Research Statement

Matthew Andrews

My research lies in the broad areas of algorithm design and networking with a particular interest in approximation algorithms and scheduling theory. During my time at Bell Labs I have been involved with a number of practical network optimization projects, many of them related to Lucent's wireless and optical network businesses. More recently I have also been heavily involved with a DARPA project that aims to develop next-generation networking protocols for mobile ad-hoc networks. These projects have motivated many of my theoretical results and have also required expertise in algorithm implementation and extensive software simulation.

Much of my recent work has concerned the approximability of a set of fundamental routing problems. These include the buy-at-bulk network design problem, the unsplittable flow problem (which includes the edge-disjoint paths problem) and the congestion minimization problem. Another focus of my work is a theory of worst-case behavior in packet networks known as Adversarial Queuing Theory. In this area a particular interest is the performance of the First-In-First-Out (FIFO) scheduling protocol. I have also worked extensively on a number of problems related to resource management in cellular and ad-hoc wireless networks. These problems are typically made difficult by the time-varying nature of wireless communications channels.

In this statement I describe my past work together with the problems I would like to work on in the future. For the sake of readability, I do not list my coauthors in the main text. My coauthors for each paper may be found in the bibliography.

1 Complexity of network design and routing

A topic that has interested me greatly in recent years is the complexity of network design and routing. Consider a communication network and a demand matrix that specifies how much traffic is to be carried. Two interesting questions arise. First, what is the best way to design a network so that all the demands can be carried? Second, once the network has been built, what is the best way to route the demands on the network? Many of the intrinsic issues associated with these questions are captured in the following three fundamental problems.

- **Buy-at-bulk network design.** For this problem we must design a network of minimum cost that can route all demands. For each deployed link in the network, the link cost is a subadditive (e.g. concave) function of the link capacity. The cost of the network is the sum over all links of the link costs. The subadditivity of the link costs reflects the fact that the cost of bandwidth often has economies of scale. Hence it makes sense to aggregate demands so that they can be routed on a high-capacity link. The fact that we must decide where the bandwidth should be aggregated is the component of the problem that makes it difficult to solve.

- **Unsplittable flow.** In this problem each edge has a fixed capacity and at most one path is used for any demand. The aim is to route as many of the demands as possible, subject to the constraint that no link capacity is violated. A special case of this is the well-known *edge-disjoint paths* problem (EDP) in which all the demands and link capacities have size 1 and so all demands must be routed on edge-disjoint paths.

- **Minimum congestion routing.** Here the aim is to choose a single path for all demands in such a way that the maximum number of demands routed through any link is minimized.

All three of the above problems are known to be NP-hard. In particular, EDP and the minimum congestion routing problem date back to the original collections of NP-hard problems. For this reason the focus of recent research has been on approximation algorithms. (An α-approximation algorithm for an optimization problem is a polynomial-time algorithm that always returns a solution that is within a factor α of optimal.) The current best approximation ratios for undirected graphs are shown in the middle column of Figure 1 and are quite high. The most critical question, and the focus of a great deal of research, is whether or not any of the three problems has a constant-factor approximation.

Undirected graphs	Upper Bound	Lower Bound
Buy-at-bulk	$O(\log M)$ [34, 36, 39]	$\Omega(\log^{\frac{1}{4}-\varepsilon} M)$ [4]
Unsplittable flow	$O(M^{1/2})$ [37]	$\Omega(\log^{\frac{1}{2}-\varepsilon} M)$ [30, 12]
Minimum congestion routing	$O(\log M/\log\log M)$ [43]	$\Omega((\log\log M)^{1-\varepsilon})$ [29]

Figure 1: Bounds on the approximability of network design and routing problems in undirected graphs. Here M is the number of links in the network. My research has focused on deriving the lower bounds.

