5 Discussion

The global Internet operates primarily as a distributed client/server system, where the bulk of events entail fetching files of various types—Web page, image, video, and software—from remote sites. Heavy-tailedness of file size distribution is a structural property of distributed systems, an empirical "law" on par with Poisson session arrival and locality of reference. Memoryless inter-session arrival has enabled Markovian analysis of real-world systems, including capacity planning in telephony whose principles date back to Erlang's pioneering work [49]. Locality of reference is at the heart of caching, a resource management technique without which economic information processing would be severely impeded. Collectively, the three laws form the cornerstones of effective system design and engineering, with heavy-tailed file size being the junior member.

Heavy-tailed files are responsible for generating self-similar network traffic, an emergent trait of the Internet that transcends its phenomenological curiosity. That is, self-similar burstiness has repercussions to network performance, planning, and control. Heavy-tailedness is a robust property in the sense of being a conserved property: it manifests itself as long-range dependence when channeled across a network, surfaces as heavy-tailed queueing delay when fed into a router's buffer, and translates into long-term predictability when harnessed for traffic control. Heavy-tailed file size is like an invariant that morphs into other heavy-tailed network phenomena but is not readily suppressed.

Fractal properties of network traffic in the form of $1/f$ noise and chaotic dynamics of TCP's nonlinear feedback control are not unrelated (cf. Section 3.2.3), but of secondary import to Internet engineering. The latter provides grounding in real-world considerations that curb chaos-centric interpretations of the Internet when viewed as a complex system. Self-similar traffic is a representative example where a "physics of network traffic" is established spanning measurement, network mechanics, and mathematical modeling. A strength of the Internet is that it comes with concrete application driven issues where complex systems notions exert a tangible, pragmatic influence. These range from structural causes of correlation at-a-distance, to slow convergence and equilibria, to prediction and control. As an engineered physical system, the Internet affords precise quantitative measurement allowing testing of theories with respect to cause and effect that may involve complex what-if scenarios whose scope goes beyond those feasible in biological, social, and natural physical systems.

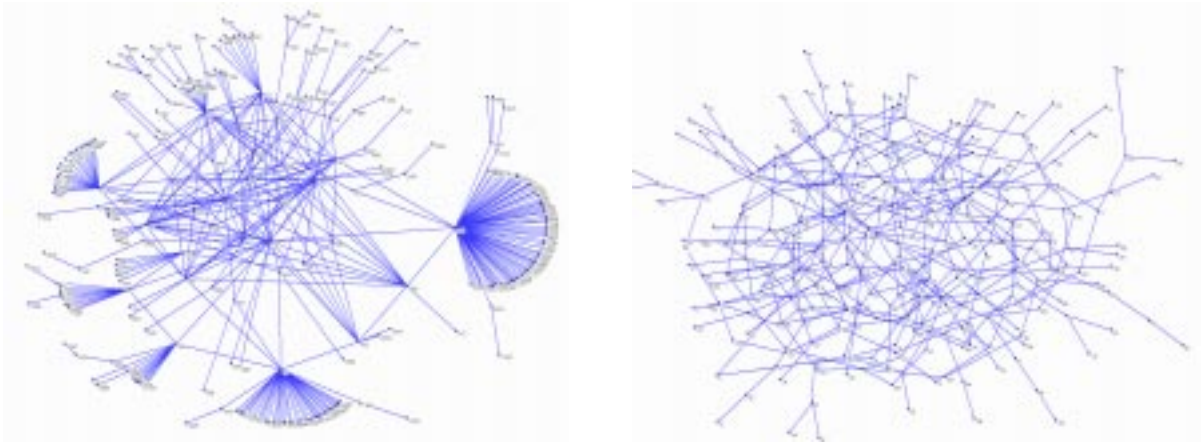
The origin of self-similar traffic traces back to heavy-tailedness of file sizes. Venturing a step further, one may ask: why are files heavy-tailed? This is not an uninteresting question, but one that may not have a satisfactory scientific answer. Why are customer inter-arrival times approximately memoryless? Why is locality of reference so prevalent? It is not difficult to advance philosophical ruminations, but perhaps few that lead to scientific rigor or productive consequence. We leave the empirical laws as axioms.

4 Power-Law Network Topology

4.1 What is power-law network topology?

In the late 1990s, empirical measurements showed that a number of real-world graphs including World Wide Web graphs [12, 22, 79], Internet domain networks [50], call graphs [1, 3], and certain biological and social networks [71, 97, 114] exhibited an unexpected connectivity pattern: the neighborhood size of nodes was quite variable, following a power-law distribution. As with heavy-tailed file sizes in self-similar traffic, this implied that most nodes are small, i.e., have few neighbors, but a few are very large. The reason these findings were surprising is that they did not fit existing models—perhaps even conceptions—of how networks are connected, examplified by random graphs whose size distribution has an exponentially decreasing tail. This does not mean that random graphs were regarded as good models of engineered and natural networks. In the field of combinatorial optimization there has been a perennial search for realistic benchmark graphs on which the average performance, as opposed to worst-case performance, of optimization algorithms could be evaluated. In spite of this recognized deficiency and need, pinning down the essential features of real-world networks proved elusive.

Figure 14(a) shows a 300-node Internet domain graph where nodes are administrative domains and edges denote peering relations between domains. Administrative domains, also called autonomous systems (ASes), are identified by 16-bit numbers (e.g., Purdue University has AS number 17). A typical peering relation is one where one domain is a customer of another, the provider. An AS is a logical entity that need not be geographically localized: for example, a major transit domain that provides connectivity service to other domains may have points-of-presence (POPs) across multiple continents where access routers are deployed. Access routers are connected by backbone networks internal to an AS. A link between two ASes means that there is at least one pair of border routers belonging to the respective domains that are directly connected, sometimes through a multi-party connection called an exchange. In an Internet domain graph, these and other details are ignored. Figure 14(b) shows a 300-node random graph with the same edge density



(a) (b)

Figure 14: Power-law vs. random network topology. (a) 300-node Internet AS graph. (b) 300-node random graph with the same edge density.

as the 300-node Internet domain graph. In $G_{n,p}$ random graphs—n is the number nodes and p a probability—an instance of a random graph is generated by selecting each edge independently with probability p. Thus $G_{n,p}$ assigns a probability distribution over the finite sample space of all n-node graphs. If $p = 1/2$, all n-node graphs are equally likely. A graph is sparse if the number of edges is sub-quadratic, i.e., significantly less than the total number of edges $\binom{n}{2}$. Power-law graphs turn out to be sparse. The connectivity structure of the two graphs shown in Figure 14 is markedly different. Whereas the random graph is homogeneous and looks locally about the same, the power-law graph possesses "hubs" of varying sizes—some very large—that are connected through a "backbone." In a random graph, the neighborhood size of a node, called its degree, is concentrated around the mean np, a consequence of LLN. A random graph is an approximately regular graph, where a graph is k-regular if all its nodes have the same degree k. The graph in Figure 14(a) is, in this sense, highly irregular and not a typical instance of $G_{n,p}$.

4.2 Power-law random graphs

4.2.1 Power-law degree distribution

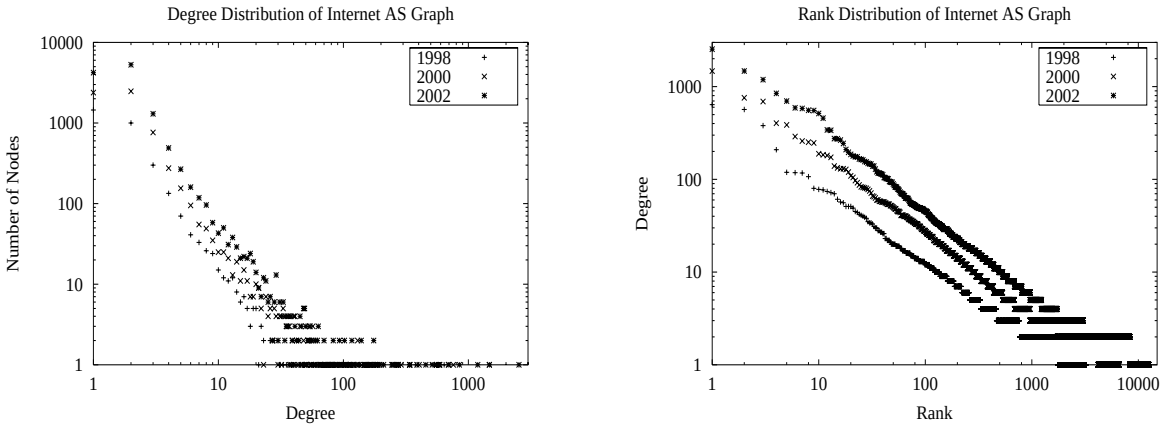
Random graphs, pioneered by Erdös and Rényi [47], possess a binomial degree distribution, thus making nodes with many neighbors exponentially rare. In a power-law graph, a polynomial degree distribution is postulated,

$$\Pr\{\deg(u) = k\} \propto k^{-\beta},$$

where $\beta > 0$ is an exponent that depends on the application area whence a graph comes from. For example, in a number of empirical measurement graphs, including Internet AS graphs, $2 < \beta < 3$. Figure 15(a) shows the degree distribution, on log-log scale, for Internet AS topologies from Oregon Route-Views measurement data that are based on route table dumps [96, 135]. For a wide range of degree values—99% of nodes have degrees less than 50 and 95% have degrees less than 10—we observe a linear relationship with β slightly exceeding 2. For large-degree nodes whose small relative frequency is drowned out in a degree distribution plot, a rank distribution plot can be used where nodes are sorted in nonincreasing order of degree and the resultant rank is related to degree [50]. For example, in a 2002 Internet AS graph, rank 1 is occupied by AS 701—UUNET, a tier-1 transit provider—which has degree 2,538. By plotting rank versus degree, focus is shifted to high degree nodes at the expense of low degree nodes where an overwhelming majority have degree 1 or 2. Figure 15(b) shows the log-log rank distribution for the same data set. We observe a linear fit, consistent with a power-law relation, with slope a little less than 1. For a number of technical reasons, Internet AS topologies inferred from measurement data provide a partial and inexact view of domain-level connectivity. Although power-law degree distribution is a robust phenomenon observed across AS topologies obtained from different measurement sources, when drawing conclusions on the implications of power-law connectivity, it is imperative to consider the limitations of the measurement data and inferred topologies as well as application-specific idiosyncracies.

4.2.2 Power-law molecular stew

From a structural perspective, we may interpret a degree sequence that specifies the degree of nodes in a graph in nondecreasing order as a set of n "molecules" where molecule i has w_i bonds. Figure 16 illustrates the ingredients of a power-law "molecular stew" which may be stirred to generate higher order structures. It is reminiscent of simulated annealing, albeit in multi-dimensional space



(a) (b)

Figure 15: Power-law degree distribution of Internet AS graph. (a) Log-log degree distribution. (b) Log-log rank distribution.

as 3-D is too restrictive for effective stirring. High-degree nodes, by virtue of their abundant links, are likely to form connections with each other—not always directly—yielding a dense connected component comprising a skeleton backbone. Since high-degree nodes are few, many unfilled bonds will dangle from the backbone. The bulk of dangling bonds must link up with low-degree nodes, the most common building blocks in the power-law ingredient pool. This yields locally star-like shapes—reminiscent of Asian fans—a characteristic feature of power-law graph drawings (see, e.g., Figure 14(a)). The bondings, thus far, were of a "large-large" and "large-small" kind. The remaining molecules—a minority of small and intermediate nodes—may contribute to the final structure two-fold: "small-small" pairings lead to elongated branches sticking out from the backbone, and others further the connectivity of the skeleton backbone. With respect to the power-law exponent β, the larger the exponent the fewer the presence of high-degree nodes in the molecular stew, which diminishes the effect of large-large and large-small bondings. This may lead to fewer and smaller fans, and a less intertwined backbone.

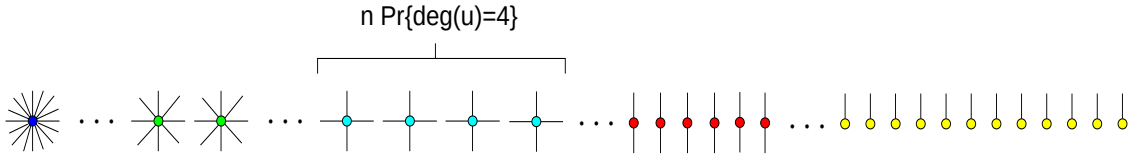


Figure 16: Molecular stew with ingredients determined by power-law degree sequence.

4.2.3 Power-law random graph model

Fan and Lu [32, 84] studied a random graph model based on expected degree sequences that may be viewed as a generalization of $G_{n,p}$. Given an expected degree sequence $\mathbf{w} = (w_1, w_2, \ldots, w_n)$ where w_i denotes the expected degree of node i, an edge between nodes i and j is independently

selected with probability p_{ij} proportional to the product of the weights, $p_{ij} = w_i w_j / \sum_{\ell=1}^{n} w_\ell$. It is easily verified that the expected degree of node i is w_i. $G_{n,p}$ may be viewed as the special case where $\mathbf{w} = (np, np, \ldots, np)$. The condition $\max_i w_i^2 < \sum_j w_j$ is placed which assures that p_{ij} is less than 1 and the degree sequence $\mathbf{w}$ is graphical, i.e., there exists a graph with the given degree sequence. A related approach based on exact degree sequences, called configuration model [92], has been investigated by Aiello et al. [3]. A key advantage of the expected degree sequence random graph model is its built-in independence. It comes, however, at the cost of defining a graph family whose members satisfy a prescribed degree sequence only on average. Erdös and Gallai [46] provide necessary and sufficient conditions for a degree sequence to be graphical, which also admits an iterative procedure for constructing a graph instance. We may generate other instances of graphs with a given degree sequence by constructing a Markov chain that performs a pair-wise edge switching operation: given two disjoint edges, the disconnected end points are joined and the original edges deleted. Clearly the local graph perturbation preserves degree sequence. In the configuration model, we may produce power-law multi-graphs—in a multi-graph two or more edges are allowed between a pair of nodes—by making w_i copies of each node i (each copy is void of edges), then forming a random matching on the resultant $\sum_\ell w_\ell$ nodes where nodes are randomly paired. When the w_i copies of node i are collapsed back into a single node, node i may share two or more links with another node j. Rigorous results can be shown for the configuration model with power-law degree sequence under random matching, such as the existence of a giant component and logarithmic bound on the size of smaller components [3]. Little is known about the Markov chain model.

In [31, 32, 84] the basic properties of random graphs with a given expected degree sequence are established. Chung likens random graphs that obey a power-law expected degree sequence with $2 < \beta < 3$ to an octopus. The body of the octopus is a dense core of diameter $O(\log \log n)$ that contains $n^{c/\log \log n}$ nodes. The nodes in the core are large in the sense that their degrees are at least $n^{1/\log \log n}$. The average distance of smaller nodes to the core is $O(\log \log n)$. The octopus has arms that extend to $O(\log n)$ from the core. When $\beta > 3$, the power-law random graph is more expansive with average distance $O(\log n)$. At $\beta = 3$, the average distance is of order $\log n / \log \log n$. Some important features do not depend on the higher-order properties of a degree sequence. For example, the distribution of connected components, including the giant component, depends on the average (expected) degree $\sum_{\ell=1}^{n} w_\ell / n$ but not on the individual make-up of the weights. The small diameter—and even smaller average distance—is indicative of Milgram's small world phenomenon [91]. In social networks where links are defined by acquaintance relationships, six hops suffice to reach presidents and hollywood actors, a phenomenon referred to as "six degrees of separation." Since classical random graphs also possess a logarithmic diameter, the small world phenomenon—highlighted in the influential work of Watts and Strogats [140]—is an important but perhaps not determining feature of empirical networks modeled by power-law graphs. In Section 4.4 we consider more subtle properties with implications to shielding the Internet from network security attacks.

4.2.4 Growth model: preferential attachment

A random graph evolution is a stochastic process that "grows" a graph instance by sequentially adding edges or nodes. It is a tool introduced by Erdös and Rényi [48] to study emergent structural properties of random graphs, including phase transitions at critical edge densities. In [12] a graph evolution for power-law graphs is proposed where a new node is preferentially attached to existing nodes based on their degree: a node is chosen for attachment with probability proportional to its

degree. Thus a node with many connections is likely to get even larger as its chances of winning in the connect-to-the-new-node competition is self-reinforcing. This growth dynamics captures one form of the saying "the rich get richer and the poor get poorer," and may be a causal factor underlying the skewed connectivity of power-law graphs. In [18] it is shown that a formalization of this process leads to a random graph with a power-law degree distribution with exponent $\beta = 3$. In [84] a generalized graph evolution is studied that is able to generate power-law graphs with a tunable exponent.

As a causal explanation of power-law graphs, preferential growth models are intuitively appealing since they embody biases in social dynamics where popularity, reputation, or notoriety have a tendency to become concentrated, at least up to a point. In the context of inter-domain peering, organizational behavior—other things being equal—may sway a new domain to subscribe bandwidth from a large, well-known transit AS due to: a tacit perception of reliability, simply because smaller providers are unknown, a bandwagon effect ("many of our business associates are connecting to UUNET, so should we"), and one-hop reachability to a large customer base. However, these are but a subset of relevant factors and other things are not always equal. For example, a medium size provider, due to political connections, may have POPs at international sites that a larger provider does not. In some instances, a smaller provider may have beaten a larger transit AS to the finish line with respect to new technology upgrade. Time scale is another important factor. Inter-domain peering agreements occur at the time scale of weeks, months and years, during which changes in economic climate, political upheaval, and technology innovation enter into the fray. Nonstationarity has been an issue when modeling self-similar traffic via stationary processes [44]: traffic at noon is different from traffic in the morning. Time-of-day periodicity, however, exists at time scales larger than those relevant for self-similar traffic engineering. For power-law topologies arising in application areas such as inter-domain connectivity, router-level topology, call graphs, and Web graphs, nonstationarity due to exogenous variables may be more problematic when advancing causal explanations of power-law connectivity.

