Proposal to NSF CISE

Information Technology Research (NSF 02-168)

FAST Protocols:

Theory, Implementation, Experiment, Deployment

Steven H. Low (PI)
CS/EE Caltech

Guy Almes Internet2	Werner Almesberger CS Caltech	Julian Bunn CACR Caltech
Les Cottrell SLAC Stanford	John C. Doyle CDS/EE/BE Caltech	Wu-chun Feng LANL
Cheng Jin CS Caltech	Harvey Newman Physics Caltech	Fernando Paganini EE UCLA
Stanislav Shalunov Internet2	Linda Winkler ANL	Steven Yip Cisco

February 12, 2003

Contents

1 Executive summary

Our **goal**, focused, ambitious yet realistic, is to develop theories and algorithms for the future ultrascale networks, implement and demonstrate them in state-of-the-art testbeds, and deploy them in communities that have a clear and urgent need today. Specifically, we will

- develop theories to understand issues in large scale networks, including performance and stability, interaction among different protocols as well as across evolutionary generations of the same protocol, interaction of congestion control and routing, noisy feedback and randomness inherent in these networks;
- design robust, stable and scalable protocols that can achieve reliable high performance in production networks, based on a strong foundation of theory and simulation, as well as a systematic program of experiments on national and international testbeds;
- progressively deploy the protocols to be developed in this project across production networks, starting with High Energy and Nuclear Physics (HENP) networks and Abilene, as well as advanced facilities including the NSF-funded TeraGrid;
- work with standards bodies, such as IETF, GGF (Global Grid Forum), and Grid projects around the world to deploy them in toolkits and production networks in communities that urgently need ultrascale networking.

The **motivation** is to meet the needs of next generation research projects in many scientific fields, including physics, biology, earth and atmospheric sciences, and many others which are data intensive and require global-scale Grids. These projects face unprecedented challenges, having to produce, store, and transfer hundreds of Terabytes of data per experiment around the world [53, 52]. Rapid advances in computing, communication and storage technologies will provide the required raw capacities. The key challenge we face, and intend to overcome, is that some of our current network control and resource sharing algorithms cannot scale to this regime.

The **integrated approach**, where theory development, implementation and experiments inform and influence each other intimately, is what makes our project unique and fundamental progress possible, and what gives us a real chance of significant impact and deployment. *An ITR medium project will make such an approach possible which disparate small projects cannot.*

This approach has already been tested in a pilot project that addresses the performance and stability issues of TCP in multi-Gbps networks. The preliminary FAST (Fast AQM Scalable TCP) Linux kernel developed in this project has achieved 925Mbps and 95% utilization with a single TCP flow, stably over an extended period (an hour) on an intercontinental path with a 180ms delay round trip, using 1460-byte packet size (user data).

Through the pilot project, we have already established, and will expand, strong working relations with strategic user and theory groups; see accompanying Letters of Support.

Broader impacts: HENP and its worldwide collaborations could be a model for new modes of information sharing and communication in society at large. Fast protocols to be developed in this project, being one of the enabling technologies, thus have a broader impact that extends beyond the bounds of scientific and engineering research. The project will provide a unique training to graduate and undergraduate students, from theory to experiment. Research results will be incorporated into into an advanced networking course at Caltech, and a new interdisciplinary course being co-developed by the PI and colleagues, aimed at bringing together faculty and students to work on problems at the boundaries of control, communication, and computing, to be offered through the Departments of Electrical Engineering, Computer Science, and Control and Dynamical Systems at Caltech. We will pursue outreach activities that include workshops to bring together collaborators, colleagues, and students around the world and technology leaders in industry for focused study in selected topics.

2 Motivation

There is a clear and urgent need for multi-Gigabit networks in the scientific community, today, and a need for ultrascale networks in the near future that provide more than 100 Gbps of sustained throughput end-to-end to transfer Petabyte files. In this section, we provide more details on the need for ultrascale networking and argue that protocols, not hardware capacity, will become the bottleneck in the future.

2.1 Demand for ultrascale networking

The HENP (High Energy and Nuclear Physics) community has a long tradition of pushing computing and networking technologies to their limits, in production environments. This trend has accelerated in the last few years both due to the Petabytes (10^{15} bytes) of data acquired, stored, distributed and processed by the worldwide HENP collaborations, and due to the development of Data Grids, which aim to make the data available rapidly, transparently, and *dynamically* to scientists around the globe. Experiments now underway at SLAC, Fermilab and Brookhaven are already accumulating Petabyte datasets. The next generation of particle physics experiments now under development, due to begin operation in 2007 at CERN in Geneva, will deal with data volumes of tens of Petabytes (in 2007–2008) to Exabytes (10^{18} bytes) in the decade following. This will impose tremendous new demands on computing, communication and storage technologies.

A current example illustrating the data and computationally intensive character of HENP problems encountered by research teams is the search for Higgs particles at the LHC (the Large Hadron Collider at CERN). A full optimization of the separation of the Higgs discovery signal from potentially overwhelming backgrounds is estimated to require 10^8 fully simulated and reconstructed background events, drawn from 10^{11} generated events (sets of simulated four vectors) using loose pre-selection criteria. The processing requirement is approximately 10^6 CPU-days, or 10,000 of today's fastest processors used round the clock for three to four months. The data resulting from this study will be on the order of 200–400 Terabytes. This implies a need to transfer 2–4 Terabytes per day produced in bursts, which will take 0.5–1 hour of transfer time per day at a throughput of 1 Gbyte/sec end-to-end over the wide area.

The largest projects in HENP involves more than 2000 physicists in more than 30 countries. Collaborations on this global scale would not have been attempted if the physicists could not plan on excellent networks and fast protocols: to interconnect the physics groups throughout the lifecycle of the experiment and, and to make possible the construction of Data Grid systems capable of handling, distributing and sharing the data and analysis among physicists around the world. Protocols to be developed in this project will enable the next generation data sharing and analysis systems, where Terabyte samples are readily and spontaneously available, accessed and transported in minutes, on the fly, rather than hours or days as is the current practice. This will enable new ways to do science and help drive future scientific discovery. See attached Letters of Support for more applications of ultrascale networking.

2.2 TCP/IP paradigm

The ability to scale silicon technology improves the performance of the devices and decreases their cost, both at an exponential rate. Modeling studies and extrapolations of the rapid advances in computing, communication, and storage technologies show that sufficient capacity will be available for the new generation of scientific computing. The key challenge we face, and intend to overcome, is that some of our current network control and resource sharing algorithms cannot scale to this regime. This has led to serious doubts about whether the current TCP/IP paradigm of statistical multiplexing and end-to-end control is suitable for future ultrascale networks. We believe it is, with *proper* protocols.

Statistical multiplexing is ideal for applications that generate bursty traffic, or that have an elastic bandwidth requirement. It is however difficult to characterize the resource requirements of such applications, and hence connection admission control is rarely implemented in packet networks.[1] Since the number of connections in the network is not controlled, their source rates must be dynamically regulated to avoid overwhelming the network or the receivers. This is the purpose of end-to-end flow control.

It is possible to stream a Terabyte file at a large fixed rate over a reserved "circuit", without the need for end-to-end flow control. This circuit-switching approach, however, is inefficient (except in specialized applications) because, unlike uncoded voice, there is not a natural rate to reserve for bulk transfers. Indeed, bulk transfers are inherently elastic, in that they can take full advantage of increased bandwidth or reduce rate to accommodate other traffic. It is likely that the available bandwidth will fluctuate during the (long) lifetime of a bulk transfer, due to arrivals and departures of other transfers. It is therefore much more efficient to allow the transfers to share bandwidth dynamically, in future networks as they do in current ones. Statistical multiplexing, together with end-to-end flow control, is a key factor that has enabled an explosive set of applications to share the Internet efficiently.

Traffic generated by scientific applications is ideal for end-to-end control because of its extreme heavy-tailed nature [37, 56, 70, 12, 72]. An important implication of such traffic is that, even though most files are small ("mice"), most packets belong to huge files ("elephants") and hence can be effectively controlled end to end. The heavier the tail, the better end-to-end control works, because the duration of typical elephant connections will be large compared with the convergence time of the control mechanism.

2.3 Protocol scalability problems

A breakthrough that has allowed the Internet to expand by five orders of magnitude in size and in backbone speed in the last 15 years was the invention in 1988 by Jacobson of an end-to-end congestion control algorithm in TCP (Transmission Control Protocol) [25].