My work

We have been able to resolve this critical open question by showing that unless NP has efficient algorithms, none of the three problems has a constant-factor approximation. More precisely, let M be the number of links in the network and let $ZPTIME(n^{polylog\ n})$ be the set of languages that have randomized algorithms that always give the correct answer and have expected running time $n^{polylog\ n}$. In a sequence of papers [4, 30, 29, 12] we showed that unless $NP \subseteq ZPTIME(n^{polylog\ n})$, then on undirected graphs with M links buy-at-bulk network design has no $\Omega(\log^{\frac{1}{4}-\varepsilon} M)$-approximation, unsplittable flow has no $\Omega(\log^{\frac{1}{2}-\varepsilon} M)$-approximation and minimum congestion routing has no $\Omega((\log\log M)^{1-\varepsilon})$-approximation. These bounds are listed in the right column of Figure 1.

All of these results are based on the novel idea that I introduced in [4] of defining an instance on a graph that almost has high girth. The important feature of a high-girth graph is that for any demand it has at most one short path. Moreover, for any "good" routing most demands have to be routed on short paths since otherwise they would consume too much bandwidth. This implies that for most demands the routing for each demand is essentially fixed. This allows us to relate the routing problems to other hard problems such as maximum independent set.

We have extended this work in a number of ways. In [12] we allowed each link to have capacity $c = o(\log\log M/\log\log\log M)$ and showed that in this case the unsplittable flow problem does not have an $\Omega(\log^{\frac{1-\varepsilon}{c+1}} M)$-approximation. More recently we have been looking at routing in directed graphs. We were able to show in [32] that the directed congestion minimization problem is $\Omega(\log^{1-\varepsilon} M)$-hard to approximate. This almost matches the upper-bound of $O(\log M/\log\log M)$ due to Raghavan and Thompson [43]. (See Figure 2.)

In the papers [27, 31], we used high-girth graphs to show that designing optical networks with wavelength-division multiplexing is hard. The main additional feature of such problems is that in addition to choosing a route for a demand, we also need to assign it a wavelength under the constraint that no two demands that share a fiber are assigned the same wavelength.

Directed graphs	Upper Bound	Lower Bound
Minimum congestion routing	$O(\log M / \log \log M)$ [43]	$\Omega((\log M)^{1-\varepsilon})$ [32]

Figure 2: Bounds on the approximability of the congestion minimization problem in directed graphs.

Plans for the future

There are a number of directions in which I would like to take this work. First, for undirected graphs, none of the lower bounds match the current best known upper bounds. Moreover, for unsplittable flow and congestion minimization, the gaps are still exponential. I would very much like to close these gaps. Second, I believe that the general technique of creating hard instances on high-girth graphs is a powerful generally applicable technique. I am curious as to whether it can be applied to obtain inapproximability results for other NP-hard problems. I would also like to study how the approximability of network design and routing change when the recent, powerful technique of network coding [1] can be utilized.

2 Scheduling Theory

I first worked on scheduling theory in graduate school and I have maintained a strong interest in the subject. Many of the questions I have studied concern a topic known as *Adversarial Queuing Theory*. This area of research looks at the performance of scheduling algorithms in networks under worst-case assumptions on traffic patterns. A key question is whether or not a scheduling algorithm is *stable*. That is, can the queue sizes in the network grow without bound even if the injection pattern does not overload any link. If stability does indeed hold, other questions we are interested in include minimizing the end-to-end delay of traffic through the network and also minimizing the aggregate queue size. A related problem is to choose routes for packets in such a way that the scheduling algorithms work well. I have also recently become interested in analyzing how scheduling algorithms interact with congestion control algorithms such as TCP.

This work can roughly be divided into two types, depending on the admissible traffic that is allowed into the network. In the *permanent sessions model* we require that all packets are injected into a fixed set of flow paths or *sessions*. In the *temporary sessions model* the flow paths into which packets are injected can change over time. A question that I find especially intriguing is the stability of the simple First-In-First-Out (FIFO) scheduling discipline and how it differs in the two models. I believe that FIFO is an important protocol to study because of its simplicity and its wide deployment in packet networks.

My work

Basic scheduling theory. In an early paper [10] we looked at classifying a variety of scheduling protocols in terms of whether or not they lead to network stability. In particular, we showed that a set of protocols including *Longest-in-System* (that always gives priority to the oldest packet in the network) are always stable as long as the arriving traffic does not overload any link in the network. In contrast, there is a simple example that shows that a number of protocols including FIFO are not stable.