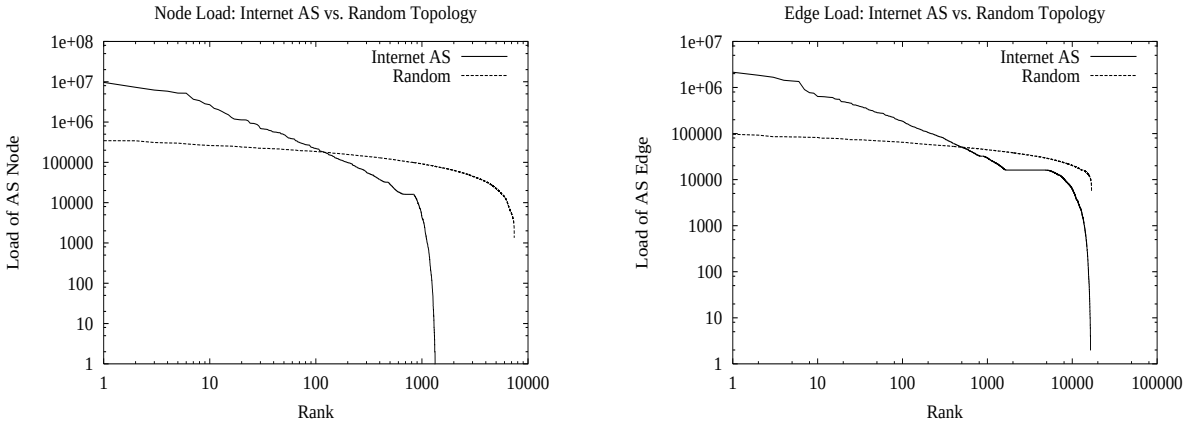
4.3 Performance implications and control

4.3.1 Network flow and load imbalance

In this article, we focus on the implications of power-law connectivity when the underlying graph represents a flow network, as is the case for Internet AS and router graphs. Web graphs, in contrast, are first and foremost information networks where relational structure as captured by various criteria of semantic closeness, including the presence of clusters, admit analysis of massive data sets for effective information retrieval [10, 67]. For example, Google's Web page ranking algorithm [101] uses a popularity index—how many pages refer to a Web page—to determine the importance of search results. Flow and relational structure are intimately related, and some of the points advanced for flow networks have meaningful interpretations in the information retrieval context.

Let us consider a network $G = (V, E)$ where a routing algorithm has determined a path $u \rightsquigarrow v$ between every pair of nodes $u, v \in V$ in the network. Assuming uniform traffic demand, i.e., all source-destination pairs are equally likely, we define the load of a node u, $L(u)$, as the number of paths that traverse through u. Thus node load captures an innate "stress" that is placed on nodes in the flow network—a function of topology and routing—which can translate to hot spots or congestion if the load of a node is large compared to its capacity. An analogous definition holds for edge load $L(e)$, $e \in E$. Figure 17(a) shows the log-log plot of node load as a function of load rank—

nodes are ordered with respect to their load where rank 1 is occupied by a maximal load node—for a 2002 Internet AS graph and a random graph of the same size and edge density. Figure 17(b) shows the corresponding log-log plot of edge load. For both Internet AS and random graphs, we observe an approximately linear regime followed by a sharp fall. The flat region in Figure 17(b)

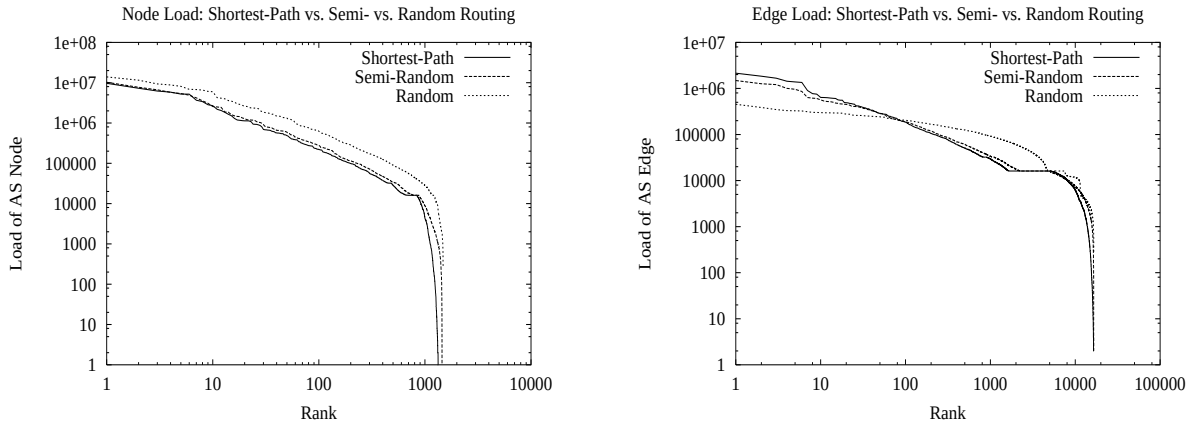


(a) (b)

Figure 17: Comparison of load imbalance between Internet AS and random topology. (a) Node load. (b) Edge load.

toward the tail is due to ties in the rank. A smaller flat region is discernible in Figure 17(a). The main difference between Internet AS and random graphs is that the slope of the log-log plot in the former is steeper than that in the latter. This implies that the load imbalance in the Internet AS topology is more pronounced. In random graphs, routes are dissipated and the resulting load more balanced. Power-law connectivity has a propensity to induce hot spots, skewing the responsibility placed on some nodes and edges over others.

This structural difference between Internet AS graphs and random graphs is insensitive to details in the underlying routing. Figure 18(a) shows the log-log node load distribution of the 2002 Internet AS topology under three different routing: shortest-path, semi-random, and random. In shortest-path routing, a shortest path is computed between source and destination using a destination based version of Dijkstra's algorithm which iteratively constructs a routing tree rooted at the destination. Destination based route construction follows the procedure employed by BGP, Internet's global routing protocol. In semi-random routing, the shortest-path preference is modified such that when a new node is added to the routing tree—the purview of policy in inter-domain routing—instead of choosing a minimal distance node, a random node is selected. The randomly selected node, however, is then attached to a node in the tree that it is closest to, preserving a tendency for inducing short paths. In random routing, both the next node selection and its attachment to the routing tree are done randomly. The average path lengths of shortest-path, semi-random and random routing are 3.64, 3.83 and 6.57, respectively. The corresponding maximum path lengths are 11, 16 and 22. In Figure 18(a) we observe that non-shortest-path routing has little influence on the shape of the node load curve. In power-law networks, imbalances in node load cannot be evened out by routing. The skewed node load distribution is an invariant of the power-law topology. Figure 18(b) shows that fully random route construction—an extreme form of policy diversity—does improve the balance



(a) (b)

Figure 18: Impact of routing on Internet AS load imbalance: shortest-path, semi-random, and random routing. (a) Node load. (b) Edge load.

in edge load. Semi-random routing has little effect on edge load. By forgoing route efficiency, it is possible balance the edge load but not the node load ("all roads still lead to Rome").

The consequences of severe load imbalance on network engineering are: one, a structural propensity for congestion if capacity is not accordingly matched, two, congestion at a few nodes—node or edge failure being a special case where nothing passes through—impacts a significant fraction of the overall traffic demand, and three, load imbalance is an innate property of power-law networks where network engineering exacts only limited influence. A caveat to the above is the assumption of uniform traffic demand which is not reflective of actual Internet demand. The Internet is principally a client/server network where demand is closer to "many-to-few" than "many-to-many." Voice-over-IP (VoIP) and peer-to-peer file sharing applications such as KaZaA promote a flat demand structure that may amplify in the future, but at the present they comprise a minority. When traffic demand is many-to-few, caching becomes an effective network engineering strategy that can keep a significant portion of the traffic demand local. One method for selecting cache proxies is to pick high-degree nodes where replicated files and services are hosted. High-degree nodes will continue to carry a high processing burden, however, their transit burden will be significantly reduced. Topology-aware caching has been considered in [73]. Multicasting is another application that can benefit from topology-aware placement. In multicasting, a spanning tree is constructed that allows broadcasting among a group of users. Multicasting on power-law topologies leads to graph embedding problems that aim to minimize various cost functions including the size of the multicast tree. Of related interest is the structure of multicast trees when the 2-level hierarchy of Internet routing—inter-domain at the AS level and intra-domain at the router level—is incorporated. Assuming members of a multicast group are chosen randomly from the nodes in an Internet AS topology, the scale-free nature of power-law topology—random subgraphs remain power-law with the same exponent—allows a scale-free characterization of the structure of AS level multicast trees once the multicast algorithm (e.g., minimum spanning tree) is specified. Considering that transit domains are made up of router level backbone networks, if an AS level multicast tree is further expanded to include details at the router granularity, we arrive at a fine granular picture

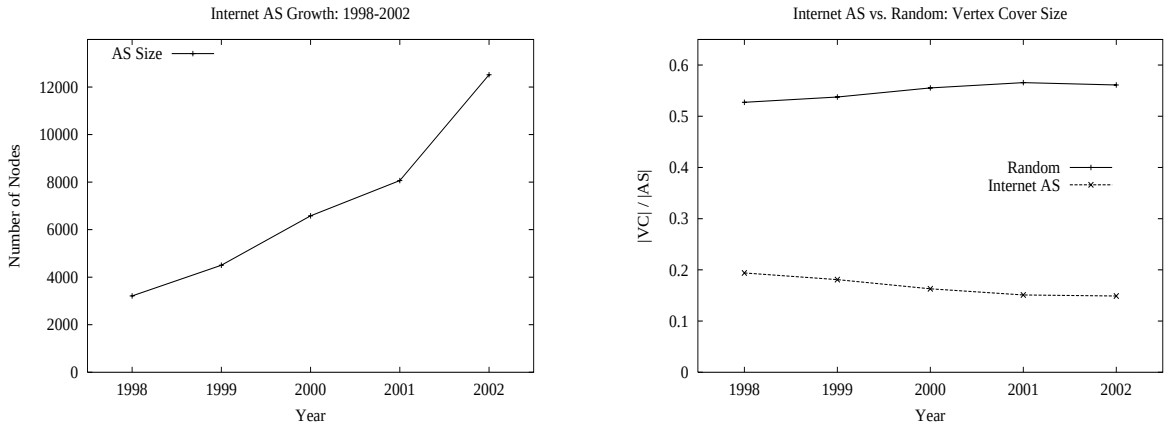
of multicast trees that incorporates the connectivity structure of router networks. Similar to path length inflation when routes are computed in a hierarchical fashion [130]—shortest-AS-path at the inter-domain level followed by shortest-router-path within each domain—multicast trees computed in a 2-level fashion are expected to be less efficient than those constructed from the global router level topology. Perhaps most intriguing is a succinct characterization of multicast trees obtained from 2-level hierachical routing, which may exhibit scaling properties such as those observed in [30]. Szpankowski et al. [128] have proposed a somewhat curious-looking tree structure, called self-similar tree, that possess distinctive unary segments in an attempt to explicate the power-law scaling behavior of multicast trees studied in [30, 111]. We conjecture that power-law connectivity, under 2-level hierarchical routing and existence of strong correlation between AS size and AS degree [129], induces multicast trees that resemble self-similar trees. Lastly, we note that for performance considerations in AS level graphs, "capacity," "congestion," "failure," and other performance measures must be carefully applied to yield meaningful interpretations. For example, a tier-1 transit AS is geographically dispersed with POPs across the continental United States and select international sites. A node failure in an Internet AS topology corresponding to such a transit AS would not be meaningful, even under massive power outages and other large-scale disturbances. This also holds for edge failures since a single AS level link between two transit domains may correspond to two or more physical peering points located in different parts of the country, say, one in New York, another in LA, and a third in Indianapolis. Unless all three physical connections go down, the two transit domains remain connected and their AS level link up.

4.3.2 Vertex cover: Distributed control and optimization

A vertex cover of a graph $G = (V, E)$ is a subset of vertices $S \subseteq V$ such that every edge $e = (u, v) \in E$ is incident on S, i.e., either u or v (or both) belong to S. A vertex cover (VC) achieves a covering of edges in a flow network which allows distributed detection and control of communication events involving the generation and forwarding of packets. In the next section we will discuss its application—in the form of distributed packet filtering—to denial of service and worm attack prevention. In this section, we are interested in finding small VCs in Internet AS graphs. Since nodes in a VC are engaged in detection and control, having a small node set that covers all edges, i.e., is a VC, is important for economy and ease-of-deployment. For example, in the global Internet, getting one domain to adopt a new technology is a complicated matter. To get multiple domains to agree on a common technology base, such as the deployment of distributed filters, is an even tougher challenge due to administrative autonomy, policy barriers and conflicts of interest. Thus our primary cost measure with respect to implementing distributed detection and control is the size of a node set—VC or otherwise—where actions are undertaken.

Finding a minimal VC in an arbitrary graph is an NP-complete optimization problem [58]. A problem is in NP if checking whether a candidate solution is indeed a solution is easy, i.e., can be done in polynomially many steps in the size of the input. Finding a solution may be easy—then the problem belongs to the class P $\subseteq$ NP—or it may be not. Intuitively, we would think that appreciating good music is easier than composing it. Artists and scientists would not be able to make a living without this maxim. Critics are professional solution checkers. The "P vs. NP" problem in computer science captures this popular and, on the surface, obvious truth but, to date, no one has been able to prove that P $\subsetneq$ NP. That is, existence of a problem whose solutions are easy to check but difficult to find. A problem is NP-complete if it is in NP and dominant: the ability to solve it would make every other problem in NP solvable. Finding a minimal VC is such a problem. Although a fast algorithm for minimal VC is not known—likely because none

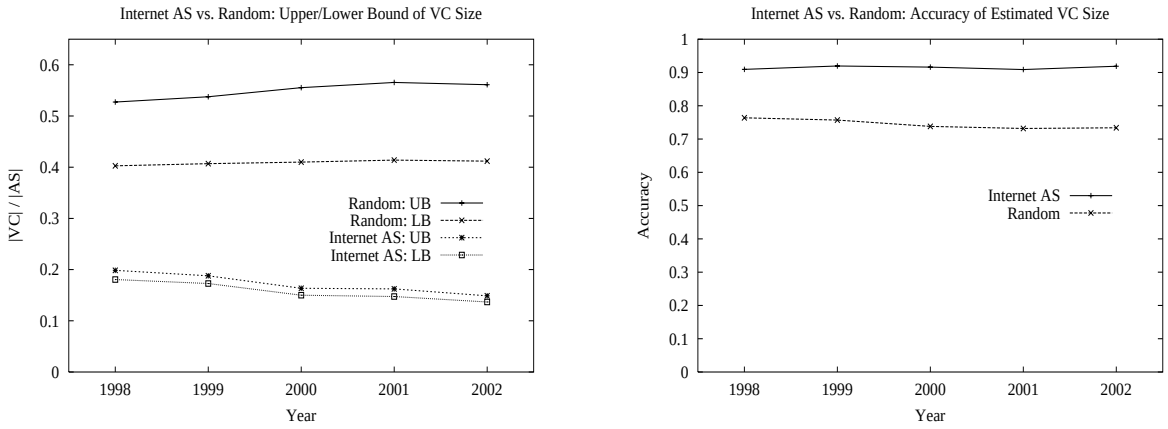
exists—there is a fast algorithm that always gives a VC that is at most twice as large as a minimal one. The algorithm starts from an empty cover, iteratively picks an edge, inserts the two incident nodes into the cover, removes the two nodes and any edges incident on them from the graph, and repeats the process on the smaller graph until no more edges remain. It is easily checked that the algorithm—more accurately procedure since we haven't specified how to pick the edges—achieves a factor-2 performance guarantee. A drawback of this algorithm under random edge selection is that it does not do well in practice: it is outperformed by a greedy algorithm that grows a VC by first inserting a node with the highest degree, removes the node and its incident edges from the graph, and repeats the process until no more nodes remain. An issue with the greedy algorithm is that no rigorous performance traits are known except that there are graphs on which the greedy heuristic fares poorly: the size of the VC found is at least a factor of $\log n$ bigger than a minimal VC. In the following, we show the size of small vertex covers found by running both the greedy heuristic and factor-2 approximation algorithm. The latter chooses edges randomly in the iterative growth process, and for each graph instance is run 100 times with different random seeds. For the graphs benchmarked below, the greedy algorithm always found smaller VCs compared to the factor-2 approximation algorithm. The factor-2 approximation algorithm, however, is useful for lower-bounding the minimum VC size. From 100 runs, pick the worst VC size found, divide it by 2, and it gives a lower bound on the minimal VC size.



(a) (b)

Figure 19: Internet AS topology. (a) Growth during 1998–2002. (b) Vertex cover size.

Figure 19(a) shows the size of Internet AS topologies inferred from NLANR/Oregon Route-Views measurement data for the period 1998–2002. We observe a slight super-linear trend. Figure 19(b) shows the smallest VC size found for the Internet AS topologies and corresponding random topologies of the same size and edge density. The VC size for Internet AS graphs falls below 15% in 2002, whereas for random graphs it increases above 55%. A practical implication of the VC gap is that if Internet AS connectivity were random, significantly more nodes would be needed to cover all edges in the network. Power-law connectivity affords economy of deployment through strategic placement. To ascertain the accuracy of the estimated minimal VC sizes, we plot both the upper bound (obtained by the greedy algorithm) and lower bound (obtained with the help of the factor-2 approximation algorithm) in Figure 20(a). We observe that for Internet AS



(a) (b)

Figure 20: Accuracy of estimated minimal VC size. (a) Upper and lower bound of minimal VC size for Internet AS and random graphs. (b) Accuracy of estimated minimal VC size.

topologies the two bounds are very close, achieving an accuracy level greater than 90% as shown in Figure 20(b). In the case of random topologies, the upper and lower bounds are significantly further apart leading to an accuracy level below 75% in 2002. Asymptotic estimates for the size of a minimal VC are known for $G_{n,p}$ random graphs under the condition that the average degree be large (i.e., $np \to \infty$) [56]. Since power-law graphs possess constant average degree, the corresponding random graphs must have constant average degree to achieve the same edge density. One of the technical challenges in power-law graphs is to understand what implications power-law connectivity has on combinatorial optimization. Presence of large degree nodes that induce locally star-like subgraphs makes optimization, to a first approximation, easier. However, additional increase in accuracy requires an improved understanding of the "backbone" structure of power-law topologies where intermediate-sized nodes play an important role. VC is a good starting point to study these questions as finding a minimum VC is intimately related to finding a maximum independent set and clique. An independent set of a graph $G = (V, E)$ is a subset of vertices where the nodes are mutually disconnected. A subset of nodes is a clique if everyone is connected to everyone else. It can be easily checked that if $S \subseteq V$ is a VC, then $V \setminus S$ is an independent set of G and a clique of $G^c = (V, E^c)$ where E^c is the complement set of E. Another interesting avenue for optimization in power-law graphs is a deterministic definition of power-law graphs on which optimization questions are studied. For example, given a (graphical) degree sequence $\mathbf{w} = (w_1, w_2, \ldots, w_n)$, the set of all graphs that satisfy the degree sequence is well-defined. One would like to define a sequence of graph families $\mathcal{G}(n, \beta, r)$ where β is the power-law exponent and r a fudge factor—G is allowed to be a member of $\mathcal{G}(n, \beta, r)$ if its degree sequence is r-close to the power-law degree sequence—where optimization can be addressed in a deterministic setting. Does minimum vertex cover remain NP-complete for $\mathcal{G}(n, \beta, r)$? One suspects that this is still the case due to the combinatorial messiness that the "backbone" of power-law topologies seem to harbor wherein smaller—but polynomially sized—hard problems may be embedded. Finding a robust formulation of deterministic power-law graphs that can be related to the average degree sequence framework and is amenable to analysis is the first challenge.