This algorithm, designed when most parts of the Internet could barely carry the traffic of a single uncompressed voice call, however, cannot scale to the future ultrascale networks that must be able to carry the traffic of 1.5 million concurrent voice calls. This is due to serious equilibrium and stability problems in high capacity long distance networks. For instance, the current TCP protocol requires an extremely small loss probability to support the window size of ultrascale networking. To achieve a throughput of just 10 Gbps over a distance with 180 ms round-trip delay (e.g., between CERN in Geneva and Stanford Linear Accelerator Center (SLAC) in CA) will require a *packet* loss probability on the order of 10^{-10}. This can be difficult to achieve end to end. Even if this loss probability is achieved, it represents an extremely noisy feedback signal (rare event) for the sources to reliably use for control. Moreover, since TCP must induce loss in order to estimate the available bandwidth, however rare losses are, when they *inevitably* occur, it takes more than 3 hours to recover to full utilization. Finally, AIMD (Additive Increase Multiplicative Decrease) induces instability at high speed, making wild oscillations unavoidable [22, 43].

These problems prevent TCP from making effective use of available bandwidth as network capacity grows; see Figure 1 and Table 1 below. As 1-10 Gbps Ethernet, OC192 and faster network backbones, and improved operating systems and default settings are becoming the norm, *one of the main impediments to ultrascale networking will soon be the lack of scalability of some of our current control protocols.*

3 Pilot project: performance and stability

We now briefly review preliminary results obtained in a pilot project to drastically improve the performance and stability of TCP in ultrascale regime. It illustrates the scalability problem of current TCP and our integrated approach, from theory to experiment.

[1] We note however that it is possible to implement distributed admission control in packet networks [31, 21].

3.1 Network model

Exciting advances have been made in the last couple years on understanding the equilibrium and dynamic behavior of large networks, such as the Internet. There is now a preliminary theory both to analyze the scalability problems of existing protocols, and to guide the design of new protocols that can in principle scale to arbitrary capacity, delay and topology. We now give a brief overview of this theory, *focusing on our own contribution. See, e.g., [59, 19, 42, 45] and references therein for an extensive bibliography.*

A congestion control algorithm consists of two components, a source algorithm, implemented in TCP, that adapts sending rate (or window) to congestion information in its path, and a link algorithm, implemented in routers, that updates and feeds back a measure of congestion to sources that traverse the link. Typically, the link algorithm is implicit and the measure of congestion is either packet loss probability or queueing delay. For example, the current protocol TCP Reno and its variants use loss probability as a congestion measure, and TCP Vegas [5] uses queueing delay as a congestion measure [50, 44]. Both are implicitly updated by the queueing process and implicitly fed back to sources via end-to-end loss and delay, respectively. The source-link algorithm pair, referred to here as TCP/AQM (active queue management) algorithms[2], forms a distributed feedback system. The equilibrium and dynamic properties of this system determine the network performance, such as throughput, utilization, delay, loss, fairness, response to congestion, and robustness to uncertainties.

Specifically, a network is modeled as a set of L links with finite capacities $c = (c_l, l \in L)$. They are shared by a set of N sources. Each source i transmits at rate $x_i(t)$. Let R denotes the $L \times N$ routing matrix, $R_{li} = 1$ if i uses link $l \in L$, and 0 otherwise. These transmission rates determine the aggregate flow $\hat{x}_l(t) := \sum_i R_{li} x_i(t - \tau_{li}^f)$ at each link, where τ_{li}^f denote the forward transmission delays from sources to links. Each link l maintains a *congestion measure* $p_l(t)$, called "price", that has different interpretations in different protocols (e.g. loss probability in TCP Reno, queueing delay in TCP Vegas). A sources i has access only to the *aggregate* price $\hat{p}_i(t) := \sum_l R_{li} p_l(t - \tau_{li}^b)$ in its route, where τ_{li}^b denote the backward delays in the feedback path. Decentralization requires that source rates $x_i(t)$ be adjusted based only on aggregate prices $\hat{p}_i(t)$, and prices $p_l(t)$ be adjusted based only on aggregate rates $\hat{x}_l(t)$. This can be represented as

$$\dot{x}_i \; = \; F_i(x_i(t), \hat{p}_i(t)) \qquad \text{and} \qquad \dot{p}_l \; = \; G_l(p_l(t), \hat{x}_l(t)) \tag{1}$$

Here, different TCP protocols are modeled as different F_i and different AQM's are modeled as different G_l.

The model is very general: *any* network under end-to-end control fits in this framework. Even though the theory is discussed in the context of TCP congestion control, it applies to any end-to-end scheme that works within the decentralization constraints inherent in a large network. It therefore applies not only to TCP, but also to any end-to-end flow control algorithm, e.g., those implemented on top of UDP.

3.2 Theory

The equilibrium properties of the network can be readily understood by interpreting TCP/AQM as a distributed algorithm over the Internet to maximize aggregate source utility, and a source's utility function is (often implicitly) defined by its TCP algorithm, see e.g. [30, 41, 47, 49, 50, 33, 36, 44, 40] for unicast and [28, 15] for multicast.

The key idea in the duality model [41, 44, 40] is to interpret source rates $x(t)$ as primal variables, prices $p(t)$ as dual variables, and congestion control (1) as a distributed primal-dual algorithm over the Internet to solving the problem of maximizing aggregate utility subject to capacity constraints:

$$\max_{x \geq 0} \quad \sum_i U_i(x_i), \qquad \text{subject to} \quad Rx \; \leq \; c \tag{2}$$

and its dual:

$$\min_{p \geq 0} \quad \sum_i \max_{x_i} \left(\; U_i(x_i) - x_i q_i \right) \; + \; \sum_l p_l c_l \tag{3}$$

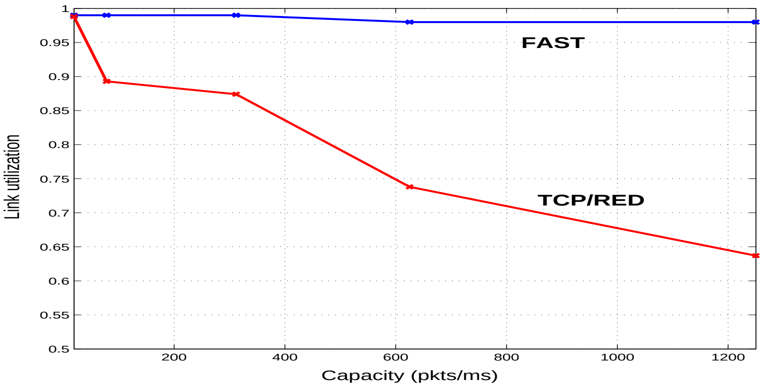
[2] We will henceforth refer it as a "TCP algorithm" even though we really mean the congestion control algorithm in TCP.

i.e., the equilibrium point of (1) are optimal solutions of (2)–(3) with appropriate utility functions. Different TCP/AQM protocols all solve the same prototypical constrained nonlinear program, but they use different utility functions and implement different iterative rules (F_i, G_l) to optimize them.

The duality model provides a natural framework to understand the equilibrium properties of the current protocols. It allows us to predict the equilibrium source rates, link loss probabilities and queue lengths in a *multi-link* multi-source heterogeneous network for the various TCP/AQM protocols [44, 40]. Moreover, since the underlying optimization problem is a concave program, these equilibrium properties can be efficiently computed numerically, even for large scale networks that are hard to simulate.

Is the equilibrium of the distributed feedback system of TCP/AQM stable? It is shown in [22, 43] that the current algorithms can become unstable as delay increases, or more strikingly, as network capacity increases! This is one of the main difficulties in operating in fast long-distance networks. Stability is important for two reasons. First, if the performance (throughput, loss, delay) and fairness, that are determined by the equilibrium, are desirable, then we want the equilibrium to be stable so that the network is either in equilibrium or in pursuit of the (desirable) equilibrium. Second, we currently do not have a theory to predict the network behavior when it loses stability. It is hence risky to operate a large network in an unstable regime, and unnecessary if, as we do now, know how to operate it in a stable regime without sacrificing performance.

The lack of scalability of TCP due to both equilibrium and stability problems is illustrated in Figure 1 with packet-level simulations using `ns-2` (Network Simulator). The figure shows the link utilization of TCP/RED and that of FAST algorithm we developed. As link capacity increases, the utilization under TCP/RED drops steadily, in stark contrast to that under FAST.



(a) Link utilization

Figure 1: Link utilization of TCP/RED and FAST at bandwidth from 155Mbps to 10Gbps (packet size = 1KB).

If we can rebuild both TCP (source) algorithm and AQM (link) algorithm from scratch, then we now know how to design TCP/AQM algorithm pairs, that are as simple and decentralized as the current protocol, but that maintain linear stability in networks of *arbitrary* capacity, size, delay and load [54]. (See also [64, 65, 33, 35, 15, 36, 29, 27].) The main insight from this work is that, to maintain stability in high capacity large distance networks, sources should scale down their responses by their individual round trip delays and links should scale down their responses by their individual capacity. This insight combined with that from [44] leads to a TCP algorithm that can maintain linear stability without having to change the current link algorithm [8, 55]. Moreover, it suggests an incremental deployment strategy where performance steadily improves as ECN deployment proliferates [8].