FIFO scheduling. The performance of FIFO has interested me throughout my research career. As already mentioned, in [10] we showed using a simple example that FIFO can be unstable in the temporary sessions model where the flow paths can dynamically change.

The question of whether or not FIFO is stable in the permanent sessions model is much more complicated. I believe that this question is of interest for two reasons. First, it determines if the instability in the example of [10] is caused by the dynamically-changing flow paths or if instability can occur even if the

paths remain fixed. Second, the permanent sessions model is the model most often studied in the networking literature and so I feel that it is therefore important to analyze FIFO in that context.

In [5] I was able to show via a complex example that FIFO can indeed be unstable in the permanent sessions model. This demonstrated that dynamically changing paths are not a necessary condition for FIFO instability. One feature of the instability example is that although the network was underloaded, it was only just underloaded, i.e. by a factor of $1 - 3 \times 10^{-9}$. An intriguing question therefore was whether FIFO could be unstable at lower loads. I recently managed to answer this question in the affirmative by showing that in the permanent sessions model, FIFO can be unstable at any network load $\varepsilon > 0$ [8].

End-to-end delay minimization For networks that carry real-time traffic such as voice or video it is desirable that end-to-end delay bounds are kept low. It is therefore important to schedule traffic so that excessive delays are avoided. With respect to the temporary sessions model, one feature of all the stable protocols in [10] is that they have an exponential worst-case bound on end-to-end packet delay. A natural question was whether or not such large delays are necessary in this model. In [13] we showed that in fact there is a simple deterministic, distributed strategy that guarantees polynomial end-to-end delay.

In [14] we looked at the permanent sessions model with the aim of minimizing the end-to-end delay of the packets along each session. For session i we presented a scheduler that achieves a delay of $O(\frac{1}{\rho_i} + d_i)$ where ρ_i is the maximum injection rate into session i and d_i is the length of the path for session i. These bounds are essentially best possible up to constant factors. A more practical version of this algorithm with slightly (i.e. logarithmically) worse delay performance was presented in [21]. In [3] I analyzed the Earliest-Deadline-First (EDF) protocol in a similar context. In [22] we showed how to schedule traffic to meet an arbitrary set of end-to-end delay bounds.

In recent work I looked at the problem of *routing* traffic in order to meet delay bounds, assuming that the end-to-end delay along a path is a function of the aggregate congestion along that path. In [9] I showed that somewhat surprisingly, even if the demands can be split among multiple paths, the optimal routing is NP-hard to calculate exactly. The hardness comes from the fact that we only care about the delay bound on paths on which we route non-zero flow. Choosing this set of paths is a hard combinatorial problem.

Interaction between scheduling and congestion control. A current popular research area is analyzing networking protocols as distributed algorithms for solving network utility maximization problems. An early paper on this subject was by Kelly, Maulloo and Tan [40] who showed that TCP can be viewed as a primal-dual algorithm that converges to the optimal value of a concave optimization problem. This work has since been extended to different versions of TCP and to different networking models by many other researchers (e.g. [42, 41]). However, one common feature of all this work is that it only uses an approximation of the queuing dynamics. In particular, if some flow passes through a queue then its arrival rate at the queue is modeled at all times as the flow injection rate into the network. In reality however, the flow arrival rate at a queue may be different from the external flow injection rate since whenever a queue on a flow path builds up in the network, the arrival rates at all downstream queues are affected.

Together with Alex Slivkins, my summer intern in 2005, I looked at the question of whether the convergence results still hold when the queuing dynamics are explicitly modeled. In [20] we were able to construct an example consisting of a network of (mostly FIFO) queues such that the system oscillates and never converges to the optimal solution of the associated network utility maximization problem. This implies that when looking at the behavior of protocols such as TCP in complex networks it is important to study how the congestion control interacts with the scheduling dynamics.