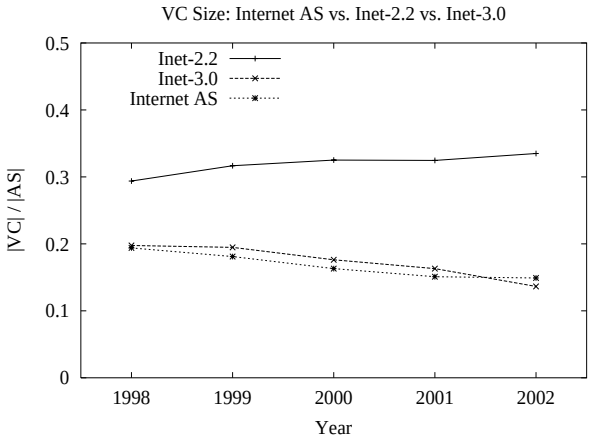


Figure 21: VC size comparison: Internet AS vs. Inet-2.2 vs. Inet-3.0.

We conclude this section with a note on artificial power-law topology generation that emulate real-world graphs, in particular, Internet AS topologies. Figure 21 shows the VC sizes for artificial topologies produced by Inet [72, 143], a topology generator that aims to mimick empirical Internet AS connectivity. There is a significant discrepancy in VC sizes between Internet AS topologies and corresponding artificial topologies generated by Inet-2.2 [72], which was first pointed out in [103]. This has, in part, prompted modifications to Inet-2.2 leading to Inet-3.0 [143], which yields topologies with VC sizes close to that of Internet AS topologies. One of the key changes in Inet-3.0 is the incorporation of "higher order" connectivity structure, meaning that in addition to how many neighbors a node possesses, its neighborhood composition with respect to degree distribution is injected into the graph construction process. The performance of Inet-3.0 has significantly improved over that of Inet-2.2, however, there remains room for improvement especially with respect to clustering [143]. A topology generation approach that captures higher order connectivity structure—$A, A^2, A^3, \ldots$ where $A = (a_{ij})$ is the adjacency matrix of a graph, i.e., $a_{ij} = 1$ if i and j are connected, 0, otherwise—may harness additional structure that is presently missed. As with second-order stationary processes for modeling self-similar traffic, higher order structure beyond a certain point may not matter.

4.4 Application: Denial of service and worm attack prevention

4.4.1 What are denial of service and worm attacks?

A denial of service (DoS) attack aims to disrupt services by depleting network resources such as bandwidth and CPU required to deliver the services. This is done by sending bogus work in the form of junk traffic and service requests that tie up resources preventing a network system from operating in its normal mode. Attacks may be targeted at servers, hosts, and increasingly the network infrastructure itself [93]—such as routers and name servers—with far-reaching impact. Distributed denial of service (DDoS) attacks that forge their source IP address, called spoofing, are especially severe, due to their concentrated force and difficulty in affecting timely recovery. Locating an attack source can take on the order of hours, sometimes days, by which time the damage has already been done. Vulnerability to denial of service is not unique to the Internet. An old form of DoS attack is "ring the door bell and run" practiced by select kids in days gone

past. Although innocuous, the occupant's time and energy are wasted, drawing attention away from other matters. On the Internet, repercussions from DoS attack are amplified due to the high attack volume enabled by broadband networks, ease of anonymity afforded by spoofing, and finger tip attack coordination from a computer anywhere in the world. Classical DoS attacks are many-to-one in the sense that an attacker recruits multiple attack hosts to overwhelm a target. Denial of service from spam mail, on the other hand, is one-to-many which is made possible by the slow processing speed of the target—human leafing through e-mail—and the financial incentives underlying the denial of service activity. One of the few effective defenses against DDoS attacks is assigning a cost to the attacker, either in the form of economic cost (e.g., usage pricing) or attacker identification followed by attribution with legal ramifications. Both achieve a deterrent effect, a principal reason why a DoS attacker wouldn't buy up all the baguette at the local bakery, or run off without paying.

A worm is a computer virus that travels from system to system on the Internet through network protocols invoked by distributed applications. A computer virus is parasite code—a block of instructions and data embedded in application or system code, its host environment—that upon execution can impart harm on a computing system involving information erasure, system hijacking (e.g., for the purpose of DoS attack), infection and self-replication, among an array of possibilities. In the 1980s a preferred mode for viruses to infect other systems was through floppy disks that contained infected applications or had infected boot sectors. Today's mobile viruses—i.e., worms—exist as parasitic attachments to messages that are transmitted, interpreted, and processed by network protocols as part of networked client/server applications. Worms face a tougher challenge than their brethren when aiming to take over a system. They need to remain clandestine—network protocols, which are structured programs expecting well-formed messages, must not see through the disguise—until they reach a specific point in the protocol code that contains a fatal vulnerability. The message containing the worm is structured so that it triggers the vulnerability leading to a coup d'état. The most prevalent vulnerability targeted by past worms, including the Morris worm (also called the Internet worm [127]) and Code Red [26], is the buffer overflow vulnerability. When a message is parsed, arguments are copied into the program's memory space holding data. If an argument is overly long, it may spill over into the program's instruction space, overwriting the program's instructions with parasite code hidden in the argument. To pull this off requires expertise on the part of an attacker, but the vulnerability would have been prevented had the code been more carefully written to check the length of arguments.

4.4.2 Fixing a hole in the security roof

Before we discuss the role played by power-law connectivity in "fixing a hole where the rain gets in" (to paraphrase Lennon & McCartney), let us examine some pertinent features of DDoS and worm attacks that constrain the type of defenses that may be mounted against them. An unspoofed DDoS attack is inherently difficult to shield against. Stemming an unusually large traffic flow—if deemed a DDoS attack—is an application of flow control and not difficult to do. The crux of the problem, however, is: how does one distinguish friend from foe? There are no known methods for keeping false positives in check ensuring that "the baby is not thrown out with the bath water." The Achilles' heel of intrusion detection systems is that anomalies, statistical or otherwise, are not sufficiently reliable to warrant automated responses that may, in the end, do more harm than good. They are useful to detect increased "chatter," but determining what the chatter means is, at the present, more art than science. If DDoS attacks are unspoofed, then their IP source addresses reveal the identity of the traffic source—attacker and legitimate—which then becomes the purview

of policy actions. Economic factors such as usage pricing—in the U.S. most Internet access is based on flat pricing—have only a small bearing on the problem as attackers, who compromise others' machines, do not bear the economic burden. A meticulous attacker who recruits hundreds of attack hosts by planting viruses well in advance of a timed DDoS attack is difficult to catch. A less meticulous attacker who has not sufficiently hidden his/her tracks—for example, by supervising an unspoofed DDoS attack in real-time—is much easier to apprehend. An effective solution, and perhaps the only long-term solution, for protecting against DDoS attacks is deterrence through attribution whose first step involves timely source identification.

Worm propagation shares many similarities with virus propagation in biological systems, the subject matter of epidemiology [11, 64]. Both are contact processes that spread through specific interaction, and in the absence of counter-acting forces rapidly infect the bulk of a susceptible population. The mathematical foundations of epidemiology are two-fold: a macroscopic theory of population dynamics based on dynamical systems theory and a microscopic theory of disease propagation based on percolation theory. The classic SIR epidemic model [64], which has its origins in the seminal work of Kermack and McKendrick [75], relates susceptibles (S), infectives (I), and removed (R)—i.e., the dead or otherwise inert—in a population N ($= S+I+R$) through a system of nonlinear differential equations. It is assumed that dI/dt increases proportional to SI but decreases proportional to I, $dS/dt \propto -SI$, and $dR/dt \propto I$. The product form SI captures the uniformity assumption that each susceptible is potentially exposed to a fixed fraction of infectives (and vice versa). Ignoring the dead—worms utilize infected systems for further propagation—the transient behavior of the population dynamics exhibits a characteristic "S-curve" where all susceptibles eventually get infected. With R in the picture, the steady state population may contain survivors, i.e., non-zero susceptibles, when the death rate exceeds the infection rate which drives I to zero. In an endemic model, death is accompanied by birth which in the worm propagation context may be interpreted as infectives being healed by reinstalling or cleaning up of infected systems. An S-curve characterizes the spread of worms on the Internet, examplified by Code Red whose spread was monitored by CAIDA [23] and shown at a DARPA meeting in July 2001 amidst a sequence of attacks.

4.4.3 Distributed packet filtering

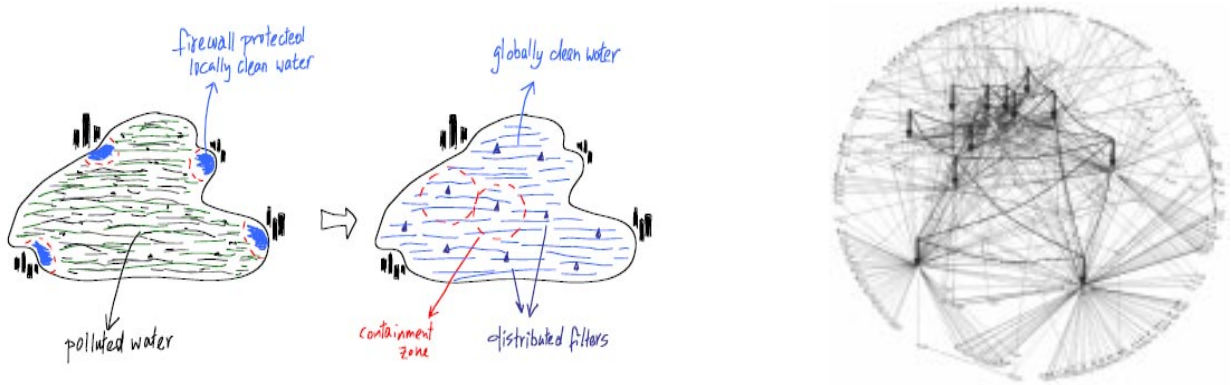
Cooperative protection and partial deployment Distributed packet filtering (DPF) is a new approach to protecting the Internet against DDoS and worm attacks. DPF embodies a "cooperative protection under partial deployment" paradigm that breaks with tradition in two important respects:

- *Local vs. global protection.* Firewalls epitomize local protection whose aim is to shield an entity from adverse outside effects. Amazon.com may succeed in shielding itself against a worm attack, however, if a significant fraction of its customer base is disabled by the same attack, the impact is ultimately shared. Selfish protection—manifested as local protection—only goes so far.

- *Partial vs. full deployment.* A network security solution, to be effective, must consider partial deployment a fundamental maxim of Internet vulnerability. Epidemiology teaches us why software patches, when partially deployed, are unable to contain the spread of worms. Insisting on increased user diligence assumes the problem away, is unrealistic, and not the domain of science and engineering.

The limitation of local protection under partial deployment exposes a locally protected service provider to DDoS attack: when others are unprotected, this allows them to be recruited for DDoS attack which turns others' vulnerability into one's own. A firewall suffers under lack of efficient friend from foe discrimination capability which impedes selective admittance of non-attack traffic. For DDoS and worm attacks, the fates of the protected and unprotected are intertwined. To overcome the limitation of local protection under partial deployment, a new dictum is required: (i) Protective action must be centered at transit points, not end systems, to exploit checkpoint screening and containment afforded by transportation networks. (ii) Transit points must cooperate to affect global protection; a locally protected network system is inherently vulnerable to DDoS and worm attack. (iii) The collective action of a few, under partial deployment, must yield an overwhelming synergistic effect that protects the whole. In general, realizing a solution that satisfies all three conditions is a tall order. The first two properties are architectural features and, as such, part of the "cooperative protection under partial deployment" design space. Their viability critically depends on (iii)—a performance feature—and the most difficult part of the new approach. It is here where power-law connectivity plays a crucial role.

Casting a net over Lake Wobegon To illustrate the notion of distributed packet filtering, suppose Lake Wobegon is being polluted by hostile elements carrying out water contamination attacks. Contaminated water affects fish and wildlife, and eventually threatens water supplies. Local filtering can cordon off a shore segment and purify the water therein for human consumption. However, Inter-city commerce—mediated by ships and boats on water routes—dwindles for fear of admitting contaminated water particles. The fishing and sea food industry is shot. Cities that did not heed precaution turn into ghost towns. Under distributed filtering, a number of cities band together and install water filters across the whole lake. Individual filters are not aimed at protecting a specific town or city but the system as a whole. Any one community has little incentive to install a filter outside its immediate living space: a filter or two in the middle of the lake would not do much good anyway. It is only when a sufficient number is deployed that contamination introduced anywhere on the lake gets trapped by the filter net and further spread is contained. The state-of-affairs in Lake Wobegon before and after distributed filtering is illustrated in Figure 22(a).



(a) (b)

Figure 22: (a) Pollution in Lake Wobegon before, and after, distributed filtering. (b) Filter net for DPF in power-law network.

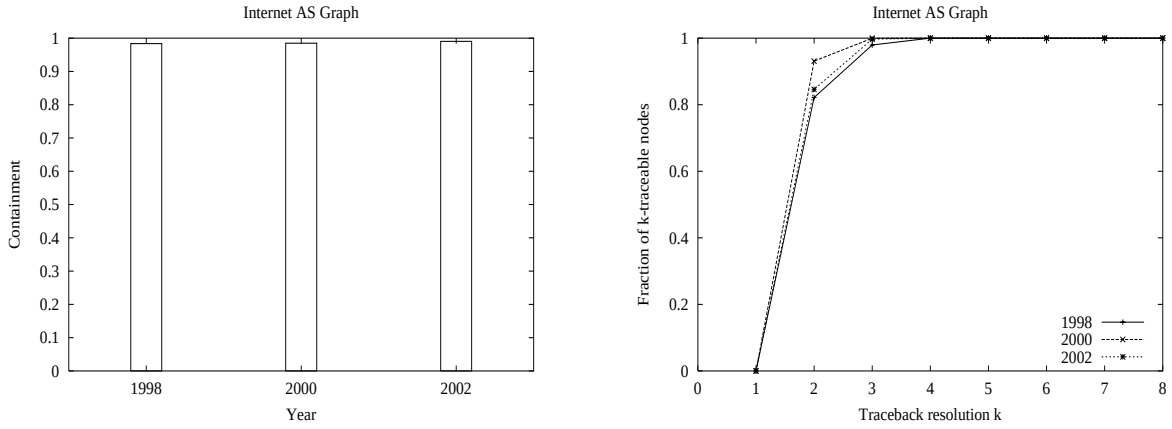
Distributed packet filtering only satisfies properties (i) and (ii): cooperative protective action carried out at transit points for the good of the whole. Supposing Lake Wobegon is "super-sized" to Internet scale, without property (iii) that mandates a small filter deployment with a big bite, distributed filtering would not be feasible. Only when an economic filter placement achieves decisive protection can relevant transit parties be brought together—some screaming and hollering—and induced to form a coalition for the greater good. The larger the required coalition, the less chance this has of succeeding. Thus minimizing deployment becomes a key objective. In a transportation network, the paths that an entity can take are constrained by the underlying connectivity structure. In communication networks, routing further restricts the route that a packet can take. In road systems, checkpoints at border crossings and intersections have been used to shield countries and cities from unauthorized access. These gateways, when engaged as distributed filters, have been less effective at apprehending fugitives of justice: under partial deployment, there are simply too many ways for a person to travel from one point to another undetected. In power-law networks, the tables are turned. Even though checkpoints are few, it is next to impossible to go anywhere without encountering a sentinel. Figure 22(b) shows a power-law network with strategically placed distributed filters. In the next two sections, we will outline two forms of distributed packet filtering—route-based and content-based filtering—and summarize how global protection is effected by power-law connectivity under partial deployment.

4.4.4 DDoS attack: Route-based packet filtering

What is route-based filtering? The primary task of a router, upon seeing a newly arriving packet, is to ask where it is headed ("quo vadis") and send the packet on its way based on the answer provided in the IP destination address. We advance that a router ask a second question—"where do you hail from?"—and discard the packet if it can be unequivocally determined that it is lying. That is, the IP source address is spoofed. For example, if a packet arrives at an interface claiming to originate from A destined for B when, based on routing, it is impossible for such a packet to enter through the said interface, the packet is spoofing and dropped. When this verification is done without utilizing the destination address—in the above scenario there may not exist any destination address for which a packet from A would enter through the given interface—we call it semi-maximal route-based filtering, in contrast to maximal filtering where both addresses are used. If, by exploiting route constraints, we discard spoofed packets before they can reach their target and impart harm—e.g., as part of a spoofed DDoS attack—we endow the system with authentication capability without engaging cryptography. There are legitimate instances where spoofing is useful, such as in Mobile IP where an intermediary forwards packets on behalf of a mobile user. These protocols are not widely used, however, and can be modified to work without resorting to source address spoofing.

Proactive and reactive protection Route-based distributed packet filtering was introduced in [103]. Its goal is to achieve proactive protection by discarding spoofed packets before they can reach their target, which stops spoofed DDoS attacks in their tracks. We quantify proactive protection by defining a performance measure Φ, called containment, where $0 \leq \Phi \leq 1$ denotes the fraction of innocuous nodes: a node is innocuous if any spoofed packet emanating from it, destined to anywhere, is discarded by a filter in the filter net. For example, if $\Phi = 0.95$, this means that 95% of all nodes are not suitable as staging grounds for a spoofed DDoS attack. An optimization version of the problem is: given a desired containment Φ, find a minimum filter net such that proactive protection with Φ is achieved. It is not difficult to prove that optimal spoofed DDoS containment

is NP-complete. The need to minimize filter deployment may necessitate imperfect containment which, in turn, brings out the problem of identifying the source of an attack. In January 2002, the Internet had more than 12,000 domains, which leaves open the possibility that with $\Phi = 0.95$, hosts from 600 domains could collude in a spoofed DDoS attack. Locating the physical source of a spoofed IP packet, also called traceback, becomes an issue. We quantify traceback by defining k-traceability where a node is k-traceable if, upon receiving an IP packet, the physical source of the packet can be localized to within k sites. The performance measure $0 \leq \Psi(k) \leq 1$ defines the fraction of k-traceable nodes. For example, if $\Psi(6) = 1$, this would mean that all nodes are able to localize the origin of received IP packets—spoofed or otherwise—to within 6 sites. Compared to the uncertainty of 600 potential sites under $\Phi = 0.95$, this represents a factor-100 improvement.



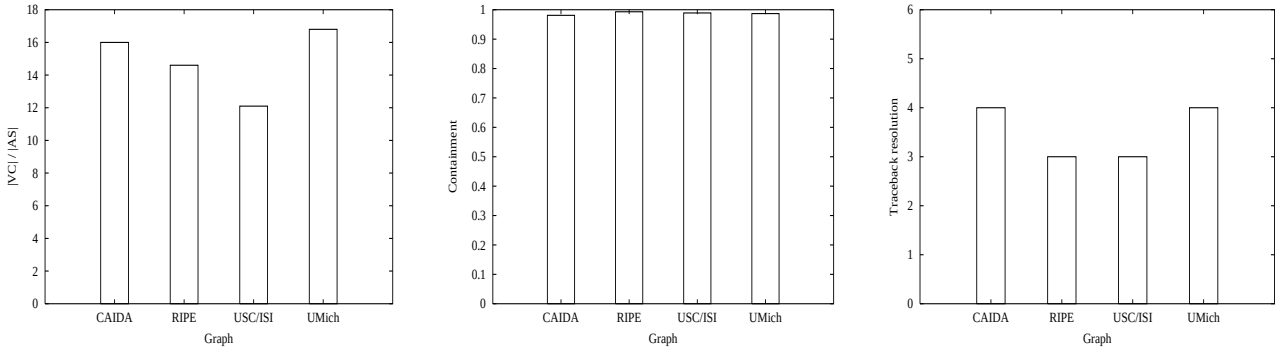
(a) (b)

Figure 23: Spoofed DDoS attack protection. (a) Containment. (b) Traceback.