This implies that by modifying just the TCP kernel at the *sending hosts*, we can stabilize the Internet with the current Droptail routers. It motivates the implementation of the FAST (Fast AQM Scalable TCP) stack in Linux kernel. The FAST kernel builds on insights in Reno, NewReno, SACK, and Vegas, and uses congestion information embedded in both loss and delay to adjust its window. It reacts rapidly yet stably to achieve high performance.

3.3 Experiment

The FAST kernel was demonstrated publicly for the first time at the SuperComputing Conference (SC2002) in Baltimore, MD, in November 2002 by a Caltech-SLAC research team working in partnership with CERN, DataTAG, StarLight, Cisco, and Level(3). The demonstration used a 10 Gbps link donated by Level(3) between Starlight (Chicago) and Sunnyvale, as well as the DataTAG 2.5 Gbps link between Starlight and CERN (Geneva), and the Abilene backbone of Internet2. The network routers and switches at Starlight and CERN were used together with a GSR 12406 router loaned by Cisco at Sunnyvale, additional Cisco modules loaned at Starlight, and sets of dual Pentium 4 servers each with dual Gigabit Ethernet connections at Starlight, Sunnyvale, CERN and the SC2002 show floor provided by Caltech, SLAC and CERN. The network setup is shown in Figure 2.

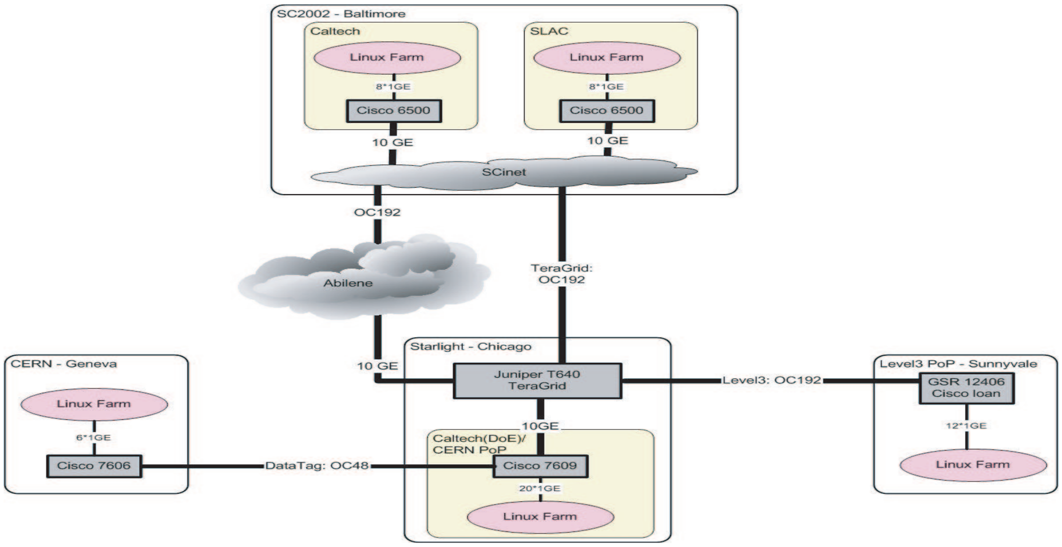


Figure 2: Network setup in SC2002

We have conducted a number of experiments, all using standard MTU (Maximum Transmission Unit), 1500 bytes including TCP and IP headers. In particular, we have demonstrated 925 Mbps (95% utilization) stably with a single TCP flow between CERN in Geneva and Level(3)'s PoP (point of presence) in

#flow	throughput Mbps	utilization	delay ms	distance km	duration s	transfer GB
1	925 (266)	95% (27%)	180	10,037	3,600	387 (111)
2	1,797 (931)	92% (48%)	180	10,037	3,600	753 (390)
7	6,123	90%	85	3,948	21,600	15,396
9	7,940	90%	85	3,948	4,030	3,725
10	8,609	88%	85	3,948	21,600	21,647

Table 1: Experimental results: average statistics. Statistics in parentheses are for current TCP implementation in Linux v2.4.18.

Sunnyvale, over a distance of over 6,000 miles on a single Gigabit Ethernet port at each end of the path, almost 4 times the performance of the current protocol. The peak window size was 14,255 packets. The details of five of these experiments are shown in Table 1. All statistics are *averages* over the duration of the experiments. For comparison, the corresponding statistics for the current TCP implementation in Linux 2.4.18 kernel, using SACK, are shown in parentheses in Table 1, with optimized parameter `txqueuelen` = 10,000 packets, for 1 and 2 flows (`txqueuelen` = 100 packets for FAST). The throughput traces of these experiments are shown in Figure 3.

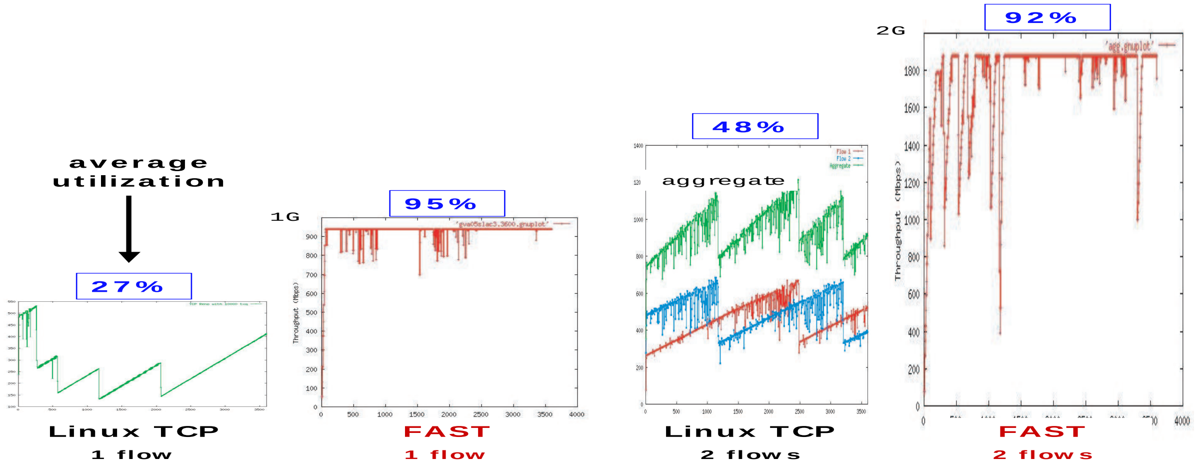


Figure 3: Throughput traces for 1 and 2-flow experiments (first two row of Table 1): x-axis is time, y-axis is aggregate throughput, and percentage is utilization.

4 Goal and issues

The preliminary success of the pilot project both raises new issues that must be addressed to support a large scale deployment of FAST kernel, and confirms our integrated approach to tackle these issues. We propose to re-reexamine, and redesign when necessary, control and resource management protocols for ultrascale networking.

4.1 Goal and issues identified

Our **goal** is to develop and deploy scalable and robust protocols, based on sound theoretical foundations, in communities that need them, today. Our **approach** is to

1. develop theories and algorithms,
2. simulate and implement them,
3. test and demonstrate them in state-of-the-art testbeds such as Abilene, TeraGrid and HENP networks, and
4. work with standards bodies, such as IETF, GGF (Global Grid Forum), and Grid projects around the world to deploy them in toolkits and production networks in communities that urgently need ultrascale networking.

It is the *combination* of theory, implementation and experiments that makes our project unique and fundamental progress possible, and that gives us a real chance of significant impact and deployment. *An ITR medium project makes such an integrated approach possible that disparate small projects cannot.*

The pilot project has made exciting progress on performance and stability issues of TCP, but has also brought more questions into sharp focus:

1. Performance and stability:
   The current theory mostly addresses *linear* stability. How do we understand nonlinear stability with feedback delay and network behavior in unstable regime?

2. Interaction with current TCP:
   What are the fairness properties of FAST and how does it interact with the current TCP when they co-exist in the same network?

3. Interaction with IP routing:
   How does routing affect the equilibrium and stability of general TCP/AQM algorithms at a fast timescale (including FAST), and conversely, how does TCP/AQM algorithms affect routing stability at a slow timescale?

4. Noisy information:
   Without ECN (Explicit Congestion Notification), sources can only observe packet loss and queueing delay end to end. What are the tradeoffs in using packet loss versus queueing delay as a congestion measure, how do distortions and quantization of congestion information affect the equilibrium and stability of the network, and how best to use both sets of information?

5. Stochastic effects:
   How to extend the current duality framework to incorporate short-duration TCP and UDP sources, and randomness inherent in a large-scale network? How do they affect network equilibrium and stability? How to design FAST algorithms that are robust to, or can exploit, randomness?

These are fundamental questions that we have identified in the pilot project. They have strong implications on protocol design and implementation. We have also identified several implementation issues, some of which we will describe in the next section. No doubt new theoretical, implementation, and experimental challenges will arise as we pursue our goal.

In the rest of this section, we elaborate on each of these unresolved issues. We will discuss our approach to address them in the next section.

4.2 Performance and stability

By "performance", we mean throughputs for each source, equilibrium queueing delays and loss probabilities at each link and end-to-end at each source, and link utilization at each link. These properties are determined by the equilibrium point of the dynamical system described by (1), and the equilibrium point is the solution of the utility maximization problem (2) and its dual (3). How do we *design* appropriate F_i and G_l to achieve a desirable equilibrium? Given a fixed TCP F_i, what is the "best" AQM G_l, and conversely, given G_l, what is the "best" F_i?

We now have a linear stability theory to understand the behavior of a large-scale network around the equilibrium in the presence of feedback delay, e.g., [54, 64, 22, 65, 35, 34, 43, 55, 8]; see also [59, 19, 42, 45] for more extensive bibliography. However, very little is known about global stability *in the presence of delay* [41, 68, 16]. Can we extend and apply nonlinear dynamical systems and bifurcation theories to understand global stability of large scale networks in the presence of delay, and more importantly, their behavior outside stability regions [63, 57]? Our simulation experience suggests that some TCP algorithms F_i (e.g., Reno) not only has a smaller stability region than other algorithms (e.g., Vegas), but also, the instability and performance degradation are more severe when it loses stability. We will study this phenomenon more precisely and develop models to quantify or bound instability and its effect on performance. If we succeed, then perhaps we can safely operate a network in unstable regimes, under appropriate TCP/AQM algorithms (F_i, G_l).

Research on these issue, both theory and implementation, will be funded by ongoing NSF grants.

4.3 Interaction with current TCP

Fairness is a property about the equilibrium rate vector x of the system (1). What is the fairness we should strive for, and how can it be achieved in a large-scale distributed network in a decentralized and adaptive manner? We illustrate these questions with a concrete example.

The fairness between FAST and TCP Reno sources (or its variants such as NewReno or SACK) when they share the same network is a difficult issue because they use different congestion measures (loss probability for Reno and queueing delay for FAST, as for Vegas). The simple duality model described in Section 3 must be extended to a game-theoretic or economic model to study their interaction. Suppose there are m FAST sources, with equilibrium rates $x_i, i = 1, \ldots, m$, and n Reno sources, with equilibrium rates $y_i, i = 1, \ldots, n$. Let the protocol parameters of FAST sources be $\alpha := (\alpha_i, i = 1, \ldots, m)$. The equilibrium rates $x = x(\alpha) := (x_i, i = 1, \ldots, m)$ and $y = y(\alpha) := (y_i, i = 1, \ldots, n)$ of FAST and Reno sources, respectively, depend on the protocol parameter α. Let α take value in a convex set A. Let $\overline{x}(\alpha)$ be the unique Vegas rates if there were no Reno sources ($n = 0$), and let $\overline{y}$ be the unique Reno rates if there were no Vegas sources ($m = 0$). Let $\underline{x}(\alpha)$ be the unique Vegas rates if network capacity is $c - R_y y$ where R_y is the routing matrix for Reno sources. Let

$$X^* \quad := \quad \text{co}\{\underline{x}(\alpha),\ \overline{x}(\alpha), \alpha \in A\}$$

where co$S$ is the convex hull of a set S. X^* includes all possible Vegas rates if Vegas were given strict priority over Reno or if Reno were given strict priority over Vegas, and all rates in between. We have the following conjecture:

Conjecture 1 *Under mild conditions, given any target $x^* \in X^*$, there exists a unique $\alpha^* \in A$ such that $x(\alpha^*) = x^*$.*

The conjecture implies that given any fairness criterion, in terms of a desirable rate allocation x^*, there exists a unique protocol parameter α^* that achieves it. It also implies, however, that improper choice of α can squeeze out Reno sources or FAST sources, leading to extreme unfairness. This conjecture can be shown to hold in the case of a single bottleneck link. Moreover, in that case, the unique α^* can be computed from network parameters.

Does the conjecture hold in a general network? This would imply the existence of a unique α^* that achieves any target fairness. If so, can we compute the α^* *given* global information? Can the FAST sources i individually compute the desired α_i^* in a decentralized manner using only local information, and iteratively, adapting to changing network conditions? What is the stability of this iterative procedure and how will it interact with congestion control that operates at a faster timescale?

4.4 Interaction with routing

IP chooses shortest-path within Autonomous Systems. If path weights depend on the load or performance metrics at the links, such as queueing delay or loss probability, then routing adapts to network congestion and can potentially better balance traffic in the network. How, then, can we understand the interaction of TCP/AQM and IP routing, when both adapt, at different timescales, to alleviate congestion?

The duality model described in Section 3 assumes that the routing matrix R is fixed at the timescale of interest, and interpret TCP/AQM as maximizing the aggregate utility over source rates. Consider the problem of maximizing utility over *both* routes and rates:

$$\max_{R \in \mathcal{R}} \max_{x \geq 0} \quad \sum_i U_i(x_i) \qquad \text{subject to} \;\; Rx \;\leq\; c$$

where $\mathcal{R}$ is a given finite set of available paths between all source-destination pairs. Then the associated dual problem is

$$\min_{p\geq 0} \max_{R\in\mathcal{R}, x\geq 0} L(R,x,p) \quad = \quad \min_{p\geq 0} \sum_i \max_{x_i\geq 0} \left(U(x_i) - x_i \min_{R_i\in\mathcal{R}_i} \sum_l R_{li} p_l \right) + \sum_l p_l c_l \tag{4}$$

where $\mathcal{R}_i$ denotes the set of available routes for source-destination pair i and R_i (column of routing matrix R) is an element of $\mathcal{R}_i$. The striking feature of the dual problem is that *the maximization over R takes the form of shortest-path routing with prices p as link costs.* This raises the tantalizing possibility that TCP–AQM/IP might turn out to be a distributed primal-dual algorithm that maximizes utility over both rates and routes, with proper choice of link costs.

We show in [66], however, that the primal problem is NP-hard and hence in general cannot be solved by shortest-path routing. This prompts a set of fundamental questions. Viewed as an approximation algorithm, how well does shortest-path routing, together with TCP/AQM, solve the utility maximization problem? What is the competitive ratio of this distributed approximation algorithm? This *interpretation* of TCP/IP as a distributed approximation algorithm seems novel and may have interesting theoretical and practical ramifications.

How does utility maximization interact with routing stability? Specifically, suppose routing changes at a slower timescale than TCP/AQM, so that in each discrete period k with routing $R(k)$, TCP/AQM converges instantly and source rates $x(k) = x(R(k))$ and prices $p(k) = p(R(k))$ are the primal and dual solutions of (2)–(3) with fixed routing $R = R(k)$. Suppose in each period k, routing $R(k)$ is determined by shortest-path algorithm with link cost $d_l(k)$ that has both a static and a dynamic component, $d_l(k) = \beta\tau_l + \alpha p_l(k)$. Here τ_l are the fixed latencies and $p_l(k) = p_l(R(k))$ are the dual-optimal prices on links l in period k. The protocol parameters α and β determine the responsiveness of routing to network traffic: $\alpha = 0$ corresponds to static routing, $\beta = 0$ corresponds to purely dynamic routing, and the larger the ratio of α/β, the more responsive routing is to network traffic. Under what condition on α, β is routing $R(k)$ stable? When it is stable, what is the maximum utility in equilibrium?

For a special ring network, it can be proved that there is an inevitable tradeoff between maximum achievable utility and routing stability. In this case, shortest-path routing based purely on congestion prices is unstable. Adding a sufficiently large static component (large α) to link cost stabilizes it, but the maximum utility achievable by shortest-path routing decreases with the weight α on the static component. We conjecture that these conclusions hold in general networks.

Conjecture 2 *Given any network represented as a graph, let R_α be the equilibrium routing and $V(\alpha)$ be the maximum utility in equilibrium, i.e., solution of (2) with R_α as the routing matrix. Then*

1. *$\alpha_1 < \alpha_2$ implies $V(\alpha_1) \leq V(\alpha_2)$.*
2. *Routing is stable ($R(k) \to R_\alpha$ as $k \to \infty$) if α is sufficiently small and unstable if α is sufficiently large.*

4.5 Noisy information

Without ECN, a source can only observe either loss or delay end to end. What is the effect of the accuracy of feedback information on equilibrium and on stability?