Figure 23 summarizes proactive and reactive protective performance for NLANR/Oregon Route-Views Internet AS topologies under VC based filter placement: 19%, 16%, and 15% for 1998, 2000, and 2002, respectively. The graph size during the same period increased from 3,015 to 12,517. Figure 23(a) shows containment Φ for the three years which increased from 98% in 1998 to 99% in 2002. Thus with 15% filter deployment, we are able to achieve 99% proactive protection restricting the attacker to just 1% of the domains for staging spoofed DDoS attacks. Figure 23(b) shows the fraction of k-traceable nodes, $\Psi(k)$, as a function of k for the same benchmark topologies. We observe that for all three years $\Psi(4) = 1$. That is, all nodes, upon receiving an IP packet, can locate the origin of the packet to within 4 sites. In the 2002 topology, 85% of the nodes can narrow the range down to 2 sites. Constant containment and traceback performance in the presence of a 4-fold increase in the size of the topology makes route-based DPF a scalable DDoS solution.

Robustness As noted earlier, Internet AS topology data represent an approximate picture of AS level connectivity, which are imbued with inaccuracies introduced by the measurement data as well as inferences drawn from them. To ascertain whether route-based DPF performance is sensitive to the details of the underlying measurement data, we evaluate VC size, proactive and reactive protection for a range of measurement topologies that draw on different measurement methods, data sources, and inference procedures. Figure 24 shows proactive and reactive performance of

route-based DPF on Internet AS topologies from CAIDA [24], RIPE [120], USC/ISI [89], and the University of Michigan [134]. Figure 24(a) depicts VC sizes found by the greedy algorithm which fall in the 12–17% range for the four data sets. The VC sizes are, overall, consistent with those found for NLANR/Route-Views topologies. Figure 24(b) shows that containment lies in the 98–99% range, and Figure 24(c) shows the minimum value of k for which $\Psi(k) = 1$, which yields $k = 3$ or 4. That is, all nodes are able to localize the physical source of received packets to within 3 or 4 sites.



(a) (b) (c)

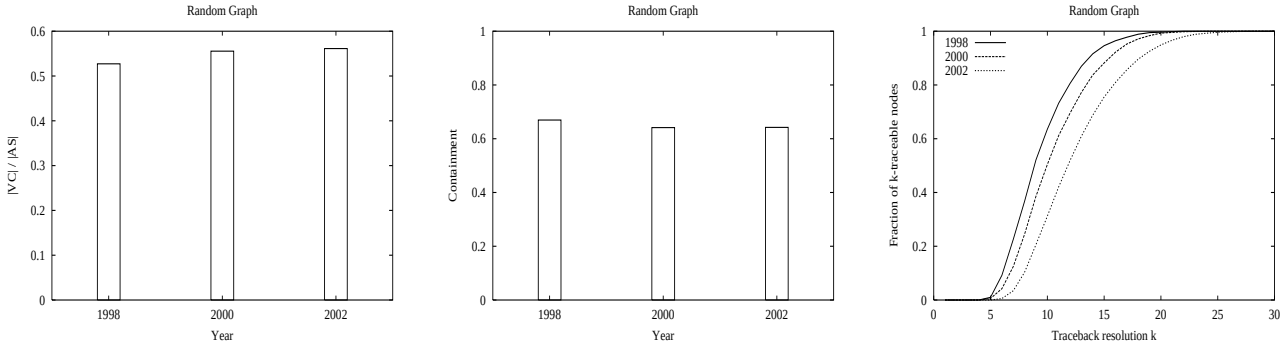
Figure 24: Protective performance on CAIDA, RIPE, USC/ISI, and UMich topologies. (a) VC size. (b) Containment. (c) Traceback.

Random vs. power-law connectivity Figure 25 shows the performance of route-based DPF on random graphs of the same size and edge density as the NLANR/Route-Views AS topologies. Figure 25(a) shows the much larger VC sizes found in random graphs (cf. also Figure 20(a)). For example, in 2002 the size of the VC filter net exceeds 55%. Figure 25(b) shows that despite the nearly 4-times larger filter net, proactive protection in random graphs is significantly reduced from 99% to 65%. Figure 25(c) shows that traceback performance has degraded as well, with the smallest k such that $\Psi(k) = 1$ reaching 27 in the 2002 graph. The results indicate that route-based DPF in randomly connected networks would not be a viable solution. Power-law connectivity is essential to scalable protective performance effected by strategic, economic filter placement.

VC filter placement and power-law connectivity The vertex cover heuristic for selecting filter sites induces two properties relevant for facilitating proactive and reactive protection in power-law networks:

- *Preference of high degree nodes.* The greedy VC heuristic assigns preference to high degree nodes which leads to economy in filter deployment due to the prevalence of stub ASes that are connected to large transit ASes. For example, in the 2002 Internet AS topology, more than 1/3 of the nodes are stub domains of degree 1.

- *Uniform filter density along routes.* A more subtle property—critical for reactive protection—is uniform filter density along any end-to-end path. This is a consequence of the VC property: to cover all edges along a route, at least every other node must belong to the filter net. Uniform filter density severely limits "holes" through which spoofed packets can sneak in.



(a) (b) (c)

Figure 25: Protective performance of route-based DPF on random graphs. (a) VC size. (b) Containment. (c) Traceback.

Containment is primarily effected by the first property. Traceback is effected by the second property. A filter placement method that chooses the top 15% of high-degree nodes—a purified form that implements the first property—achieves traceback resolution 263, i.e., $\Psi(263) = 1$, which is significantly worse than $\Psi(4) = 1$ achieved by 15% VC based placement. Moreover, $\Psi(261) = 0$. That is, no node achieves 261-traceability and there is a sharp transition at $k = 263$ ($\Psi(k) = 0.02$ at $k = 262$). The influence of the second property is illustrated in Figure 26 which depicts a routing tree rooted at node J, the target of a spoofed DDoS attack. Assuming H is a non-filter transit node, by the VC property, A, D, G, and I must be filter nodes. They form a perimeter around H isolating it from the rest of the network. Let ψ_X denote the size of the maximum spoofable address space that a node in the subtree rooted at X can engage to attack node J. That is, X can craft a spoofed packet that reaches J—i.e., not discarded by the filter net—and J cannot resolve the origin of the packet beyond ψ_X candidate sites. We assume filter nodes do not allow spoofed packets to emanate from within. We use ψ_J to denote the traceback resolution of J. If Y is the parent of X, then $\psi_X \leq \psi_Y$. Thus $\psi_J = \arg\min_k[\Psi(k) = 1] = \max_X \psi_X$ where $\Psi(k)$ is restricted to target J and the maximum ranges over all X in the routing tree of J.

In the example shown in Figure 26, B is a stub node, a degenerate singleton subtree. Its parent D is a filter node, hence $\psi_B = 1$. The same holds for C. For transit node D which is a filter node, $\psi_D = 1$. $\psi_E = \psi_F = 2$ and $\psi_G = 2$. The configuration in Figure 26 relaxes the VC property by allowing both E and F to be non-filter nodes, leaving edge (E, F) uncovered. E and F belong to the same pocket: two nodes are defined to belong to a pocket if one can reach the other without traversing a filter node. Nodes in the same pocket possess the same spoofable address space: F can claim to be E, and E can claim to be F, with impunity. In general, given a routing tree and filter net, the following recursion holds for ψ_X if X is a filter node:

$$\psi_X = \max\{\psi_{c(X)} : c(X) \text{ is a child node of } X\}.$$

The max operator assures that traceback uncertainty does not additively build up at filter node junctions in the routing tree. A similar relation holds for a non-filter node Y as long as its neighbors—children and parent in the routing tree—are filter nodes, which is assured by VC:

$$\psi_Y = \max\{\psi_{c(Y)} : c(Y) \text{ is a child node of } Y\} \ + \ 1.$$

In the example, $\psi_H = \max\{\psi_A, \psi_D, \psi_G\} + 1 = \max\{\psi_A, 2\} + 1$. Using the recursions, we can establish an upper bound on the traceback resolution when filter placement is VC: halve the diameter of the given network. The exact bound is given by the maximum number of non-filter nodes on any path from stub to destination in the routing tree. Since power-law graphs have small diameter—e.g., the 2002 Internet AS topology has diameter 11 (random power-law graphs have diameter $O(\log n)$)—this yields an analytical bound on traceback resolution in power-law graphs under VC filter placement that utilizes the small world property. The preceding development also shows how to relax the VC property to further reduce filter deployment: find a filter placement that yields small pockets so that the coarsified routing tree—nodes in the same pocket are grouped into a single (weighted) super-node—has accurate traceback resolution.

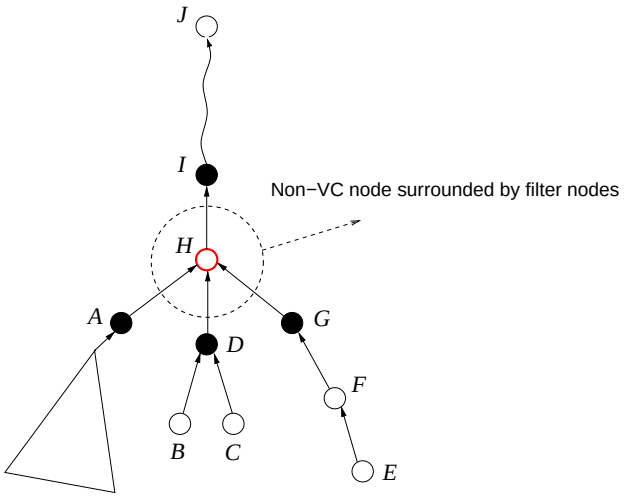


Figure 26: Routing tree rooted at node J. A, D, G, I are VC filter nodes, and H is a non-filter node.

4.4.5 Worm attack: Content-based packet filtering

What is content-based distributed packet filtering? In content-based distributed packet filtering, content-based filters are installed at strategic locations in a network such that packets carrying worm parasites are detected and discarded before they can reach their target. Content-based DPF is both a generalization and specialization of content-based filtering carried out in firewalls. It is a generalization in the sense that filter deployment at transit points is guided by the principle of protecting the system as a whole, in contrast to local protection affected by firewalls. It is a specialization in the sense that content-based filters are primed to detect packets carrying worms, an undertaking whose scope is narrower and more focused than devising content-based filters in general. For example, in the Code Red worm, a cleverly formatted GET request—a type of HTTP message—along with a disallowed body ends up triggering a buffer overflow that ultimately transfers control to the worm code [88]. One simple filter rule is to discard any IP packet that transports a TCP packet carrying a HTTP GET message with a body: the HTTP standard stipulates that a GET message cannot have a body. The length of the body—3,569 bytes—prompts the malformed HTTP GET message to be split across multiple IP packets which can complicate the worm detection process at routers. In other content-based filters, the GET request may be inspected to ascertain if it is a Code Red worm—the request part contains a characteristic signature—without having to

check for the presence of a body. Both filter rules are resistant to mutations where parts of the request or body are perturbed to defeat detection. This resilience is availed to worm filtering by the structured nature of network protocols: worms must abide by protocol rules until such time when a target vulnerability—e.g., buffer overflow—can be triggered. This makes parasitic variants stick out, rendering them efficiently detectable ("it's difficult to conceal a gun, samurai sword, or other weaponry when doing a triple somersault"). A caveat is that recognition of a specific vulnerability—e.g., buffer overflow in Windows indexing service IIS which Code Red exploits—must precede its prevention through content-based DPF. Anticipating and protecting against future instances of buffer overflow vulnerabilities is an impossible task: the vulnerabilities can be very subtle, as is the case in Code Red, instigated by an unbounded "steadfastness" exhibited by programmers at writing potentially buggy code. The championship belt in mediocre craftsmanship is held by a company in Redmond that produces equivalents of Trabants in the days of the iron curtain, but the responsibility is shared by many including computer science departments at universities where many of the engineers have been educated. The good news is that new worms such as Code Red were unleashed after public announcement of new vulnerabilities and patches. Deployment of updated content-based filters at distributed filter sites before public release can achieve preventative protection.

Worm propagation dynamics Epidemiology teaches us that patching, due to partial deployment and local protection, cannot protect the segment of the Internet where patch updates have not been installed. Persistence of partial deployment is also the root cause why repeat attacks—many worm attacks are reincarnations—continue to wreck havoc weeks and months after first impact. Figure 27 shows worm propagation dynamics under two extreme contact—i.e., victim selection—rules: local where nearest neighbors are selected, and random (or global) where targets are chosen randomly from a global address space. A lightweight preprocessing step, called scanning, precedes the actual

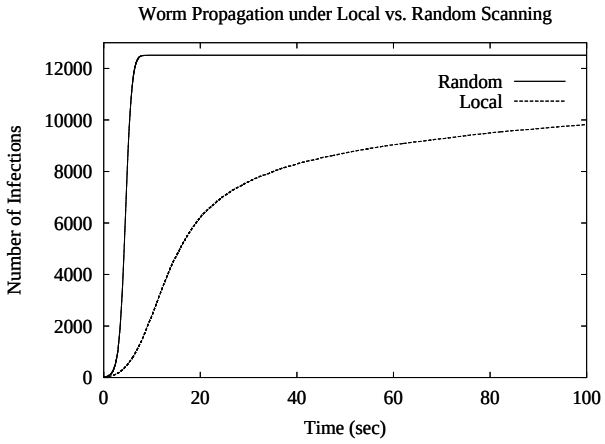


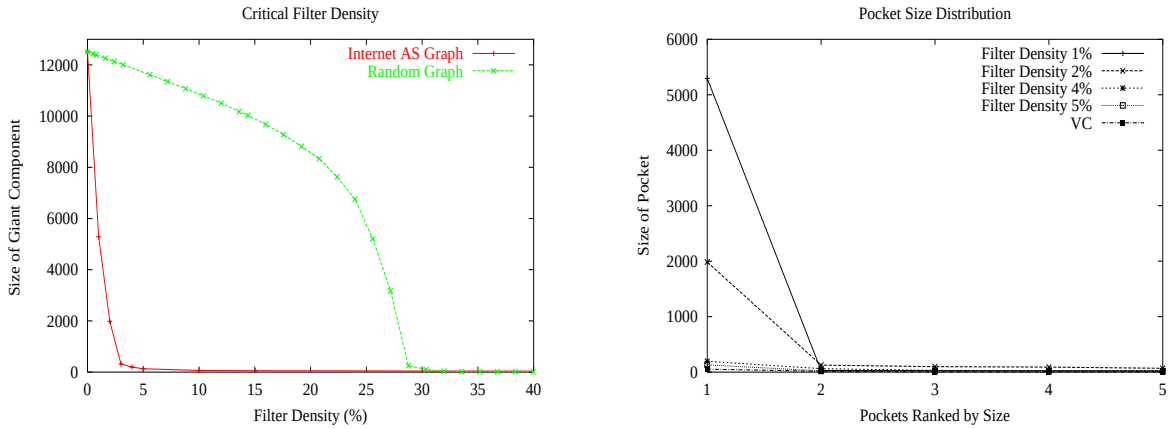
Figure 27: Worm propagation dynamics in 12,512-node Internet AS graph under local vs. random scanning.

worm transfer, aimed at discerning whether a target is potentially vulnerable. In actual attacks, both selection strategies, their mixture, and custom selection rules are used. At a scan rate of a few scans per second, it takes but a minute to infect a large chunk of the Internet. Containing worm attacks under partial deployment is only feasible by deploying filters at strategic transit nodes—part of a carefully managed infrastructure defense—through which IP traffic must pass through. When

the underlying network topology is power-law, the contact rule implemented by the application layer worm propagation has little effect. For example, in the Melissa worm [25]—called a macro virus because it is propagated as a Word attachment in Outlook e-mail that executes embedded code, i.e., macros, when opened—the contact rule consults the user's address book to determine subsequent targets. Whatever the connectivity structure induced by users' address books—perhaps a small world social network not unlike Milgram's [91]—power-law AS topology constrains the paths that a packet can take from source to destination, making it difficult for worms to travel undetected. A defense architecture that deploys filters in mail server networks would not be viable due to its mesh connectivity: a mail message sent from a host in domain A to a host in domain B is, at its core, a transaction between mail servers at A and B. Mail servers at other domains do not get involved. Partial filter deployment in mesh networks only yields local protection, with the same performance limitation as firewalls.

Worm containment: power-law vs. random connectivity Given a network $G = (V, E)$, routing, and filter net $S \subseteq V$, v is reachable from u if the path from u to v does not contain a filter node. Thus a packet from u to v can travel undetected and is infectable by u, denoted $u \twoheadrightarrow v$. The relation "$\twoheadrightarrow$" is transitive and partitions V into equivalence classes where u and v are in the same equivalence class if both $u \twoheadrightarrow v$ and $v \twoheadrightarrow u$. "$\twoheadrightarrow$" also induces a partial order—more precisely, a forest of disjoint trees—on the resulting equivalence classes. We are interested in finding a small filter net S such that all trees are small. Since a single infected node at the root of a tree can infect all other nodes in the tree—the most economic way for a worm attacker to wreck maximum havoc—the smaller the trees the more effort is required on the part of the attacker to contaminate a network system. If $\mathcal{T}$ is a largest tree, then at least $|V|/|\mathcal{T}|$ nodes must be separately compromised to infect the whole network G. Conversely, a local outbreak in $\mathcal{T}$ is contained within $\mathcal{T}$, the goal of isolation in disease control.