The current protocol uses loss probability as a congestion measure. Like Vegas, FAST uses queueing delay as a congestion measure [44, 8, 55]. Queueing delay has two important advantages over loss probability. First, each measurement of loss (a packet is ACK'ed or or not) provides one bit of congestion information whereas each measurement of queueing delay contains multiple bits of information. Hence queueing delay conveys congestion at a finer granularity. Second, as shown in [44, 54, 8, 55], the dynamics of delay as a congestion measure has the right scaling with respect to capacity, which enhances its scalability to ultrascale networks. Loss probability does not have this scaling.

Suppose, instead of (3), link l (implicitly) generates loss probability $p_l(t)$ and queueing delay $q_l(t)$ based on local information $\hat{x}_l(t)$, the aggregate input rate at link l:

$$\dot{p}_l(t) \;=\; G_l(p_l(t), q_l(t), \hat{x}_l(t)) \qquad \text{and} \qquad \dot{q}_l(t) \;=\; H_l(p_l(t), q_l(t), \hat{x}_l(t))$$

Suppose TCP algorithm can react to both loss $p_l(t)$ and delay $q_l(t)$, but only their aggregate end-to-end values and possibly with distortions, quantizations, and randomness. Then, let $\hat{p}_i(t) = \sum_l R_{li}\delta_l(p_l(t))$ be the end-to-end loss probability observed at source i, (approximately) consisting of sum of distorted probabilities $\delta_k(p_l(t))$ in its path. Let $\hat{q}_i(t) = \sum_l R_{li}\epsilon_l(q_l(t))$ be the end-to-end distorted queueing delay observed at source i. Then a general TCP algorithm can be represented by (compare with (1))

$$\dot{x}_i \;=\; F_i(x_i(t), \hat{p}_i(t), \hat{q}_i(t))$$

For instance, current TCP reacts only to $\hat{p}_i(t)$ and FAST reacts only to $\hat{q}_i(t)$. What is the tradeoff? How best to use both $\hat{p}_i(t)$ and $\hat{q}_i(t))$ in adapting $x_i(t)$? What are the effects of distortions, quantization, and randomness, represented by $\delta(p_l(t))$ and $\epsilon(q_l(t))$, on the equilibrium (utility maximization) and stability of the network?

A promising direction of attack is to apply ideas from rate distortion theory and optimization theory [11, 20, 4, 46] to study the effect of quantization on the the maximum achievable utility in problem (2)–(3), and to apply recent research on control under quantization or bandwidth constraints [17, 6, 24, 23] to study its effect on linear stability.

4.6 Stochastic effects

The model (1) ignores all randomness inherent in a large network. This can be justified by the results of [2, 58, 14, 62, 3], which implies that, asymptotically as the number of sources increases, the behavior of the stochastic system can be described by a deterministic ordinary differential equation as in (1). What is the effect of randomness on utility maximization and on network stability? What are the noise rejection and robustness properties of different TCP/AQM algorithms (F_i, G_l), including FAST?

The duality model described in Section 3 ignores short-duration TCP flows ("mice") as well as UDP flows. Can we extend the duality model to understand this richer reality? Modeling HTTP flows as on-off sources, it is shown in [7] that one can associate utility functions to these finite-duration flows, called application utility, and explicitly relate these utility functions to the TCP utility functions at the transport layer. Global stability, in the absence of feedback delay, of a network of finite-duration TCP's is analyzed in [13]. Even though these papers consider the case where finite-duration flows arrive and depart randomly, both assume that TCP converges *instantly* to solve the utility maximization problem (2)–(3) as the number of flows changes. Building on insights from these works, we will develop new ways to understand stochastic effects, including randomly arriving and departing finite-duration flows, and to design protocols that are robust against, or can exploit, randomness inherent in large scale networks.

5 Proposed approach

The pilot project is both a first step toward our goal of deployment and a validation of our approach to achieve it. It *comprehensively* addresses a specific problem, the performance and stability of TCP in ultrascale networks. It builds on four years of theoretical work by various groups around the world, starting from papers in 1998, e.g., [30, 41], and culminates in the preliminary FAST kernel prototype developed at Caltech last year [26]. It is the beginning of our ambitious plan to develop and deploy scalable and robust protocols for ultrascale networking. It is the combination of theory, implementation, and experiments that is truly unique and that gives us a real opportunity to make an impact. We will follow the same approach, and seize this opportunity to make it happen.

5.1 Theory

Achieving our goal demands a fundamental understanding of the issues discussed in the last section, and implementable solutions that are based on sound theory.

These issues deal with performance and stability of general large scale networks under end-to-end control, the co-existence of FAST with current protocols, its interaction with dynamic routing, and its robustness and adaptation to noisy information and stochastic effects. As discussed in the last section, the analysis, design, and optimization of large scale distributed nonlinear feedback system with delay and randomness present formidable difficulties. We will apply, and extend, mathematical tools from control theory, optimization theory, game theory, information theory, stochastic processes, to algorithms and complexity. This requires a sustained investment and significant groundbreaking in theory development and provides a unique educational opportunity for graduate students and postdocs involved in the project.

The pilot project illustrates the kind of theoretical results we expect to develop in this project. It also illustrates their practical impact: theory provides the foundation for protocol design and plays an indispensable role in implementation, providing a framework to understand issues, clarify ideas and suggest directions, leading to a more robust and better performing implementation.

5.2 Implementation

Despite achieving a performance that is 2–4 times that of the current protocol, FAST kernel is still a very preliminary proof of concept, far from being deployable in production on a large scale. We have focused on performance and stability in its design and implementation. We have not built in robustness nor fairness with current protocol, we do not understand its interaction with dynamic routing, or its performance with mice traffic or in the presence of noise. As our theoretical understanding of these issues advances, the resultant algorithms and insights will be incorporated into the kernel. This may require substantial redesign of the software architecture from time to time, which is inevitable, and indeed necessary, as theory development, implementation and experiments *must* inform and influence each other intimately, a key feature that will be enabled by an ITR medium project.

Implementation has to deal with effects that are ignored in mathematical (and simulation) models. For example, theory often assumes the control algorithm has available to it reliable end-to-end delay measurement, which turns out to be very hard to obtain in practice. A large amount of our effort in the current implementation deals with this problem, especially during loss recovery. We often discover phenomena that are hard to explain from only end-to-end measurements. In this project, we will develop/integrate instrumentation tools and infrastructure, such as those developed by Cottrell's (Senior Personnel) group in Internet End-to-end Performance Monitoring (IEPM) [10, 48, 9], Web100 [69] and Net100 [51], to better monitor throughput, delays, losses and queue lengths. This will provide us with detailed understanding of dynamics and complications in real networks, critical for implementing a more robust and better performing kernel.

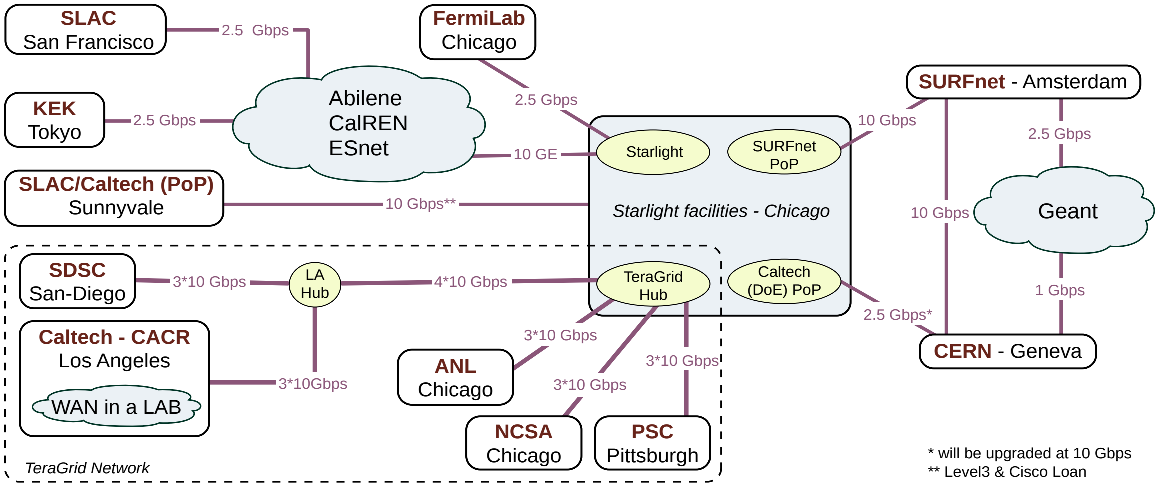
We will continue to do our development on the Linux platform. As theory, design and implementation mature, we will port the kernel to other major platforms such as BSD, Microsoft, IBM, HP. The timing will also be determined by the need to support deployment and our effort to work with commercial companies.

5.3 Experiment

We are partnering with a host of institutes and projects in the US, Europe and Japan, both to leverage on their extensive national and international infrastructure and to test, and eventually deploy, the protocols to be developed in this project in their toolkits and environments.