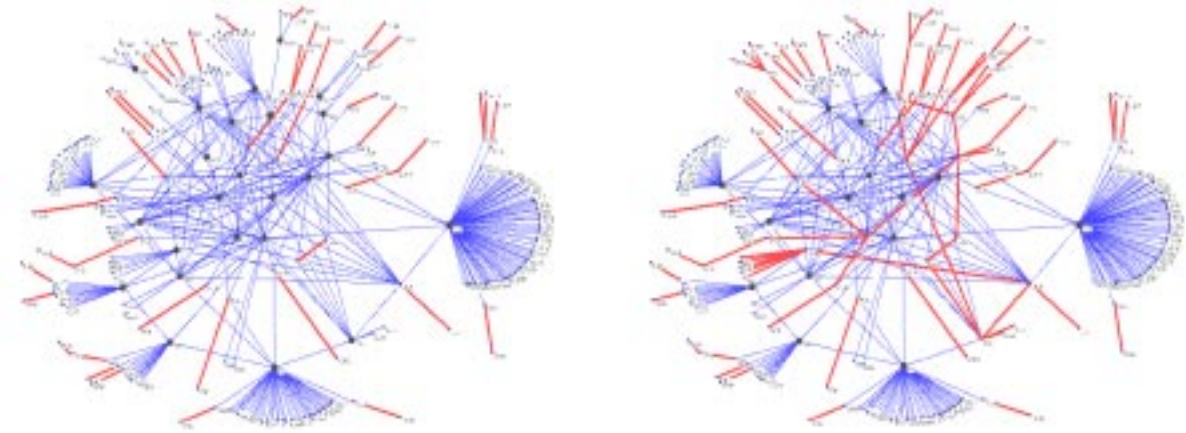
Figure 28(a) shows the size of a largest tree—we also call trees "pockets"—for the 2002 Internet AS topology under a pruned VC filter placement: x% filter density means that after computing a VC it is pruned to the smaller target size x by discarding low-degree nodes first. We observe that



(a) (b)

Figure 28: (a) Critical filter density for 12,514-node Internet AS graph vs. corresponding random graph. (b) Pocket size distribution under varying filter density ranked by size.

there is a critical filter density around 3–4% at which there is a sharp change in the size of a maximal pocket. The threshold phenomenon is also present in a corresponding random graph of the same size and edge density, albeit with the threshold located near 28%. The engineering implications are two-fold: one, worm containment under economic deployment is facilitated by power-law connectivity, and two, deploying filters below the critical threshold is of little use while deploying filters above the critical threshold is wasteful—knowledge of the critical filter density is crucial to achieving effective protection. Figure 28(b) shows the size of a second largest, third largest, fourth largest, and fifth largest pocket under different filter densities. We observe that containment is dominated by the largest pocket which is unique. Figure 29(a) depicts the distribution of pockets in a 300-node subgraph of the 2002 Internet AS topology where filter deployment is above the critical density. We observe isolated pockets of size 1–4. Figure 29(b) shows pocket size distribution under a filter deployment below the critical filter density. We observe long distance linkages producing a large pocket. The critical filter density holds for CAIDA, RIPE, USC/ISI, and UMich topologies, and under different routing algorithms including semi-random and random routing.

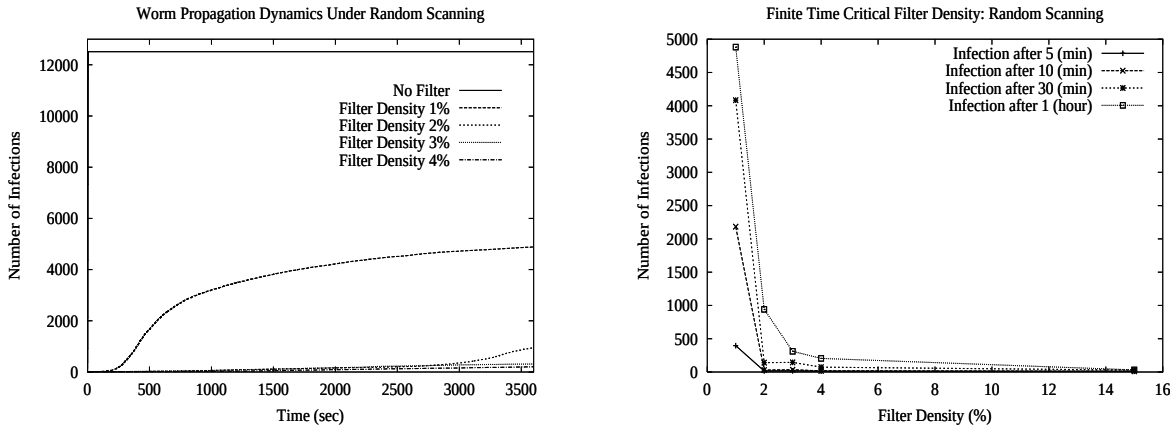


(a) (b)

Figure 29: Worm propagation in Internet AS topology. (a) Containment of infection. (b) Large-scale contamination.

Finite time dynamics Figure 30(a) shows worm propagation dynamics under random scanning for different filter densities. We observe that 2% filter deployment, even though ineffective asymptotically (cf. Figure 27), is able to slow the speed at which worms spread over finite time horizons. Figure 30(b) shows that the critical filter density is accordingly shifted to the left, whose magnitude depends on the finite time horizon. This may give a stripped down defense mechanism a little breathing room before it is fully activated, with the caveat that the deployed filters must already be able to detect the propagating worm. Thus the benefit of deploying fewer filters than mandated by the asymptotic critical filter density is largely restricted to repeat attacks of known worms where, for efficiency reasons, not all filter sites are primed against all known worms. Randomization over the set of known worm signatures coupled with on-demand filter loading may be needed depending on the overhead associated with mutation-resistant filtering and the size of the worm set.

Load based filter placement It turns out that a load-driven filter placement strategy is able to



(a) (b)

Figure 30: (a) Spread of infection as a function of time for different filter densities. (b) Critical filter density at finite time horizon.

achieve a 50% smaller asymptotic critical filter density than VC oriented filter placement. In one load based filter placement algorithm, vertices are ordered by node load—actually a normalized variant that discounts overlaps—and the highest ranked r nodes selected for inclusion in the filter net. The r nodes are then removed from the graph to yield one or more connected components. Node load is computed for the largest connected component, and the selection and partitioning procedure recursively repeated until all connected components are below a target size. Figure 31 shows the critical filter density when $r = 5$. For very high degree nodes, the order dictated by node load is nearly the same as that determined by degree, therefore their performance effect similar. It is for intermediate load rank where filter selection significantly deviates between the two strategies, with node load more robustly identifying key junctions whose removal splits a connected component into many smaller connected components.

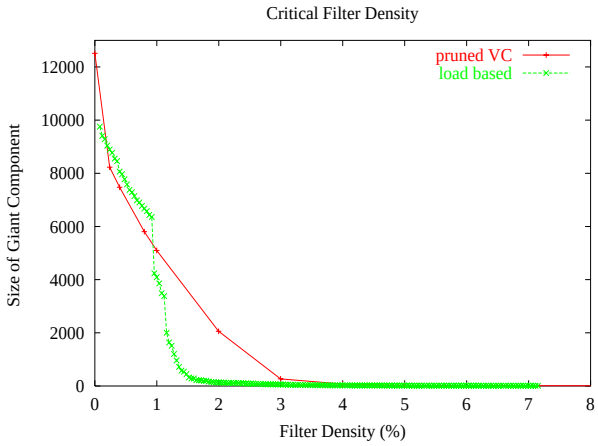


Figure 31: Critical filter density under load-based filter placement.

4.5 Discussion

The recently discovered power-law connectivity of various real-world networks, including Internet related graphs, helps address a long-standing question: how do real-world networks look like? Do they possess shared properties that may be used for performance evaluation? Although the area of power-law networks is still in its infancy, initial studies provide strong indication that power-law connectivity is a wide-spread phenomenon with repercussions to understanding, designing, and controlling networked systems. It is interesting that both network traffic and network topology owe much of their structure to heavy-tailed object sizes—files for self-similar traffic and neighborhood size in the case of power-law topology—that lead to self-similarity across multiple scales where a part resembles the whole. In the network connectivity context, this is also referred to as scale-freeness [12]: a random subgraph of a power-law graph inherits the parent's power-law degree distribution, a consequence of random sampling. For the same reason that self-similarity alone does not adequately characterize network traffic—e.g., Brownian motion, when interpreted as network traffic, is self-similar but not long-range dependent—so does scale-invariance only hit part of the nail for network topology. A more essential property is power-law degree distribution, a trait distinct from self-similarity, that is responsible for generating a wide range of variability.

A modeling difference between self-similar traffic and power-law network topology lies in the explication of their causality. In self-similar network traffic, we can empirically ascertain that file sizes tend to be heavy-tailed which provides the causal kernel upon which a reductionist explanation of the traffic phenomenon can rest. One may speculate why file sizes are heavy-tailed, but it is unclear that this leads to scientifically verifiable conclusions. For network topology, power-law connectivity appears to embody the dynamics principle "the rich get richer and the poor get poorer." Although appealing as a qualitative observation, as a quantitative explanation of why power-law connectivity arises in specific contexts it misses several important ingredients. They include static design—airline and telecommunication networks, when initially laid out, are organized around hubs and POPs that are linked through a backbone, a design approach with a heavy dose of bootstrapped centralized management—and economic, social, and technological factors that can affect the growth and demise of individual components. A scientifically rigorous description of the origin of power-law connectivity seems to necessitate a better understanding of human behavior—collective and singular—a challenge transcending the confines of power-law connectivity.

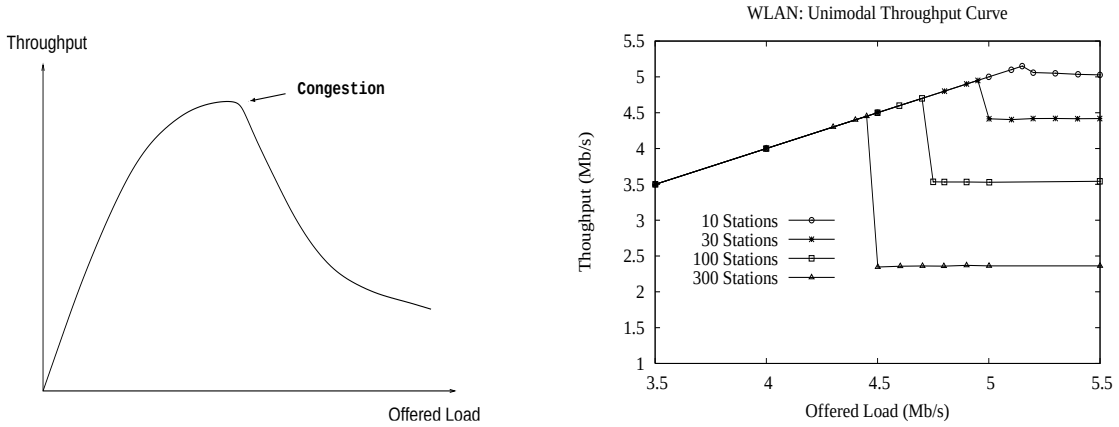
Power-law networks open the door to fresh questions in optimization: are NP-hard graph problems in combinatorial optimization easier with respect to approximability, or even poly-time solvability, when graphs are restricted to be power-law? For example, the upper and lower bounds on VC sizes for Internet AS graphs show that VC approximation using the greedy algorithm yields significantly improved results vis-à-vis random graphs. Random walk, percolation, imperfect information games, and a host of other dynamic and static problems on graphs present new twists to established areas. There is, as yet, no good definition of deterministic power-law graphs, which is one of the starting points in this endeavor. The application of optimization problems to network security and resource provisioning indicates that a better understanding of power-law networks has the potential of approaching practical problems in novel ways.

Lastly, we remark that families of random graphs obeying a power-law degree distribution may, or may not, be sufficiently adequate to capture pertinent properties observed in real-world networks. Based on vertex cover size observations in real and artificial Internet AS graphs [103, 143], it appears that accurate capturing of vertex cover properties requires second-order connectivity structure, in addition to first-order structure in the form of power-law degree sequence. Refinements and their justification remain tasks for the future.

5 Game Theory

5.1 What are noncooperative network games?

A canonical example of a noncooperative network resource game that arises in the Internet context is congestion control. In a 2-party congestion control game, Alice and Bob—two characters familiar in crytography—share a common network resource that may get congested if too much traffic is submitted to the network. Congestion, technically, means that the total, i.e., system-wide, throughput declines if offered traffic exceeds a certain level. This is represented by a unimodal, dome-shaped load-throughput function [59, 109] (cf. Figure 32(a)) whose exact shape is dictated by the specific network context. One example, both old and new—old because the protocol in question has its roots in Abramson's ALOHA packet radio network [2] used in the early 1970s to connect the University of Hawaii's island campuses—is the throughput of wireless hot spots. In a wireless local area network (WLAN), resource sharing is governed by a competition-oriented protocol called carrier sense multiple access with collision avoidance (CSMA/CA). A wireless station such as a laptop or handheld device listens to the radio channel ("carrier sense"), and if the channel is deemed idle data transmission is attempted. If more than one station sends packets at about the same time ("multiple access"), the physical signals "collide" leading to corruption. Collision is detected at the sender by the absence of an acknowledgment packet—in Ethernet, which uses a variant of the CSMA/CA protocol, collision can be detected at the physical layer without recourse to explicit acknowledgment (ACK) packets—at which time a form of exponential backoff in the retry time interval is instituted. That is, consecutive collisions lead to increasing pauses between retry attempts. All else being equal, the more stations and/or the higher the traffic demand



(a) (b)

Figure 32: (a) Unimodal load-throughput curve and onset of congestion. (b) IEEE 802.11b WLAN throughput as a function of offered load for varying number of wireless stations.

at stations, the higher the probability of collision, which eventually leads to a decline in system throughput due to wasted bandwidth. Figure 32(b) shows the simulated load-throughput curve of IEEE 802.11b's CSMA/CA for 10, 30, 100, and 300 stations as a function of offered load. The sharp drop at a critical offered load stems from a correspondingly sharp increase in the collision

rate. In real WLANs, throughput decline tends to be less pronounced due to biases resulting from uneven channel quality that skews the competition. In either case, there is a significant jump in the throughput variability within a session (over time) as well as across sessions at different wireless stations as the saturation point is crossed.

Returning to the congestion control game, it may be viewed as an instance of Prisoner's Dilemma mentioned in the Introduction, where Alice and Bob have recourse to two strategies upon encountering congestion: cooperation which would mean backing off to help reduce total offered load, or selfishness which may entail performing a congestion control action that does not reduce one's own traffic submitted to the network. Figure 33 shows a throughput matrix of Alice and Bob under all four combinations of congestion control strategies. When both parties cooperate (C), each achieves a throughput of 5 Mbps for a system throughput of 10 Mbps. When one is selfish but the other is

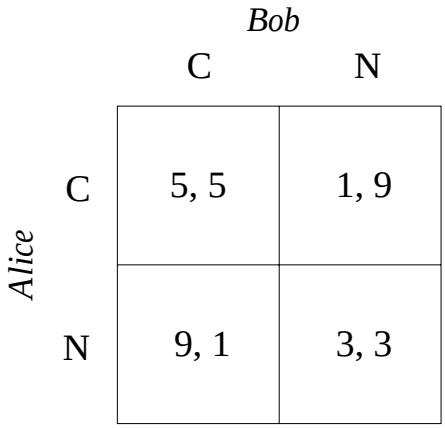
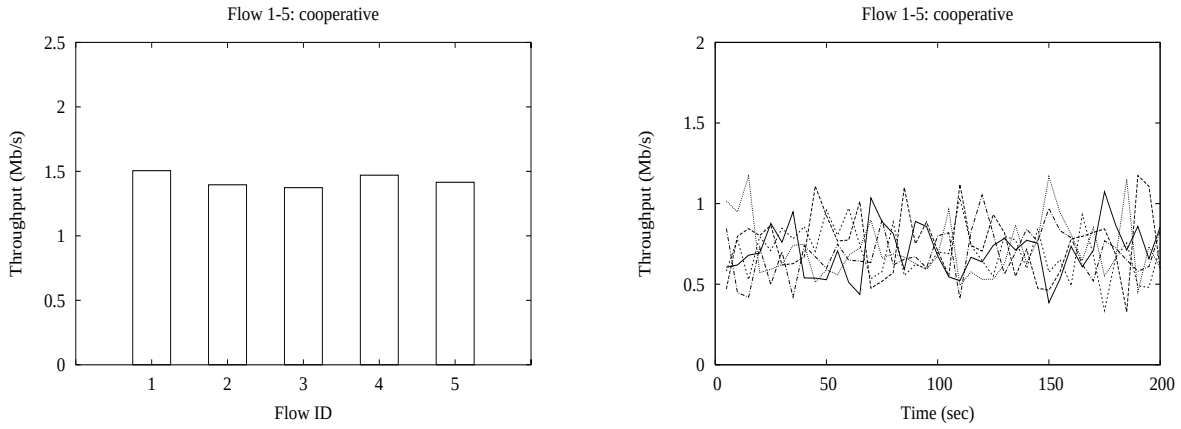


Figure 33: Alice and Bob's throughput matrix for the congestion control game. C: cooperate, N: not cooperate. Throughput is measured in Mbps.

not, the selfish party achieves a disproportionate share of 9 Mbps. When both are noncooperative, the system becomes congested with each receiving only 3 Mbps for a total throughput of 6 Mbps. In the throughput matrix example, Alice, if selfish and "rational," will choose the noncooperative strategy since her payoff—irrespective of Bob's action—always exceeds the corresponding throughput achievable by choosing the cooperative strategy: 9 Mbps vs. 5 Mbps when Bob chooses C, and 3 Mbps vs. 1 Mbps when Bob chooses N. Strategy N dominates strategy C. By symmetry, the game played by Alice and Bob results in the strategy profile (N,N) with payoff (3,3), which is strictly less than the "welfare" (5,5) attainable through cooperation.

Another example, in the context of TCP, is the throughput sharing behavior of multiple TCP sessions. Figure 34(a) shows the throughput achieved by five TCP sessions traversing a common bottleneck link. TCP, being cooperative, backs off when it thinks that a packet loss has occurred. This prevents congestion and promotes equitable sharing of resources when other extraneous factors such as distance—a long-haul TCP session is in a disadvantaged position vis-à-vis a short-haul session—are ignored. Figure 34(b) shows the corresponding time dynamics of the bandwidth sharing behavior. Figure 35(a) shows the throughput attained by five TCP flows when one of them, Flow 5, implements a variant of TCP that incorporates a measure of selfishness. Upon detecting potential packet loss, "TCP-greedy" does not initially back off. Expecting other cooperative flows sharing the bottleneck to act gentlemanly, TCP-greedy persists at a larger time scale soaking up the bandwidth abandoned by cooperative TCP flows. When throughput eventually degrades, TCP-greedy backs



(a) (b)

Figure 34: TCP dynamics: cooperative bandwidth sharing behavior. (a) Throughput share. (b) Time dynamics.

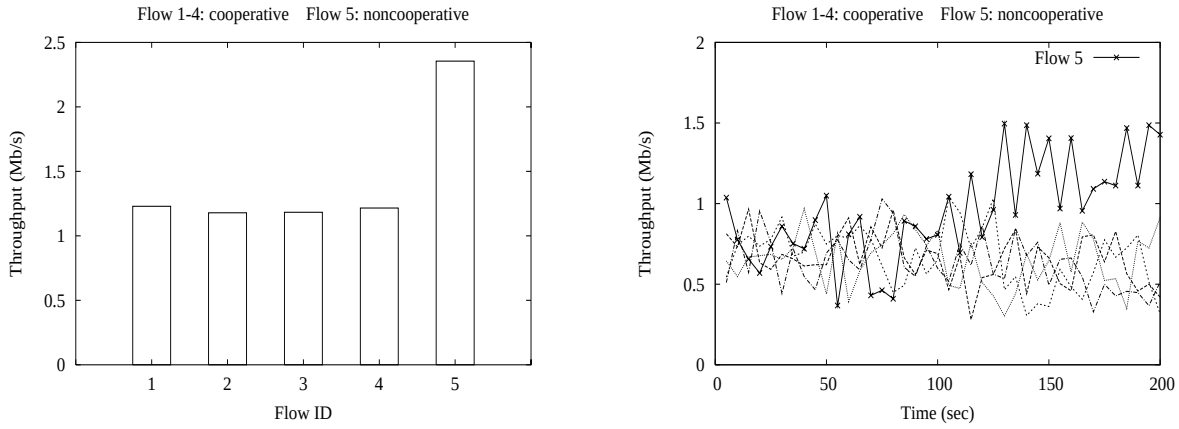
off assuming that the losses are due to self-congestion. Figure 35(b) shows the time dynamics of a simple instance of TCP-greedy where until time 100 second the protocol operates cooperatively; at time 100 second, it toggles into greedy mode, which illustrates the contrast in bandwidth sharing behavior. We note that selfishness amidst cooperation can be employed as a means of denial of service attack.

Congestion control games are single service class games where a shared resource is accessed through a single service class (i.e., as is) representing the resource. The decision variable is: how much traffic to submit to the service class or resource. In multiple service class games, a shared resource is accessed through a multiple service class abstraction, either because the underlying resource is actually a collection of distinct resources or a single resource is virtually divided into multiple resources through scheduling. The decision variables entail selecting service classes in addition to determining the traffic volume submitted. Multi-class network games are versatile—many network resource sharing problems can be cast as multi-class games—and possess a rich structure. They are discussed in Section 5.3.

5.2 Single class noncooperative network game: congestion control

5.2.1 Equilibria and optima

Consider a binary congestion control game where Alice can send "excessive" offered load L_{N} or an "appropriate" traffic load L_{C}. With L_{N} and L_{C} replacing N and C, assume that the throughput matrix in Figure 33 applies. In its continuous form—discussed in Section 5.3—L_{C} and L_{N} are but two values of a traffic control variable $\lambda \in \mathbb{R}_+$. One of the first considerations, when analyzing dynamical systems—even the one-shot congestion control game—is the issue of stability. The configuration (N,N) with payoff (3,3) is an equilibrium in the sense that from (N,N) neither Alice nor Bob, acting selfishly and unilaterally, have an incentive to deviate since the payoff of the changing party would decrease to 1. Configurations conditioned on which unilateral strategy changes do not improve the changing party's individual utility are called Nash equilibria. In the 2-party binary



(a) (b)

Figure 35: TCP dynamics: noncooperative bandwidth sharing behavior. (a) Throughput share. (b) Throughput dynamics—at time 100 second greedy action is instituted.

congestion control game—equivalent to Prisoner's Dilemma—strategy profile (N,N) is the unique Nash equilibrium. In general network resource games, including multi-class network games, a Nash equilibrium need not exist, much less be unique. Defining the system utility to be the sum of individual utilities, configurations (C,C), (C,N), and (N,C) are system optimal with total utility 10, whereas (N,N) with system utility 6 is not. A more refined, welfare oriented notion of efficiency is Pareto optimality, where a configuration is Pareto optimal if the system utility cannot be improved without sacrificing the utility of one or more players. Thus (N,N) is not Pareto optimal since the configuration (C,C) improves both players' payoff; (C,C), (C,N), and (N,C) are Pareto optimal. It is easily checked that system optimality implies Pareto optimality; other implications, in general, do not hold. From a system and Pareto optimality perspective, configurations (C,C), (C,N), and (N,C) are equally good. Additional fairness criteria such as equal share may be imposed to further distinguish between desirable and undesirable system states.

5.2.2 Pricing

The binary congestion control game, being an instance of Prisoner's Dilemma, does not bring forth fundamental new issues when considered in the Internet context. However, it admits networking driven variations on the game with potential practical relevance. One case in point is pricing. By introducing usage pricing—the larger the bandwidth or data rate consumed, the higher the economic cost assigned by an ISP to the user—the dominance of noncooperative strategy N over cooperative strategy C may be removed. Supposing transferring a file at 1, 3, 5, and 9 Mbps costs a, b, c, and d dollars ($a < b < c < d$), respectively, if Alice derives ordinal satisfaction

$$\langle 9 \text{ Mbps}, d\rangle \prec \langle 5 \text{ Mbps}, c\rangle \prec \langle 3 \text{ Mbps}, b\rangle \prec \langle 1 \text{ Mbps}, a\rangle$$

then strategy C becomes dominant over N. If Bob is equally cost-conscious and in no particular hurry, the solution to the pricing enhanced game with utility matrix shown in Figure 36 becomes

(C,C). In the case when Bob is a wealthy speed-junky with utility preference

$$\langle 1 \text{ Mbps}, a\rangle \prec \langle 3 \text{ Mbps}, b\rangle \prec \langle 5 \text{ Mbps}, c\rangle \prec \langle 9 \text{ Mbps}, d\rangle$$

the solution of the game becomes (C,N) with throughput allocation (1,9). Bob is happy because he gets the fastest service, whereas Alice is happy because she gets the most economic service. The ISP is content because it only performs metering and leaves the headache of resource contention resolution to users. The ISP would be happy if $a + d > 2c$. Pricing, when treated as an engi-

Alice \ *Bob*	C	N
C	<5,c>, <5,c>	<1,a>, <9,d>
N	<9,d>, <1,a>	<3,b>, <3,b>

Figure 36: Alice and Bob's throughput-price matrix for the congestion control game with pricing $a < b < c < d$. C: cooperate, N: not cooperate.

neering tool, has the potential of steering away an otherwise self-defeating system from the peril of the tragedy of the commons [62]. Pricing, however, is not a magical wand and has limitations that can curtail its effectiveness. For example, pricing schemes that depend on accurate knowledge of users' utilities may be construed as unrealistic given the not so well-understood nature of human preference. In some cases, pricing need not influence the outcome of a game. In the congestion control game, if both Alice and Bob are speed-junkies with unlimited spending power—e.g., $\langle 1 \text{ Mbps}, a\rangle \prec \langle 3 \text{ Mbps}, b\rangle$ and $\langle 5 \text{ Mbps}, c\rangle \prec \langle 9 \text{ Mbps}, d\rangle$ for all $a < b < c < d$—pricing does not prevent the congestion outcome (N,N).

5.2.3 Repeated games

Another direction in which the congestion control game can be looked at is as an iterated variant that injects a notion of time. Instead of playing the game in a single-shot fashion, a multiple round game is played where the players' future strategy is allowed to be influenced by the past. The Iterated Prisoner's Dilemma, highlighted in the influential work of Axelrod [8, 9], captures the problem of what time-dependent strategies are effective and, in an evolutionary twist, competitively fit when subject to evolutionary pressures. In tournaments where strategies were pitted against each other [8, 7], it was observed that one particular strategy, called tit-for-tat, outperformed the competition. In tit-for-tat, the strategy starts out cooperative and remains so as long as the other party reciprocates. If the other party switches to a noncooperative mode, tit-for-tat does the same. If the other party ever switches back to a cooperative mode, tit-for-tat is forgiving and reverts back as well. An unforgiving variant, called grim trigger, does not. In finitely repeated Prisoner's Dilemma where the total payoff is the sum of the payoffs in each round, self-optimizing players will

play the noncooperative strategy N in every round. If r is the final round, then no matter what has transpired in previous rounds, N is dominant over C in the final round by its finality—there are no future consequences for noncooperation. Applying the argument recursively to round $r-1$ yields the all-N strategy. In an infinitely repeated Prisoner's Dilemma where the boundary condition of finite r is removed and future payoffs discounted by a multiplicative factor $0 < \delta < 1$, both tit-for-tat and grim trigger are self-optimizing strategies when δ is not too small. The two strategies induce an incentive for cooperation even when players are selfish. An excellent overview of repeated games can be found in [99]. An introduction to evolutionary games is provided in [141].

In the Internet context, a single TCP-greedy session can exploit the cooperative nature of other TCP flows and grab a lion's share of network bandwidth. When many users employ greedy variants of TCP, the network may get trapped in a congested state. A survival of the fittest, "Wild Wild West" environment is not inconceivable. Technologically, deploying new transport protocols is not difficult since it only entails changes at the sender and receiver sides. The network core, due to Internet's end-to-end design paradigm, is not affected. The outlook, however, is not singularly bleak. As indicated in the discussion on self-similar traffic, most TCP sessions are short-lived due to the heavy-tailed nature of file sizes. Short-lived sessions tend to operate in an aggressive mode called slow start—a misnomer given the fast nature of the start-up process—and rarely reach steady-state at which cooperativeness is most readily exposed. Also, a 100 msec small file transfer, even if 10-fold delayed due to greedy actions of others, completes in 1 second which, on a human time scale, is not overly significant. A large file transfer with a 1 minute duration, if delayed 10-fold, however, is a different matter. On the architectural front, users with broadband Internet connectivity are limited by an access speed of 1–6 Mbps which bounds the damage that a single user can exact on aggregate backbone traffic. For cellular data services, price based controls, including usage pricing, are instituted to shield the low bandwidth cellular network from overutilization. Last, but not least, most TCP transfers are HTTP requests by clients to Web servers. As such, data flows downstream from server to client, and unless the server side TCP is modified, it is not easy to transform a TCP session into greedy mode with client side TCP modifications alone. An elegant treatment of congestion control from a distributed control and optimization perspective is provided in [74].

5.3 Multi-class noncooperative network games

5.3.1 Multi-class QoS provisioning game

Consider a multiple service class resource provisioning system much like a multi-lane highway with m toll booths that process arriving cars. The service quality—in the toll booth example, delay—received by an automobile depends on several factors including which toll lane the driver has joined and the line of cars waiting in front. All else being equal, the longer of line the longer the wait. In the Internet context, service may also be affected by a scheduler that apportions resources to the m classes according to a service policy. In the toll booth example, suppose a single person is manning all booths, running from booth to booth collecting tolls. If toll lanes are differentiated by the number of passengers in a car inclusive the driver—e.g., lane 4 for cars with four or more passengers, lane 3 for three passengers, and so forth—and the toll booth attendant implements a priority scheduling policy where cars in lane 3 are serviced only if lane 4 is empty, cars in lane 2 are serviced only if lanes 3 and 4 are empty, and so forth, then the quality of service received by a lone commuter may be less than that received by commuters in a car pool. In queueing theory, the equilibrium waiting time experienced by n customers in the m classes is studied as a function of

the stochastic property of the customer arrival process and the scheduling policy implemented by the server [21]. In the network game context, we allow the n users to pick the service class to which their packets are submitted and focus on the outcome of the resultant n-player game. The impact of stochastic arrivals—a challenging problem for non-Markovian input in its own right—and time varying decision making by players is not explicitly considered except "in equilibrium" through static analysis.

Formally, a m-class n-player noncooperative network game comprises of m service classes managed by a scheduler $\mathcal{S}$ and n players with traffic demand $\lambda_1, \lambda_2, \ldots, \lambda_n$ who choose one or more service classes via decision variables $\lambda_{ij} \geq 0$, $\lambda_i = \sum_{j=1}^{m} \lambda_{ij}$. A performance function φ—depending on $\mathcal{S}$ and the input—determines the quality of service $\varphi_1, \varphi_2, \ldots, \varphi_m$ rendered in the m service classes as a function of aggregate traffic $\mathbf{q} = (q_1, q_2, \ldots, q_m)$ where $q_j = \sum_{i=1}^{n} \lambda_{ij}$. The QoS received by the ith player is determined by his traffic allocation vector $\boldsymbol{\lambda}_i = (\lambda_{i1}, \lambda_{i2}, \ldots, \lambda_{im})$ and the choices made by other players. We assume $\partial\varphi_j/\partial q_j \geq 0$, i.e., all else being equal, the more stress on a service class the worse the QoS—e.g., delay or packet loss rate—rendered in that class. Assuming φ_j denotes packet loss rate, the throughput of service class j is given by $q_j(1 - \varphi_j)$. Pricing, if present, assigns a per unit flow cost p_j to each class. $\sum_j p_j \lambda_{ij}$ is the total cost incurred by user i and $\sum_j p_j q_j$ an ISP's total revenue. User i is endowed with a utility function U_i that depends on the traffic transmitted, QoS received, and cost. As a noncooperative game, user i's strategy set is given by the values of $\boldsymbol{\lambda}_i$, the system's strategy profile is $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_n)$, and user i aims to optimize his utility. A profit maximizing service provider who sets the price vector $\mathbf{p} = (p_1, p_2, \ldots, p_m)$ may explicitly enter into play as player $n + 1$.

5.3.2 Applications

A number of problems in networking can be mapped to the multi-class QoS provisioning game. A network access provider may furnish prioritized service classes—e.g., platinum, gold, silver, bronze, and best-effort—that users can select depending on their application needs. For example, Alice, as an investment analyst, may use the gold class to browse the web and carry out e-commerce transactions at work, while using the bronze and best-effort classes at home for casual use. Bob, a financially strapped graduate student, may predominantly use best-effort service for Internet access except when videophoning Alice, at which time he uses platinum service. The INDEX project [5, 122] provides an interesting study of demand elasticity with respect to bandwidth pricing in an experimental multi-class access network. A network content provider or web service provider may use a multi-class service abstraction to schedule CPU cycles for exporting prioritized services to client requests. An enterprise network or transit provider may use multi-class packet forwarding to provide differentiated services to its customer base. Provisioning end-to-end QoS over a network of multi-class routers using game theoretic mechanisms is discussed in [27, 28]. In multi-path routing, a set of routes—preferably disjoint—from source to destination is used to transmit traffic. In [98], game theoretic aspects of "parallel" routing are studied. By mapping each path in the parallel routing game to a service class, we arrive at an equivalent multi-class network game where the scheduling components $\varphi_1, \ldots, \varphi_m$ are mutually decoupled. Technically, the scheduler is a non-work conserving weighted fair queue (WFQ) with service weights $\alpha_1, \ldots, \alpha_m$, $\sum_j \alpha_j = 1$, that determine the bandwidth of each route/service class.

5.3.3 Key issues and analysis tools

Key issues, when studying a multi-class QoS game, include: Do multi-class QoS games possess Nash equilibria (NE)? If they do, are they efficient? Can pricing help drive the system to a desirable network state such as a system optimal NE? How are stability and optima affected by the shape of user utilities? How does scheduling impact the game? These are but a subset of questions with practical relevance.

Before we proceed with a discussion of game theoretic properties of multi-class QoS provisioning, we give a brief overview of some facts and analysis tools. First, in the following we consider only games with pure strategies, the default case implicitly assumed thus far. For example, in the binary congestion control game—a 2-player finite game where finite refers to the cardinality of the strategy set $\{\mathrm{C}, \mathrm{N}\}$—mixed strategies would admit a probability distribution over $\{\mathrm{C}, \mathrm{N}\}$ with the interpretation that a player probabilistically chooses C or N. In the Internet QoS provisioning context, we do not believe that mixed strategies are practically meaningful. In so doing, however, we forgo one of the few niceties available in noncooperative games—the existence of a Nash equilibrium in finite noncooperative games with mixed strategies—a result established by Nash [94, 95]. Indeed, as we shall see, multi-class QoS games need not have a Nash equilibrium. This is not unusual since it is known that large finite games with random payoff functions, more often than not, do not possess a NE in pure strategies [43]. Mixed strategies are represented by simplices whose faces are spanned by corner points that correspond to pure strategies. As such, optimization tends to be easier under mixed strategies. For noncooperative games in pure strategies, a powerful result exists that assures the existence of a Nash equilibrium as long as the game is sufficiently nice in two respects: one, the strategy set of each player is nonempty, compact, and convex, and two, the utility function of every player i is continuous and quasi-concave in the ith decision variable $\boldsymbol{\lambda}_i$. Recall that a function $f(x)$ is quasi-concave if for all a the upper level set $\{x : f(x) \geq a\}$ is convex. Satisfying the two conditions allows Kakutani's fixed point theorem—a generalization of Brouwer's fixed point theorem to point-set maps, i.e., correspondences—to be applied yielding a fixed point to every user's self-optimizing action, called the best-reply correspondence. Recalling the definition of NE (cf. Section 5.2), configuration $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_i, \ldots, \boldsymbol{\lambda}_n)$ is a Nash equilibrium if for all $\boldsymbol{\lambda}_i'$

$$U_i(\boldsymbol{\lambda}) \geq U_i(\boldsymbol{\lambda}')$$

where $\boldsymbol{\lambda}' = (\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_i', \ldots, \boldsymbol{\lambda}_n)$. The maximizer $\boldsymbol{\lambda}_i'$, given a strategy profile $\boldsymbol{\lambda}$—i.e., best-reply correspondence—need not be unique leading to a point-set correspondence where Kakutani's fixed point theorem can come into play. In [121] it is shown that concave games with an additional requirement—diagonal strict concavity—possess a unique Nash equilibrium. The general sufficiency result for NE existence in concave games goes back to [40, 51, 60].

5.3.4 Stability and efficiency

We will first consider a multi-class analogue of Orda et al.'s parallel routing game [98] focusing on the role of utility functions, followed by the effect of pricing and scheduling in subsequent sections. As indicated earlier, the parallel routing game with m disjoint routes maps to an equivalent m-class QoS game where the scheduler $\mathcal{S}$ is trivial in the sense that the service classes are mutually decoupled. Let $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m)$, $\sum_j \alpha_j = 1$, denote the relative bandwidth of the m service classes. A unit flow in class j receives resource share $\omega_j = \alpha_j / q_j$. We assume that QoS is determined by the per unit flow resource a traffic flow receives. That is, given $\boldsymbol{\lambda}$ and $\boldsymbol{\alpha}$, $\varphi_j \leq \varphi_{j'}$ if $\omega_j \geq \omega_{j'}$. A small value of φ means superior QoS. We consider utility functions motivated by certain multimedia

applications. A VoIP application cannot tolerate delays more than 200 msec if it is to achieve a perceptually acceptable level of service. A real-time streaming video application cannot tolerate packet loss rates above a certain level, due to the negative repercussions on video quality stemming from undecodable video frames. A scenario that does not involve multimedia: a user running a supercomputing application needs to transfer a 125 GB file in less than 3 hours. This requires a bandwidth of at least 90 Mbps to achieve the QoS target. In all three cases, there is a sharp transition near a threshold such that QoS worse than the threshold is of little use and QoS better than the threshold brings marginal additional benefit: VoIP conversations sound clear whether packets are delayed 5 or 50 msec. Incorporating this property of QoS-sensitive applications, we consider step utility functions where the utility of user i has the shape

$$U_i(c) = \begin{cases} 1, & \text{if } c \leq \theta_i; \\ 0, & \text{otherwise.} \end{cases}$$

$\theta_i \geq 0$ is a QoS threshold capturing i's preference and $c \geq 0$ is the QoS experienced. Assuming player i is allowed to split her total traffic λ_i among one or more classes—we also consider unsplittable games where all traffic belonging to user i must be sent to one class—we define user i's combined utility as $\bar{U}_i(\boldsymbol{\lambda}) = \sum_{j=1}^{m} \lambda_{ij} U_i(\varphi_j(\boldsymbol{\lambda}))$.