The joint experimental infrastructure we envision is shown in Figure 4. We have already started to work with various groups, shown in Table 2, to integrate existing infrastructure into this joint facility.

Encouraged by our experimental results at SC2002, we started planning for the next experiment in the middle of the Conference back in November 2002. Per flow throughput in SC2002 experiments was



* will be upgraded at 10 Gbps
** Level3 & Cisco Loan

Figure 4: Integrated global experimental infrastructure

limited by the Gigabit Ethernet card at the servers to 1Gbps (we achieved 925Mbps). The next experiment will use Intel's 10Gigabit experimental Ethernet card on the same facilities, and we aim to achieve 4Gbps with a single TCP flow, limited by the PCI bus. The experiment depends critically on loaned equipment: OC192 circuit between Sunnyvale and Chicago from Level(3), 10GE router modules from Cisco, and 10GE network interface cards from Intel. Even though Wu of LANL (Senior Personnel) managed to secure 1/3 of the world's experimental cards, it is a highly nontrivial feat to align the availability of loaned equipment from three different vendors. After three months' of intensive coordination efforts by more than 16 people in 7 organizations, this experiment will finally start mid February. We expect it to both break new grounds and provide information on the current FAST prototype that will be critical to our design in the next stage.

This illustrates the difficulty and the amount of resources required to organize a leading-edge global experiment on an ad hoc basis. The joint experimental infrastructure outlined above will greatly simplify this task by pulling together more permanent facilities. We now highlight two steps toward the establishment of this infrastructure that we have already initiated and will continue to expand during the project.

5.3.1 Abilene testbed

We will build an Abilene testbed that will initially consist of a dozen sites, have a national coverage, and make use of OC48 and OC192 paths which are being installed in the backbone. We intend to place a single server with Gigabit Ethernet card at OC12 sites, and 1-3 such servers at each OC48 sites. The Abilene testbed will provide us with a production network, shared by more than 200 universities, that has a rich topology and heterogeneous paths for development and testing of our protocols. We are contacting target sites and have already had agreement from Georgia Tech (SoX GigaPop, Cas D'Angelo), University of Washington (PNW GigaPoP, David Richardson), North Carolina Internet2 Technology Evaluation Center (NC-ITEC GigaPoP, John Moore and John Streck), and Pittsburgh Supercomputing Center (Matt Mathis and Ken Goodwin), and University of Florida (Dave Pokorney), to host our servers.

5.3.2 TeraGrid

TeraGrid is a NSF-funded ($88M) 40Gbps facility connecting supercomputing centers at ANL, NCSA, SDSC, CACR, and PSC. A collaboration between HENP application (headed by co-PI Bunn) and the FAST project (headed by PI Low) has been selected as a TeraGrid Flagship Application and will be one of the first applications on the TeraGrid. The computation intensive applications, which include simulations of many millions of LHC events for studies of Higgs particle decays, will utilize computing resources distributed in the TeraGrid, and require the movement of TeraByte-scale result sets or object collections

Institute/Project	Contact	Remarks
Abilene, Internet2	Guy Almes, Chief Engineer Steve Corbató, Director, Backbone Infrastructure D. E. Van Houweling, President, CEO, UCAID, Internet2	co-PI Letter of Support Letter of Support
NSF TeraGrid	Linda Winkler, Network Architect Dan Reed, Chief Architect, TeraGrid Charlie Catlett, Executive Director, TeraGrid	SP Letter of Support Letter of Support
Caltech CACR	James Pool, Executive Director	Letter of Support
SLAC	Les Cottrell, Assistant Director	SP
CERN (Geneva)	Olivier Martin, Head, DataTAG	Letter of Support
StarLight	Linda Winkler, Project Lead, Networking	SP
FermiLab	Michael Ernst, Head, CMS Project	Already testing FAST
LANL	Wu Feng, Team Leader, Adv. Networking Tech.	SP
GGF	Charlie Catlett, Chair	Letter of Support
WAN in Lab	Steven Low, Director, Caltech Networking Lab	PI
Cisco	Bob Aiken, Doug Walsten, Steven Yip	co-PI (Yip)
Level(3)	Paul Fernes	

Table 2: Partner institutes and projects for FAST experiments and deployment. SP: Senior Personnel; CACR: Center for Advanced Computing Research; SLAC: Stanford Linear Accelerator Center; CERN: European Organization for Nuclear Research; LANL: Los Alamos National Lab; GGF: Global Grid Forum; TeraGrid: involves NCSA, SDSC, ANL, Caltech, PSC.

between TeraGrid sites. The FAST kernel will be critical to provide the required throughput to satisfy the client analysis codes. This will illustrate Newman and Bunn's concept of Petabyte-scale Data Grids with Terabyte transactions.

5.4 Deployment

We will start with testing and deploying the FAST TCP kernel in the HENP community that needs ultrascale networking today. Indeed, trials of our current extremely preliminary prototype are already being conducted or set up at SLAC, FermiLab, ANL, LANL, TeraGrid, Abilene, CERN (Switzerland), University of Manchester (UK), INRIA (France), and KEK (Japan), all of which are our partners in FAST experiments. As our kernel improves and deployment experience matures, the circle of deployment will grow to other communities. This may coincide with the need for ultrascale networking beyond the research community, e.g., in the entertainment industry, a few years into the future.

We will work with Internet2, NSF TeraGrid, DataTAG (EU project), IETF, GGF (Global Grid Forum), and various HENP and Grid projects around the world to test and integrate FAST into their toolkits and standards. We have already initiated contacts and received strong supports from the leaders of these communities; see the attached 8 letters of support.

6 Timeline, broader impacts, prior awards

6.1 Timeline

The integrated approach, where theory, implementation, experiments and deployment inform and influence each other intimately, is absolutely the key to achieving our goal. While all these activities will be pursued in each of the four years, in an integrated manner, the first year will have a greater emphasis on theory and algorithm while the last year on standards and deployment.

We expect issues (from theory, implementation to experiment) of performance and stability (Section 4.2) and of interaction with current TCP (Section 4.3) will be resolved in Year 1, issues of interaction with routing (Section 4.4) resolved in Year 2, issues of noisy information (Section 4.5) and stochastic effects (Section 4.6) resolved in Year 3. Year 4 will focus on new issues that will arise in large scale deployment and standards activities. We expect the Abilene and TeraGrid testbeds will be up and running in years 1–2 and will be used extensively throughput the project duration.

6.2 Broader impacts: education and outreach

The project will provide a unique training to graduate and undergraduate students. Through theory development, the students will learn to become mathematically sophisticated; through implementation and experiments, they will remain well grounded in practical networking issues and develop useful skills. The graduates of this project will be equipped to make novel contributions to future networking research.

Results developed in this project, and around the world, will be incorporated both into an advanced networking course the PI teaches at Caltech and a new course being co-developed by the PI and colleagues at Caltech, aimed at bringing together faculty and students interested to work on problems at the boundaries of control, communication, and computing. These courses will be offered starting next quarter through the Departments of EE, CS, and Control and Dynamical Systems at Caltech.

Through the newly established Information Science and Technology Institute at Caltech and its Center for the Mathematics of Information (CMI), we will pursue outreach activities that include workshops to bring together collaborators, colleagues, and students around the world and technology leaders in industry for focused study in selected topics.

6.3 Results on prior NSF awards

Low and Doyle: We have two active NSF grants:

1. NSF ITR Small award, ANI-0113425 (2001-04), “Optimal and Robust TCP Congestion Control”

2. NSF STI, ANI-0230967, (2002-05), “Multi-Gbps TCP: Data Intensive Networks for Science & Engineering”

These grants fund the pilot project that both raises many unresolved issues, which must be addressed in order to support large scale deployment of FAST protocols, and validates our integrated approach to address these issues. Unresolved issues discussed in Section 4.2 will continue to be funded by these ongoing grants, whereas those discussed in Sections 4.3–4.6 will be funded by the current proposal.

To date, our research has produced 7 journal papers [40, 45, 39, 18, 44, 67, 42] and 10 conference papers [8, 66, 61, 55, 60, 71, 32, 38, 43, 1]. We have also given many invited talks at workshops and universities, and conducted tutorials on TCP congestion control at leading networking conferences such as IEEE Infocom (2001), ACM Sigmetrics (2001), and ACM Sigcomm (2001). We have demonstrated our first prototype in Nov 2002 which achieved 925Mbps throughput (95% utilization) on an intercontinental network, doubling the previous record at the time.

Bunn: GriPhyN, iVDGL, PPDG, European DataGrid, NSF-KDI “Accessing Large Data Archives in Astronomy and Particle Physics”, GIOD, MONARC, development of ODBMS-based scalable reconstruction and analysis prototypes working seamlessly over WANs; Grid Data Management Pilot distributed file service used by CMS in production (together with EU DataGrid); Grid-optimized client-server data analysis prototype development, MONARC simulation systems and application to optimized inter-site load balancing using Self Organizing Neural Nets; development of a scalable execution service; modeling CMS Grid workloads; optimize bit-sliced TAGs for rapid object access; development of a TeraGrid prototype for seamless data production between Caltech, Wisconsin and NCSA, Bandwidth Intensive demonstrations of Object Collection analysis by physicists (SC2001).

7 Management plan

7.1 Personnel

To achieve our ambitious goal requires a concerted effort on all four fronts: theory, implementation, experiment, and deployment. We have assembled a team with an excellent mix of knowledge, skills and experience to cover the range of proposed activities. The expertise of PI/co-PI/Senior Personnel is:

1. **Theory:**
   - **John Doyle (lead)**: control theory, complex system theory
   - Steven Low: resource allocation, networking
   - Fernando Paganini: distributed control, systems theory
2. **Implementation:**
   - Werner Almesberger: Linux guru[3], protocol design
   - Cheng Jin: kernel programming, networking
   - **Steven Low (lead)**: resource allocation, networking
3. **Experiment:**
   - **Guy Almes (lead)**: Chief Engineer of Internet2, ultrascale networking
   - Les Cottrell: Assistant Director of SLAC, performance monitoring
   - Wu Feng: Team Leader of Advanced Networking Technology, LANL, ultrascale networking
   - Stanislav Shalunov: Engineer of Internet2, ultrascale networking
   - Linda Winkler: Network Architect of TeraGrid and StarLight, ultrascale networking
   - Steven Yip: Department Head of Network Engineering, Cisco, optical networking
   - Implementation team
4. **Deployment:**
   - **Julian Bunn (lead)**: HENP, ultrascale applications
   - Implementation team and Experiment team

Other key personnel and graduate students involved in the projects include V. Doraiswami (Cisco), C. Hu (CS, Caltech), H. Newman (Physics, Caltech), S. Ravot (Physics, Caltech/CERN), S. Singh (Physics, Caltech), D. Wei (CS, Caltech).

7.2 Management

As shown above, the core project team is divided into four areas, with overlapping personnel, each led by a PI/co-PI. Each area leader is responsible for coordinating activities and budget in his area. The PI is responsible for overall budget, administration and outreach. This is summarized in Table 3.

The theory and implementation activities are done at Caltech and UCLA (both in LA area). These groups have already had a significant joint research activities and meet regularly at Caltech and UCLA.

Experiments and deployment activities are by their nature distributed nationally and internationally, and must be coordinated using emails, phone calls, and video conferencing tools. This geographically dispersed team has already established a strong long-distance working relationship through our successful

[3] See `http://www.almesberger.net/cv/projects.html` for past projects.

Activity	Lead	Affiliation
Theory	John Doyle (co-PI)	Caltech
Implementation	Steven Low (PI)	Caltech
Experiment	Guy Almes (co-PI)	Internet2
Deployment	Julian Bunn (co-PI)	Caltech
Outreach	Steven Low (PI)	Caltech

Table 3: Management structure

demonstration at SC2002 and the upcoming 10Gbps experiment. For instance, the SC2002 demonstration in November 2002 was organized and conducted by a dedicated team of 9 people but involved 30 more people in 10 organizations. We will hold a project meeting, in conjunction with industry workshop, twice a year to review the status and coordinate future plans.

7.3 Collaborations

Besides the core project team, we are setting up collaboration with several institutes and projects both on theory development and on experimentation and deployment. Our existing collaborations on experiment and deployment with various institutes and projects is discussed in Sections 5.3–5.4. We now describe our interaction with two local theory groups.

Caltech has established a new Center for the Mathematics of Information (CMI), which will consist of approximately 10-12 faculty (including Doyle and Low). CMI will create a dedicated community of mathematicians, engineers and scientists to formulate a new way of thinking about information. Fundamental new ideas will emerge from this effort to influence all of information science and technology. Our project will both contribute to this effort and benefit tremendously from interacting with other researchers in CMI.

The mission of the NSF-funded Institute for Pure and Applied Mathematics (IPAM) at UCLA is to make connections between mathematics and a broad spectrum of scientific and engineering problems and to launch new collaborations. John Doyle and Walter Willinger have organized a quarter-long program at IPAM on Large Scale Communication Networks in 2002, during which rapid progress on the pilot project was made. We will continue to participate intimately in IPAM activities.

Our existing collaborative relationship with strategically important institutes is illustrated by the 8 Letters of Supports, summarized in Table 4. We will establish collaborations with other key application communities and technology leaders during the course of this project.

Letter writer	Affiliation
Theory	
Mark Green	Director, NSF Institute for Pure & Applied Math (IPAM), UCLA
Walter Willinger	AT&T Labs – Research and IPAM
Experiment & Deployment	
Charlie Catlett	Executive Director, NSF TeraGrid; Chair, GGF
Steve Corbató	Director, Backbone Infrastructure, Internet2
Olivier Martin	Head, DataTAG, CERN
James Pool	Executive Director, CACR
Dan Reed	Chief Architect, NSF TeraGrid; Director, NCSA
Van Houweling	President & CEO, UCAID, Internet2

Table 4: Existing collaboration and Letters of Support

References

[1] S. Athuraliya and S. H. Low. An empirical validation of a duality model of TCP and queue management algorithms. In *Proceedings of Winter Simulation Conference*, December 2001.

[2] F. Baccelli and D. Hong. AIMD, fairness and fractal scaling of TCP traffic. In *Proceedings of IEEE Infocom*, June 2002.

[3] F. Baccelli and D. Hong. Interaction of TCP flows as billiards. In *Proceedings of IEEE Infocom*, April 2003.

[4] D. Bertsekas. *Nonlinear Programming.* Athena Scientific, 1995.

[5] L. S. Brakmo and L. L. Peterson. TCP Vegas: end-to-end congestion avoidance on a global Internet. *IEEE Journal on Selected Areas in Communications*, 13(8):1465–80, October 1995. `http://cs.princeton.edu/nsg/papers/jsac-vegas.ps`.

[6] R. W. Brockett and D. Liberzon. Quantized feedback stabilization of linear systems. *IEEE Trans. on Automatic Control*, 45:1279–1289, 2000.

[7] C. S. Chang and Z. Liu. A bandwidth sharing theory for a large number of HTTP-like connections. In *Proceedings of IEEE Infocom*, June 2002.

[8] H. Choe and S. H. Low. Stabilized Vegas. In *Proc. of IEEE Infocom*, April 2003. `http://netlab.caltech.edu`.

[9] L. Cottrell. Internet End-to-end Performance Monitoring (IEPM). `http://http://www-iepm.slac.stanford.edu/monitoring/`, 2000.

[10] R. L. Cottrell, C. Logg, and I.-H. Mei. Experiences and results from a new high performance network and application monitoring toolkit. submitted to the Passive and Active Measurements Conference, 2003.

[11] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* John Wiley & Sons, 1991.

[12] M. E. Crovella and A. Bestavros. Self–similarity in World Wide Web traffic: evidence and possible causes. *IEEE/ACM Transactions on Networking*, 5(6):835–846, 1997.

[13] G. de Veciana, T. J. Lee, and T. Konstantopoulos. Stability and performance analysis of networks supporting elastic services. *IEEE/ACM Transactions on Networking*, 9(1):2–14, February 2001.

[14] S. Deb, S. Shakkottai, and R. Srikant. Stability and convergence of TCP-like congestion controllers in a many-flows regime. In *Proceedings of the IEEE Infocom*, April 2003.

[15] S. Deb and R. Srikant. Congestion control for fair resource allocation in networks with multicast flows. In *Proceedings of the IEEE Conference on Decision and Control*, December 2001.

[16] S. Deb and R. Srikant. Global stability of congestion controllers for the Internet. In *Proc. of the IEEE Conference on Decision and Control*, December 2002.

[17] D. Delchamps. Stabilzing a linear system with quantized state feedback. *IEEE Trans. on Automatic Control*, 35:916–924, 1990.

[18] A. Elwalid, C. Jin, S. H. Low, and I. Widjaja. MATE: Multi-protocol Adaptive Traffic Engineering. *Computer Networks Journal*, 40(6), December 2002.

[19] V. Firoiu, J.-Y. L. Boudec, D. Towsley, and Z.-L. Zhang. Theories and models for internet quality of service. *Proceedings of IEEE, special issue on Internet Technology*, August 2002.

[20] R. G. Gallager. *Information Theory and Reliable Communications.* John Wiley & Sons, 1968.

[21] A. Goel, A. Meyerson, and S. Plotkin. Distributed admission control, scheduling, and routing with stale information. In *Proc. of ACM-SIAM Symposium on Discrete Algorithms*, 2001.