One more technical set-up before we proceed to Nash equilibria. Since the service classes are decoupled and $\varphi_j(q_j)$ nondecreasing in q_j, assuming φ_j is continuous we can express $U_i(c)$ directly in terms of a critical traffic threshold b_{ij} by the invertibility of φ_j:

$$U_i(\varphi_j(q_j)) = \begin{cases} 1, & \text{if } q_j \leq b_{ij}; \\ 0, & \text{otherwise.} \end{cases}$$

Figure 37 shows an example utility function for a 2-class system where player i's QoS threshold θ_i maps to traffic thresholds b_{i1} and b_{i2} in classes 1 and 2, and total traffic demand is given by $\lambda_i = 3b_{i1} = 3b_{i2}$. Figure 37(a) shows utility function U_i for class $j \in \{1, 2\}$, Figure 37(b) shows the corresponding weighted utility $\lambda_{ij} U_i$, and Figure 37(c) shows the combined utility $\bar{U}_i$. Figure 37(c)

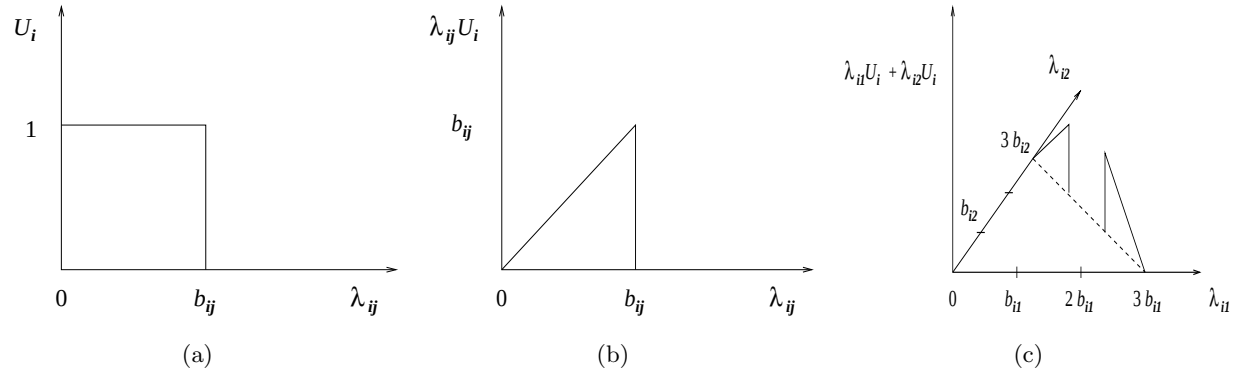


Figure 37: Utility function. (a) Single utility. (b) Weighted utility. (c) Combined utility.

shows that player i's utility is not quasi-concave in its decision variable $\boldsymbol{\lambda}_i$—the upper level set for small a has a hole in the middle—violating the sufficiency condition needed to apply Kakutani's fixed point theorem. Not meeting the sufficiency condition for concave games does not imply that

Nash equilibria do not exist. In [104] it is shown that even in 2-player 2-service class systems, Nash equilibria fail to exist under "mild" conditions. Thus in these systems, every configuration has at least one player who thinks he can do better, preventing the system from reaching a quiescent state. The NE existence problem in the multi-class QoS game stems from two factors: multiple service classes—a distinguishing feature from the single-class congestion control game—and splitting of user traffic. It can be shown that if traffic splitting is disallowed, the multi-class QoS game always has a Nash equilibrium.

The next question concerns the relationship between Nash equilibria, Pareto optima, and system optima. In particular, assuming NE exist, are they also efficient? The simple answer is no. The three configuration classes—NE, Pareto optima, and system optima—for multi-class QoS games are well-separated, except for the trivial inclusion of system optima in Pareto optima. That is, there are NE that are not system optimal, a NE that is Pareto optimal need not be system optimal, a system optimum need not be a Nash equilibrium, and there are Pareto optima that are not system optimal. The structural relationship between the three classes is depicted in Figure 38. In [104] an

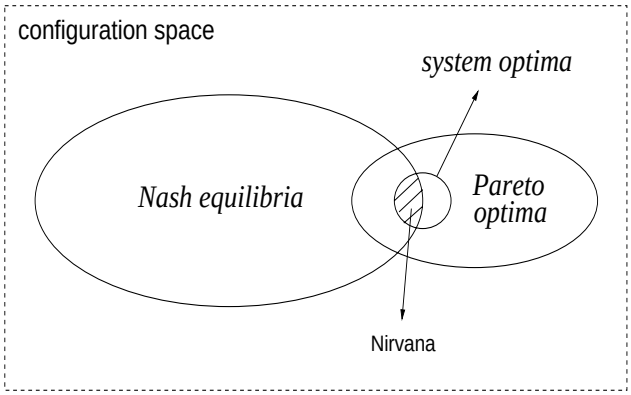


Figure 38: The structural relationship between three configuration classes—Nash equilibria, Pareto optima, and system optima—in multi-class QoS games.

exact characterization is given as to when Nash equilibria are Pareto or system optimal. A useful tool is the normal form of a configuration which establishes an equivalence relation between system optimality and Pareto optimality: a configuration $\boldsymbol{\lambda}$ is system optimal if and only if its normal form $\boldsymbol{\lambda}'$ is Pareto optimal. Through this linkage, system optimality of a configuration can be checked by verifying Pareto optimality of its normal form. Lastly, there are subclasses multi-class QoS games, called resource-plentiful games, in which NE, Pareto optima, and system optima collapse into a single class.

5.3.5 A "darker side" of pricing

We illustrate that pricing in multi-class QoS games need not lead to desirable outcomes, in particular, we show that pricing can disrupt stability. In the parallel routing multi-class QoS game, consider pricing policies such that a traffic flow that receives better QoS—hence more resources per unit flow—pays a higher price than another that does not. Since service classes are decoupled, the family of pricing functions that satisfy this property are monotone functions

$$\omega_j > \omega_{j'} \;\Leftrightarrow\; p_j > p_{j'}, \quad j \neq j'. \tag{7}$$

Such pricing functions are transparent, accurate, and fair—you pay for what you get. Now, consider a 2-class 2-player unsplittable QoS game where $b_{1j} < b_{2j}$ for $j \in \{1, 2\}$ (player 1 has a stricter QoS requirement than player 2), $\max\{\lambda_1, \lambda_2\} < \min\{b_{11}, b_{12}\}$ (both can be happy if one to each class), and $b_{1j} < \lambda_1 + \lambda_2 < b_{2j}$ for $j \in \{1, 2\}$ (if both are in the same class then player 1 will be unhappy while player 2 remains happy). Figure 39 shows an instance of the 2-class 2-player game. The

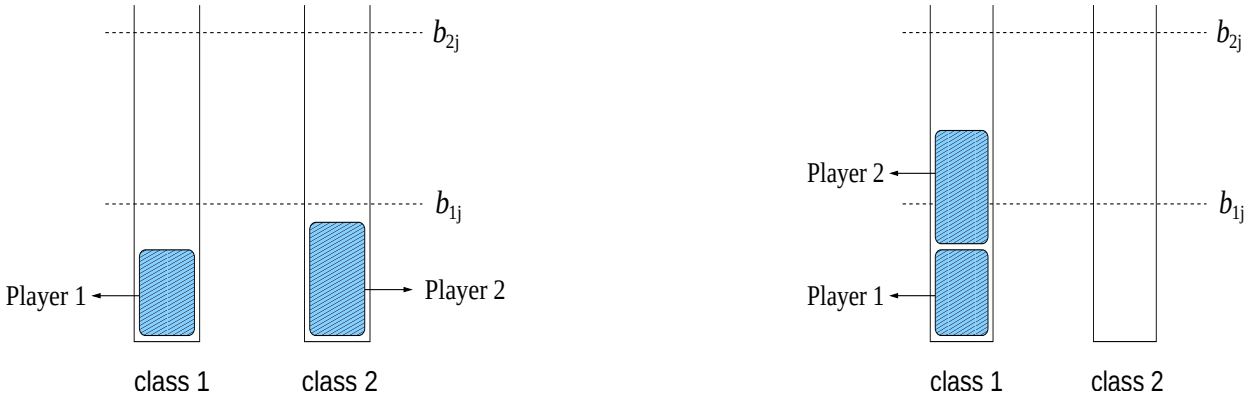


Figure 39: Detrimental effect of pricing in 2-class 2-player QoS game. (a) Nash equilibrium without pricing but unstable with pricing. (b) Configuration after player 2's selfish move; player 1 becomes unhappy.

configuration in Figure 39(a) corresponds to a Nash equilibrium of the 2-class 2-player QoS game without pricing. Both players are happy and there is no incentive to move. When monotone pricing is introduced, however, the configuration ceases to be a Nash equilibrium since player 2 can move to service class 1 and still be happy QoS-wise while paying less money-wise: $p_1 < p_2$ since $\omega_1 < \omega_2$. This is shown in Figure 39(b). Player 1's QoS, however, is violated which prompts player 1 to move to class 2. From here on the cycle repeats. This example illustrates that any usage-oriented—in the sense of (7)—engineering application of pricing turns the QoS game with system optimal NE into one that is not even stable. Assuming utilities that dictate a user minimize cost subject to achieving QoS happiness, instability resulting from the "chasing a bargain and crowding" effect is inherent and can only be eliminated by violating (7). The step function assumption on utilities can be relaxed without affecting the outcome of the game. The next section shows the beneficial effect of scheduling.

5.3.6 Influence of scheduling

In the example shown in Figure 39, it is the excess capacity—hence cost—apportioned to player 2's flow when residing in class 2 that drives it to seek a more economic solution by migrating to class 1. As long as capacity in the service classes is fixed, the problem is unavoidable. Such is the case in the parallel routing game over disjoint paths: bandwidth across paths is not transferable. This is not the case in bandwidth scheduling at routers and CPU scheduling at servers where a single physical resource—bandwidth or CPU—is shared among multiple flows under the auspices of a scheduler. In the 2-class 2-player QoS game with pricing, by removing the excess bandwidth

from class 2 when player 2 is present, the gap $b_{22} - \lambda_2 > 0$ can be narrowed so that no other service class offers a more economic solution. In multi-class QoS games where service classes are ordered according to their QoS, the game structure becomes much nicer [116]. Nash equilibria are assured, and under "mild" conditions on utility, Nash equilibria become system optimal. The sufficiency condition on QoS ordering, called (A1), (A2), and (B) [116], is satisfied by the priority scheduler mentioned in the toll booth example. Both QoS and usage-based price ordering (7) across service classes is assured, and the resultant monotonicity helps break cyclic chain reactions that harm stability and efficiency.

A problem with priority scheduling is that lower priority classes may get starved when high priority traffic is abundant. WFQ, which assigns a fraction of the shared resource to classes in accordance with service weights, prevents starvation but does not satisfy properties (A1)–(B) needed for stable and efficient resource provisioning. It turns out that there is an optimal multi-class scheduler that satisfies the ordering properties and is maximally efficient in the mean-squared error (MSE) sense [116]. Intuitively, the optimal aggregate-flow scheduler is like a WFQ whose weights are dynamically set, i.e., the weights are a function of the input. In [119] an implementation of the optimal scheduler in IOS—Cisco's router operating system—and its benchmarking on a network of Cisco 7200 series routers is discussed. A corresponding simulation study is available in [118]. In [117] a queueing based treatment of optimal aggregate-flow scheduling is presented. Finally, we remark that when multi-dimensional QoS vectors are considered—e.g., delay, packet loss rate, delay jitter, packet loss jitter, and bandwidth—a total order on the service classes is not guaranteed [29]. In general, only a partial order holds. For example, depending on the input and scheduling discipline, a high priority class with small delay or packet loss rate may experience a larger variance than a low priority class with larger delay or packet loss rate. Initial observations on multi-class QoS games with multi-dimensional QoS vectors may be found in [104].

5.4 Discussion

A game theoretic view of the Internet is interesting because a significant segment of its daily activity is carried out by network protocols, software that act on behalf of human users. The sometimes unpredictable human element is delimited by this layer of automated agents whose behavioral rules, for the most part, are transparent and remain invariant for months and years. Playing out game theory on the Internet provides a new opportunity to advance theory as well as inject much needed effectiveness into a largely thought experiment driven, unverifiable subject matter. The scale of the Internet and its end-to-end design paradigm facilitate experimental design and quantitative studies where measurement based analysis of cause and effect can put the predictive power of theories to the test. This may complement the small scale experiments with human and animal subjects available today. Perhaps in a manner similar to self-similar Internet traffic and power-law Internet topology that have provided fresh jolts and experiential grounding to queueing theory and random graph theory, respectively, the Internet may provide a platform to further advance game theory. What is, as yet, missing is the identification of a phenomenological basis—long-range dependence in queueing and power-law connectivity in random graph theory—that directly connects with the foundations of game theory. Perhaps a quantification of the tragedy of the commons—very large or very small—and its linkage to "bounded rationality" and cooperativeness of present day protocols may be avenues for further exploration.

The discussion of network games presented is far from complete, omitting a number of influential works including market-based treatment of concave resource economies [53, 52, 123], noncooperative congestion control analyses [77, 78], pricing, flow control, and admission control in multi-class

networks [34, 45, 65], among others. A more comprehensive overview of related works can be found in [104]. We conclude by noting that two works stand out as milestones in the application of game theoretic ideas, including pricing, to network resource allocation. The Spawn system [139] by Huberman et al. at Xerox PARC in the early 1990s remains as one of the few efforts that have put microeconomic mechanisms to the test in a real working environment. The load balancing measurements enabled by Spawn have acted as a proof of concept for later works. In a similar vein, the INDEX project [122] by Varaiya et al. has served an important role by demonstrating the much debated elasticity of humans—at least a 70+ population of students and faculty at Berkeley—with respect to access network bandwidth pricing.

6 Concluding Remarks

This article has attempted to provide a discussion of three subject areas—self-similar traffic, power-law connectivity, and noncooperative network games—whose underlying themes touch upon the "Internet as a Complex System" metaphore through synergistic but scientifically grounded efforts. The other two subject areas—scalable traffic control and organizational behavior—although alluded to in the overall discussion, have been omitted due to time and space constraints. A complex systems viewpoint of the sciences, including natural and engineered phenomena, has proved both successful and unsuccessful. Successful because the ideas and challenges have helped the development of other sciences such as dynamical systems theory in mathematics. Also, many of the concepts and techniques—fractals, dynamics, randomness, automata, correlation at a distance, phase transition, to mention a few—have entered into other areas and some into the everyday vocabulary of society at large. Unsuccessful because "complex systems" has not had a problem domain—at least not one that has withstood the test of time—that it can claim as its own. A problem domain that, perhaps, is amenable to an arsenal of "complex systems tools and methods" that are not necessarily the purview of other sciences. This may be a good thing, but also a tad unsatisfactory because few serious scientists would venture to say that they work in the area of complex systems since it is unclear what that would exactly mean. It is too early to say if the Internet as a Complex System metaphore has depth and requisite legs. One positive thing it has going is that the subject matters discussed in the article were developed in other areas, but they seem to require a unified understanding of the phenomena and involve synergistic application of advanced tools with new twists. Some have started using the term "Internet Science" to refer to the vast menagerie of Internet related technologies, phenomena, and issues. We submit that a common thread to the varied synergistic activities underlying the Internet may be a complex systems viewpoint, meaning that the whole—in the Internet context—is greater than its parts, and to understand the varied aspects of the Internet requires new glues to bond the pieces together that cannot be soley supplied solely by the individual subareas.

References

[1] J. Abello, A. Buchsbaum, and J. Westbrook. A functional approach to external graph algorithms. In *Proc. 6th European Symposium on Algorithms*, pages 332–343, 1998.

[2] Norm Abramson. The Aloha system—another alternative for computer communcations. In *Proc. Fall Joint Comput. Conf. AFIPS Conf.*, pages 281–285, 1970.

[3] W. Aiello, F. Chung, and L. Lu. A random graph model for massive graphs. In *Proc. ACM STOC '00*, pages 171–180, 2000.

[4] R. Albert, H. Jeong, and A. Barabási. Diameter of the World Wide Web. *Nature*, pages 130–131, 1999.

[5] J. Altmann, B. Rupp, and P. Varaiya. Internet demand under different pricing schemes. In *Proc. ACM Electronic Commerce*, pages 9–14, 1999.

[6] S. Asmussen, K. Binswanger, and B. Hojgaard. Rare events simulation for heavy- tailed distributions. *Bernouille*, 6:303–322, 1991.

[7] R. Axelrod and W. Hamilton. The evolution of cooperation. *Science*, 211:1390–1396, 1981.

[8] Robert Axelrod. Effective choice in the Prisoner's dilemma. *Journal of Conflict Resolution*, 24:3–25, 1980.

[9] Robert Axelrod. *The Evolution of Cooperation*. Basic Books, New York, 1984.

[10] Y. Azar, A. Fiat, A. Karlin, F. McSherry, and J. Saia. Spectral analysis of data. In *Proc. ACM STOC '01*, pages 619–626, 2000.

[11] Norman Bailey. *The Mathematical Theory of Infectious Diseases*. Oxford University Press, New York, 1987.

[12] A. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, pages 509–512, 1999.

[13] Richard Bellman. On a routing problem. *Quarterly of Applied Mathematics*, (1):87–90, 1958.

[14] C. Bennett and D. DiVincenzo. Quantum information and computation. *Nature*, 404:247–255, 2000.

[15] C. Bennett and S. Wiesner. Communication via one- and two-particle operators on Einstein-Podolsky-Rosen states. *Phys. Rev. Lett.*, 69:2881–2884, 1992.

[16] Jan Beran. A test of location for data with slowly decaying serial correlations. *Biometrika*, 76:261–269, 1989.

[17] Jan Beran. *Statistics for Long-Memory Processes*. Monographs on Statistics and Applied Probability. Chapman and Hall, New York, NY, 1994.

[18] B. Bollobás, O. Riordan, J. Spencer, and G. Tusnády. The degree sequence of a scale-free random graph process. *Random Structures and Algorithms*, 18(3):279–290, 2001.

[19] O. Boxma and J. Cohen. The single server queue: Heavy tails and heavy traffic. In K. Park and W. Willinger, editors, *Self-Similar Network Traffic and Performance Evaluation*, chapter 6. Wiley-Interscience, 2000.

[20] Dietrich Braess. Über ein paradoxon aus der verkehrsplanung. *Unternehmensforschung*, pages 258–268, 1969.

[21] A. Brandt, P. Franken, and B. Lisek. *Stationary Stochastic Models*. John Wiley & Sons, 1990.

[22] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, and R. Stata. Graph structure in the Web. *Computer Networks*, 33:309–320, 2000. Proc. 9th WWW Conference.

[23] CAIDA. Caida analysis of Code-Red, 2001. http://www.caida.org/analysis/security/code-red.

[24] CAIDA. skitter, 2002. http://www.caida.org/tools/measurement/skitter.