[22] C. Hollot, V. Misra, D. Towsley, and W.-B. Gong. A control theoretic analysis of RED. In *Proceedings of IEEE Infocom*, April 2001. `http://www-net.cs.umass.edu/papers/papers.html`.

[23] H. Ishii and B. Francis. Quadratic stabilization of sampled-data systems with quantization. In *Proc. of IFAC*, 2002.

[24] H. Ishii and B. A. Francis. *Limited Data rate in control systems with networks.* Springer-Verlag, 2002.

[25] V. Jacobson. Congestion avoidance and control. *Proceedings of SIGCOMM'88, ACM*, August 1988. An updated version is available via `ftp://ftp.ee.lbl.gov/papers/congavoid.ps.Z`.

[26] C. Jin, D. Wei, S. H. Low, G. Buhrmaster, J. Bunn, D. H. Choe, R. L. A. Cottrell, J. C. Doyle, H. Newman, F. Paganini, S. Ravot, and S. Singh. Fast kernel: Background theory and experimental results. In *First International Workshop on Protocols for Fast Long-Distance Networks*, February 2003.

[27] K. Kar, S. Sarkar, and L. Tassiulas. A low-overhead rate control algorithm for maximizing aggregate receiver utility for multirate multicast sessions. In *Proceedings of SPIE-ITCOM 2001*, 2001.

[28] K. Kar, S. Sarkar, and L. Tassiulas. Optimization based rate control for multirate multicast sessions. In *Proceedings of IEEE Infocom*, April 2001.

[29] K. Kar, S. Sarkar, and L. Tassiulas. A primal algorithm for optimization based rate control for unicast sessions. In *Proceedings of IEEE Infocom*, April 2001.

[30] F. P. Kelly, A. Maulloo, and D. Tan. Rate control for communication networks: Shadow prices, proportional fairness and stability. *Journal of Operations Research Society*, 49(3):237–252, March 1998.

[31] P. Key, F. Kelly, and S. Zachary. Distributed admission control. *IEEE Journal on Selected Areas in Communications*, 16:2617–2628, 2000.

[32] K. Kim and S. H. Low. Design of receding horizon AQM in stabilizing TCP with multiple links and heterogeneous delays. In *Proceedings of the 4th Asian Control Conference*, September 2002.

[33] S. Kunniyur and R. Srikant. End–to–end congestion control schemes: utility functions, random losses and ECN marks. In *Proceedings of IEEE Infocom*, March 2000. `http://www.ieee-infocom.org/2000/papers/401.ps`.

[34] S. Kunniyur and R. Srikant. Designing AVQ parameters for a general topology network. In *Proceedings of the Asian Control Conference*, September 2002.

[35] S. Kunniyur and R. Srikant. A time-scale decomposition approach to adaptive ECN marking. *IEEE Transactions on Automatic Control*, June 2002.

[36] R. La and V. Anantharam. Charge-sensitive TCP and rate control in the Internet. In *Proceedings of IEEE Infocom*, March 2000. `http://www.ieee-infocom.org/2000/papers/401.ps`.

[37] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the self–similar nature of Ethernet traffic. *IEEE/ACM Transactions on Networking*, 2(1):1–15, 1994.

[38] V. H. Li, Z.-Q. Liu, Z.-Q. He, and S. H. Low. Active queue management to improve TCP performance over wireless networks. In *Proceedings of SPIE ITCom*, July–August 2002.

[39] S. H. Low. Network flow control. In J. Proakis, editor, *Encyclopedia of Telecommunications*, pages 1625–31. Wiley, December 2002.

[40] S. H. Low. A duality model of TCP and queue management algorithms. *IEEE/ACM Trans. on Networking, to appear*, October 2003. `http://netlab.caltech.edu`.

[41] S. H. Low and D. E. Lapsley. Optimization flow control, I: basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, 7(6):861–874, December 1999. `http://netlab.caltech.edu`.

[42] S. H. Low, F. Paganini, and J. C. Doyle. Internet congestion control. *IEEE Control Systems Magazine*, 22(1):28–43, February 2002.

[43] S. H. Low, F. Paganini, J. Wang, S. A. Adlakha, and J. C. Doyle. Dynamics of TCP/RED and a scalable control. In *Proc. of IEEE Infocom*, June 2002. `http://netlab.caltech.edu`.

[44] S. H. Low, L. Peterson, and L. Wang. Understanding Vegas: a duality model. *J. of ACM*, 49(2):207–235, March 2002. `http://netlab.caltech.edu`.

[45] S. H. Low and R. Srikant. A mathematical framework for designing a low-loss, low-delay internet. In E. Altman and L. Wynter, editors, *Networks and Spacial Economics, special issue on "Crossovers between transportation planning and telecommunications"*. 2003.

[46] D. G. Luenberger. *Linear and Nonlinear Programming, 2nd Ed.* Addison-Wesley Publishing Company, 1984.

[47] L. Massoulie and J. Roberts. Bandwidth sharing: objectives and algorithms. In *Infocom'99*, March 1999. `http://www.dmi.ens.fr/\%7Emistral/tcpworkshop.html`.

[48] W. Matthews and L. Cottrell. Achieving high data throughput in research networks. In *Computing in High Energy and Nuclear Physics*, September 2001.

[49] J. Mo, R. La, V. Anantharam, and J. Walrand. Analysis and comparison of TCP Reno and Vegas. In *Proceedings of IEEE Infocom*, March 1999.

[50] J. Mo and J. Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking*, 8(5):556–567, October 2000.

[51] Net100. `http://www.net100.org`, 2001.

[52] Newman and Price. Report of the TransAtlantic Network (TAN) Committee, October 2001. `http://gate.hep.anl.gov/lprice/TAN`.

[53] H. B. Newman. Data intensive grids and networks for high energy and nuclear physics. *InterAct Magazine*, September 2002.

[54] F. Paganini, J. C. Doyle, and S. H. Low. Scalable laws for stable network congestion control. In *Proceedings of Conference on Decision and Control*, December 2001. `http://www.ee.ucla.edu/~paganini`.

[55] F. Paganini, Z. Wang, S. H. Low, and J. C. Doyle. A new TCP/AQM for stability and performance in fast networks. In *Proc. of IEEE Infocom*, April 2003. `http://netlab.caltech.edu`.

[56] V. Paxson and S. Floyd. Wide–area traffic: the failure of Poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, 1995.

[57] P. Ranjan, E. H. Abed, and R. La. Nonlinear instabilities in TCP-RED. In *Proceedings of IEEE Infocom*, June 2002.

[58] S. Shakkottai and R. Srikant. Mean FDE models for Internet congestion control. In *Proceedings of the IEEE Infocom*, June 2002.

[59] R. Srikant. Control of communication networks. In T. Samad, editor, *Perspectives in Control Engineering: Technologies, Applications, New Directions*, pages 462–488. IEEE Press, 2000.

[60] A. Tang, C. Florens, and S. H. Low. An emprirical study on the connectivity of ad hoc networks. In *Proceedings of IEEE Aerospace Conference*, March 2003.

[61] A. Tang, J. Wang, and S. H. Low. Understanding CHOKe. In *Proc. of IEEE Infocom*, April 2003. `http://netlab.caltech.edu`.

[62] P. Tinnakornsrisuphap and A. Makowski. Limit behavior of ECN/RED gateways under a large number of TCP flows. In *Proceedings of the IEEE Infocom*, April 2003.

[63] A. Veres and M. Boda. Chaotic nature of TCP congestion control. In *Proceedings of IEEE Infocom*, 2000.

[64] G. Vinnicombe. On the stability of end-to-end congestion control for the Internet. Technical report, Cambridge University, CUED/F-INFENG/TR.398, December 2000.

[65] G. Vinnicombe. On the stability of networks operating TCP-like congestion control. In *Proc. of IFAC World Congress*, 2002.

[66] J. Wang, L. Li, S. H. Low, and J. C. Doyle. Can TCP and shortest-path routing maximize utility? In *Proc. of IEEE Infocom*, April 2003. `http://netlab.caltech.edu`.

[67] W. Wang, M. Palaniswami, and S. H. Low. Optimal flow control and routing in multiple paths networks. *Performance Evaluation*, 2002.

[68] Z. Wang and F. Paganini. Global stability with time delay in network congestion control. In *Proc. of the IEEE Conference on Decision and Control*, December 2002.

[69] Web100. `http://www.web100.org`, 2000.

[70] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson. Self–similarity through high variability: statistical analysis of Ethernet LAN traffic at the source level. *IEEE/ACM Transactions on Networking*, 5(1):71–86, 1997.

[71] Q. Yin and S. H. Low. On stability of REM algorithm with uniform delay. In *Proceedings of IEEE Globecom*, November 2002.

[72] X. Zhu, J. Yu, and J. C. Doyle. Heavy tails, generalized coding, and optimal web layout. In *Proceedings of IEEE Infocom*, April 2001.