[25] Computer Emergency Response Team (CERT). CERT Advisory CA-1999-04 Melissa Macro Virus, March 1999. http://www.cert.org/advisories/CA-1999-04.html.

[26] Computer Emergency Response Team (CERT). CERT Advisory CA-2001-19 "Code Red" worm exploiting buffer overflow in IIS indexing service DLL, July 2001. http://www.cert.org/advisories/CA-2001-19.html.

[27] S. Chen and K. Park. A distributed protocol for multi-class QoS provision in noncooperative many-switch systems. In *Proc. IEEE International Conference on Network Protocols*, pages 98–107, 1998.

[28] S. Chen and K. Park. An architecture for noncooperative QoS provision in many-switch systems. In *Proc. IEEE INFOCOM '99*, pages 864–872, 1999.

[29] S. Chen, K. Park, and M. Sitharam. On the ordering properties of GPS routers for multi-class QoS provision. In *Proc. SPIE International Conference on Performance and Control of Network Systems*, pages 252–265, 1998.

[30] J. Chuang and M. Sirbu. Pricing multicast communication: A cost based approach. In *Proc. INET '98*, 1998.

[31] F. Chung and L. Lu. The average distance in random graphs with given expected degrees. *Proc. Nat. Acad. Sci.*, 99:15879–15882, 2002.

[32] F. Chung and L. Lu. Connected components in a random graph with given degree sequences. *Annals of Combinatorics*, 6:125–145, 2002.

[33] David Clark. The design philosophy of the DARPA internet protocols. In *Proc. ACM SIGCOMM '88*, 1988.

[34] R. Cocchi, D. Estrin, S. Shenker, and L. Zhang. A study of priority pricing in multiple service class networks. In *Proc. SIGCOMM '91*, pages 123–129, 1991.

[35] D. R. Cox. Long-range dependence: a review. In H. A. David and H. T. David, editors, *Statistics: An Appraisal*, pages 55–74. Iowa State Univ. Press, 1984.

[36] M. Crovella and A. Bestavros. Self-similarity in world wide web traffic: Evidence and possible causes. In *Proc. ACM SIGMETRICS '96*, pages 160–169, 1996.

[37] M. Crovella, R. Frangioso, and M. Harchol-Balter. Connection scheduling in web servers. In *Proc. USENIX Symposium on Internet Technologies and Systems*, 1999.

[38] M. Crovella and L. Lipsky. Long-lasting transient conditions in simulations with heavy-tailed workloads. In *Proc. 1997 Winter Simulation Conference*, pages 1005–1012, 1997.

[39] C. Cunha, A. Bestavros, and M. Crovella. Characteristics of WWW client-based traces. Technical Report BU-CS-95-010, Computer Science Department, Boston University, 1995.

[40] G. Debreu. A social equilibrium existence theorem. *Proc. Nat. Acad. Sci.*, 38:886–893, 1952.

[41] Robert Devaney. *An Introduction to Chaotic Dynamical Systems*. Addison-Wesley, 1985.

[42] Edsger Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, pages 269–271, 1959.

[43] M. Dresher. Probability of a pure equilibrium point in n-person games. *Journal of Combinatorial Theory*, 8:134–145, 1970.

[44] N. G. Duffield, J. T. Lewis, N. O'Connel, R. Russell, and F. Toomey. Statistical issues raised by the bellcore data. In *Proc. 11th IEE Teletraffic Symposium*, 1994.

[45] Z. Dziong and L. Mason. Fair-efficient call admission control policies for broadband networks—a game theoretic framework. *IEEE/ACM Trans. Networking*, 4(1):123–136, 1996.

[46] P. Erdős and T. Gallai. Graphs with points of prescribed degrees *(hungarian)*. *Mat. Lapok*, 11:264–274, 1960.

[47] P. Erdős and A. Rényi. On random graphs. *Publ. Math. Debrecen*, 6:290–291, 1959.

[48] P. Erdős and A. Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hungar. Acad. Sci.*, 5:17–61, 1960.

[49] A. K. Erlang. Solution of some problems in the theory of probabilities of significance in automatic telephone exchanges. *Post Office Electrical Engineers Journal*, 10:189–197, 1917. Translated from the Danish 1917 article in Elektroteknikeren, Volume 13.

[50] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the Internet topology. In *Proc. ACM SIGCOMM '99*, pages 251–262, 1999.

[51] K. Fan. Fixed point and minimax theorems in locally convex topological linear spaces. *Proc. Nat. Acad. Sci.*, 38:121–126, 1952.

[52] D. Ferguson, C. Nikolaou, and Y. Yemini. An economy for flow control in computer networks. In *Proc. IEEE INFOCOM '89*, pages 110–118, 1989.

[53] D. Ferguson, Y. Yemini, and C. Nikolaou. Microeconomic algorithms for load balancing in distributed computer systems. In *Proc. 8th International Conference on Distributed Computing Systems*, pages 491–499, 1988.

[54] S. Floyd and V. Jacobson. The synchronization of periodic routing messages. In *Proc. ACM SIGCOMM '93*, pages 33–44, 1993.

[55] L. Ford and D. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.

[56] A. Frieze and C. McDiarmid. Algorithmic theory of random graphs. *Random Structures & Algorithms*, 10:5–42, 1997.

[57] J. Gallardo, D. Makrakis, and L. Orozco-Barbosa. Fast simulation of broadband telecommunications networks carrying long-range dependent bursty traffic. *ACM Trans. Modeling and Computer Simulation*, 11:274–293, 2001.

[58] M. Garey and D. Johnson. *Computers and Intractability.* W. H. Freeman and Company, 1979.

[59] M. Gerla and L. Kleinrock. Flow control: a comparative survey. *IEEE Trans. Commun.*, 20(2):35–49, 1980.

[60] I. L. Glicksberg. A further generalization of the Kakutani fixed point theorem with application to Nash equilibrium points. *Proc. Nat. Acad. Sci.*, 38:170–174, 1952.

[61] M. Harchol-Balter and A. Downey. Exploiting process lifetime distributions for dynamic load balancing. In *Proceedings of SIGMETRICS '96*, pages 13–24, 1996.

[62] Garrett Hardin. The tragedy of the commons. *Science*, 162:1243–1248, 1968.

[63] Philip Heidelberger. Fast simulation of rare events in queueing and reliability models. *ACM Trans. Modeling and Computer Simulation*, 5:43–85, 1995.

[64] Herbert Hethcote. The mathematics of infectious diseases. *SIAM Review*, 42(4):599–653, 2000.

[65] Man-Tung T. Hsiao and Aurel A. Lazar. Optimal flow control of multi-class queueing networks with decentralized information. In *Proc. IEEE INFOCOM '87*, pages 652–661, 1987.

[66] Harold Hurst. Long-term storage capacity of reservoirs. *Trans. Am. Soc. Civil Engineers*, 116:770–799, 1951.

[67] P. Husbands, H. Simon, and C. Ding. On the use of singular value decomposition for text retrieval. In *1st SIAM Computational Information Retrieval Workshop*, 2000.

[68] Gordon Irlam. Unix file size survey - 1993. Available at `http://www.base.com/gordoni/ufs93.html`, September 1994.

[69] Van Jacobson. Congestion avoidance and control. In *Proc. ACM SIGCOMM '88*, pages 314–329, 1988.

[70] R. Jain and S. Routhier. Packet trains—measurements and a new model for computer network traffic. *IEEE J. Select. Areas Commun.*, 4(6):986–995, 1986.

[71] H. Jeong, B. Tomber, R. Albert, Z. Oltvai, and A. Babárasi. The large-scale organization of metabolic networks. *Nature*, 407:378–382, 2000.

[72] C. Jin, Q. Chen, and S. Jamin. Inet: Internet Topology Generator. Technical Report CSE-TR-443-00, Department of EECS, University of Michigan, 2000.

[73] K. Kamath, H. Bassali, R. Hosamani, and L. Gao. Policy-aware algorithms for proxy placement in the Internet. In *Proc. SPIE Scalability and Traffic Control in IP Networks*, pages 157–171, 2001.

[74] F. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49:237–252, 1998.

[75] W. Kermack and A. McKendrick. A contribution to the mathematical theory of epidemics. *Proc. Roy. Soc. Lond. A*, 115:700–721, 1927.

[76] Leonard Kleinrock. *Queueing Systems, Volume 1: Theory.* Wiley-Interscience, New York, 1975.

[77] Y. Korilis and A. Lazar. Why is flow control hard: Optimality, fairness, partial and delayed information. In *Proc. 2nd ORSA Telecommunications Conference*, March 1992.

[78] Y. Korilis and A. Lazar. On the existence of equilibria in noncooperative optimal flow control. *Journal of the ACM*, 42(3):584–613, 1995.

[79] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the Web for emerging cyber-communities. *Computer Networks*, 31:1481–1493, 1999. Proc. 8th WWW Conference.

[80] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed Internet routing convergence. In *Proc. of ACM SIGCOMM '00*, pages 175–187, 2000.

[81] W. Leland, M. Taqqu, W. Willinger, and D. Wilson. On the self-similar nature of ethernet traffic. In *Proc. ACM SIGCOMM '93*, pages 183–193, 1993.

[82] Thomas Liggett. *Interacting Particle Systems.* Number 276 in Grundlehren der Mathematischen Wissenschaften. Springer Verlag, New York, 1985.

[83] Z. Liu, P. Nain, D. Towsley, and Z. Zhang. Asymptotic behavior of a multiplexer fed by a long-range dependent process. *Journal of Applied Probability*, 36(1):105–118, 1999.

[84] Linyuan Lu. *Probabilistic Methods in Massive Graphs and Internet Computing.* PhD thesis, University of California, San Diego, 2002.

[85] A. Makowski and M. Parulekar. Buffer asymptotics for $M/G/\infty$ input processes. In K. Park and W. Willinger, editors, *Self-Similar Network Traffic and Performance Evaluation*, chapter 10. Wiley-Interscience, 2000.

[86] B. Mandelbrot and J. Van Ness. Fractional Brownian motions, fractional noises and applications. *SIAM Rev.*, 10:422–437, 1968.

[87] Benoit Mandelbrot. A note on a class of skew distribution function: analysis and critique of a paper by H.A. Simon. *Information and Control*, 2:90–99, 1959.

[88] B. McCorkendale and P. Ször. Code Red buffer overflow. *Virus Bulletin*, pages 4–5, September 2001.

[89] Mercator Internet AS Map. Courtesy of Ramesh Govindan, USC/ISI, 2002.

[90] T. Mikosch. Regular variation, subexponentiality and their applications in probability theory. Technical report, Eurandom, 1999.

[91] Stanley Milgram. The small world problem. *Psychology Today*, 2:60–67, 1967.

[92] M. Molloy and B. Reed. The size of the giant component of a random graph with a given degree sequence. *Combin. Probab. Comput.*, 8:295–305, 1998.

[93] D. Moore, G. Voelker, and S. Savage. Inferring Internet denial-of-service activity. In *Proc. USENIX Security Symposium*, pages 9–22, 2001.

[94] J. F. Nash. Equilibrium points in n-person games. *Proc. Nat. Acad. Sci.*, 36:48–49, 1950.

[95] J. F. Nash. Non-cooperative games. *Annals of Mathematics*, 54:286–295, 1951.

[96] National Laboratory for Applied Network Research. Routing data, 2000. Supported by NFS, http://moat.nlanr.net/Routing/rawdata/.

[97] M. Newman. The structure of scientific collaboration networks. *Proc. Natl. Acad. Sci.*, 4:404–409, 2001.

[98] A. Orda, R. Rom, and N. Shimkin. Competitive routing in multiuser communication networks. *IEEE/ACM Trans. Networking*, 1(5):510–521, 1993.

[99] M. Osborne and A. Rubinstein. *A Course in Game Theory*. The MIT Press, 1994.

[100] S. Östring, H. Sirisena, and I. Hudson. Rate control of elastic connections competing with long-range dependent network traffic. To appear in *IEEE Trans. Commun.*, 2001.

[101] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. In *Proc. 7th WWW Conference*, pages 161–172, 1998.

[102] K. Park, G. Kim, and M. Crovella. On the relationship between file sizes, transport protocols, and self-similar network traffic. In *Proc. IEEE International Conference on Network Protocols*, pages 171–180, 1996.

[103] K. Park and H. Lee. On the effectiveness of route-based packet filtering for distributed DoS attack prevention in power-law internets. In *Proc. ACM SIGCOMM '01*, pages 15–26, 2001.

[104] K. Park, M. Sitharam, and S. Chen. Quality of service provision in noncooperative networks: heterogeneous preferences, multi-dimensional QoS vectors, and burstiness. In *Proc. 1st International Conference on Information and Computation Economies*, pages 111–127, 1998.

[105] K. Park and T. Tuan. Performance evaluation of multiple time scale TCP under self-similar traffic conditions. *ACM Trans. Modeling and Computer Simulation*, 10(2):152–177, 2000.

[106] K. Park and W. Wang. QoS-sensitive transport of real-time MPEG video using adaptive forward error correction. In *Proc. IEEE Multimedia Systems '99*, pages 426–432, 1999.

[107] K. Park and W. Wang. QoS-sensitive transport of real-time MPEG video using adaptive redundancy control. *Computer Communications*, 24:78–92, 2001.

[108] K. Park and W. Willinger. Self-similar network traffic: An overview. In K. Park and W. Willinger, editors, *Self-Similar Network Traffic and Performance Evaluation*. Wiley-Interscience, 2000.

[109] Kihong Park. Warp control: a dynamically stable congestion protocol and its analysis. In *Proc. ACM SIGCOMM '93*, pages 137–147, 1993.

[110] V. Paxson and S. Floyd. Wide-area traffic: the failure of Poisson modeling. In *Proc. ACM SIGCOMM '94*, pages 257–268, 1994.

[111] G. Phillips, S. Shenker, and H. Tangmunarunkit. Scaling of multicast trees: Comments on the Chuang-Sirbu scaling law. In *Proc. ACM SIGCOMM '99*, pages 41–51, 1999.

[112] A. Pikovsky, M. Rosenblum, and J. Kurths. *Synchronization*. Cambridge University Press, 2001.

[113] Michael Rabin. Efficient dispersal of information for security, load balancing, and fault tolerance. *Journal of the ACM*, 36(2):335–348, 1989.

[114] S. Redner. How popular is your paper? *Euro. Phys. J. B*, 4:131–134, 1998.

[115] I. Reed and G. Solomon. Polynomial codes over certain finite fields. *J. SIAM*, (10):300–304, 1960.

[116] H. Ren and K. Park. Toward a theory of differentiated services. In *Proc. IEEE/IFIP IWQoS*, pages 211–220, 2000.

[117] H. Ren and K. Park. On the complexity of optimal aggregate-flow scheduling. Technical Report CSD-TR-02-021, Department of Computer Sciences, Purdue University, 2002.

[118] H. Ren and K. Park. Performance evaluation of optimal aggregate-flow scheduling: a simulation study. To appear in *Computer Communications*, 2002.

[119] Huan Ren. *Aggregate-Flow Scheduling: Theory and Practice*. PhD thesis, Purdue University, 2002.

[120] RIPE. Routing Information Service Raw Data, 2002. http://data.ris.ripe.net.

[121] J. B. Rosen. Existence and uniqueness of equilibrium points for concave n-person games. *Econometrica*, 33(3):520–534, 1965.

[122] B. Rupp, R. Edell, H. Chand, and P. Varaiya. INDEX: A platform for determining how people value the quality of their Internet access. In *Proc. IEEE/IFIP IWQoS*, pages 85–90, 1998.

[123] J. Sairamesh, D. Ferguson, and Y. Yemini. An approach to pricing, optimal allocation and quality of service provisioning in high-speed networks. In *Proc. IEEE INFOCOM '95*, pages 1111–1119, 1995.

[124] A. Shaikh, J. Rexford, and K. Shin. Load-sensitive routing of long-lived IP flows. In *Proc. ACM SIGCOMM '99*, 1999.

[125] C. Shannon and W. Weaver. *The mathematical Theory of Communication*. The University of Illinois Press: Urbana, 1949.

[126] H. A. Simon. On a class of skew distribution functions. *Biometrica*, 42:425–440, 1955.

[127] Eugene Spafford. The Internet worm: crisis and aftermath. *Communications of the ACM*, 32(6):678–687, 1989.

[128] W. Szpankowski, P. Jacquet, and L. Georgiadis. Personal communication, 2001.

[129] H. Tangmunarunkit, J. Doyle, R. Govindan, S. Jamin, W. Willinger, and S. Shenker. Does AS size determine AS degree? *Computer Communication Review*, 31(5), 2001.

[130] H. Tangmunarunkit, R. Govindan, and S. Shenker. Internet path inflation due to policy routing. In *Proc. SPIE Scalability and Traffic Control in IP Networks*, pages 188–195, 2001.

[131] M. Taqqu, W. Willinger, and R. Sherman. Proof of a fundamental result in self-similar traffic modeling. *Computer Communication Review*, 26:5–23, 1997.

[132] T. Tuan and K. Park. Multiple time scale congestion control for self-similar network traffic. *Performance Evaluation*, 36:359–386, 1999.

[133] T. Tuan and K. Park. Multiple time scale redundancy control for QoS-sensitive transport of real-time traffic. In *Proc. IEEE INFOCOM '00*, pages 1683–1692, 2000.

[134] University of Michigan. AS Graph Data Sets, 2002. http://topology.eecs.umich.edu/data.html.

[135] University of Oregon. Oregon Route Views, 2000. http://www.routeviews.org/ and http://archive.routeviews.org/.

[136] A. Veres and M. Boda. The chaotic nature of TCP congestion control. In *Proc. IEEE INFOCOM '00*, pages 1715–1723, 2000.

[137] A. Veres, Z. Venesi, S. Molnar, and G. Vattay. On the propagation of long-range dependence in the internet. In *Proc. ACM SIGCOMM '00*, pages 243–254, 2000.

[138] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

[139] C. Waldspurger, T. Hogg, B. Huberman, J. Kephart, and W. Stornetta. Spawn: a distributed computational economy. *IEEE Trans. Software Engineering*, 18(2):103–117, 1992.

[140] D. Watts and S. Strogats. Collective dynamics of 'small world' networks. *Nature*, 393:440–442, 1998.

[141] Jörgen Weibull. *Evolutionary Game Theory*. The MIT Press, 1997.

[142] W. Willinger, M. Taqqu, R. Sherman, and D. Wilson. Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level. In *Proc. ACM SIGCOMM '95*, pages 100–113, 1995.

[143] J. Winick and S. Jamin. Inet-3.0: Internet Topology Generator. Technical Report CSE-TR-456-02, Department of EECS, University of Michigan, 2002.

[144] G. Wu, E. Chong, and R. Givan. Congestion control via online sampling. In *Proc. IEEE INFOCOM '01*, 2001.

[145] L. Zhang and D. Clark. Oscillation behavior of network traffic: a case study simulation. *Internetworking: Research and Experience*, 1(2):101–112, 1990.

[146] G. Zipf. *Human Behavior and the Principle of Least Effort.* New York, Hafner, 1949.