Other scheduling work Many routers in the Internet are based on an architecture known as an input-queued switch. The basic fabric of these switches is a crossbar and in each time step a *matching* is set up between the inputs and the outputs of the crossbar. The goal of the scheduling algorithm is to choose which

matching to use at each time step. Most previous work on input-queued switches looked at a single switch in isolation. In [23] we studied networks of input-queued switches and showed that a popular algorithm based on maximum weight matchings can actually be unstable in networks consisting of just eight switches. We then presented an alternative algorithm that is stable in networks of input-queued switches for all admissible traffic.

Another issue that arises in Internet routers is that at high speeds, traffic must be switched at a granularity much larger than a packet. In [24] we presented algorithms for scheduling packets when they are switched in large bundles known as envelopes.

In [13] we considered the problem of combined routing and scheduling in packet networks. We presented a simple distributed routing algorithm that ensures stability when used in conjunction with any stable scheduling protocol. In [7] I presented an algorithm for selecting which packets should be served by a network in the case that the offered traffic overloads some links.

Plans for the future

Much of the work that I would like to do in connection with scheduling involves simplifying the "bad" examples that I have constructed in previous work and also determining what are the characteristics of networks that can lead to bad examples. For example, with respect to the work on stability of FIFO, I would like to know if there are simpler examples for which FIFO can be unstable when the flow paths do not change. More generally, I believe it would be intriguing to have a set of necessary and sufficient conditions that would determine whether or not a set of flow paths can lead to instability. The network calculus literature (e.g.[38]) provides some sufficient conditions but in many cases these conditions are very loose. I am intrigued by the question of whether there are a tighter set of conditions that are also necessary.

3 Wireless Network Optimization

During my time at Bell Labs, the majority of my practical work and a significant fraction of my theoretical work has concerned the optimization of wireless networks. During the past year I have been heavily involved with the DARPA Control-Based Mobile Ad-Hoc Networks (CBMANET) program. I am the design leader of a team (including researchers from Princeton, Stanford, University of Illinois, UC Santa Barbara and Lockheed Martin) that is developing a new protocol stack for wireless ad-hoc networks that is based on the theory of network utility maximization. Another particular topic that I have spent a great deal of time on concerns scheduling in a 3rd generation wireless data network with channel feedback. I have developed a number of scheduling algorithms that have been incorporated in Lucent's products. Incorporating a theoretical algorithm into a product that must be highly reliable is an extremely involved process and I was actively involved in the large amounts of code development, simulation and testing that were required.

My work

Ad-hoc wireless networks. As already mentioned, I am part of a team that is working on the DARPA CBMANET program whose aim is to design a new networking stack for wireless ad-hoc networks. One of the motivations of our design is my paper [20] in which we argue that in order to optimize network performance, it is important for the congestion control algorithms and the internal scheduling algorithms to work in collaboration. We are therefore designing an architecture in which each node maintains a separate queue for each destination in the network. The sizes of these queues form a shared state that is used by the congestion control algorithm, the intra-node scheduling algorithm as well as the MAC protocol that resolves contention between different nodes. This enables all these algorithms to work together to solve a joint network optimization problem. Within the project I am the leader of the design and implementation team. I have contributed to the development of all of the resource allocation algorithms and I have defined

the signaling mechanisms so that the necessary state can be exchanged between nodes. I have also been active in implementing the protocols in a comprehensive OPNET simulation.

In related work I have recently been studying theoretical issues that arise in wireless ad-hoc networks. Some of the routing and scheduling algorithms that we are implementing in the CBMANET project are similar in spirit to the "MaxWeight" algorithm that was first proposed by Tassiulas and Ephremides in [44, 45]. The MaxWeight algorithm always tries to minimizes the sum of the squares of the queue sizes in the network. Previous work has shown that this algorithm is robust in the sense that even in an adversarial context, it is stable for multicommodity traffic in static networks [2] and single-commodity traffic in dynamic networks [33]. However, the question of stability in dynamic networks with multicommodity traffic (which is the situation that most often arises in ad-hoc networks) was unresolved. Together with Kyomin Jung, my summer intern in 2006, we recently managed to resolve this question by showing that the MaxWeight algorithm is stable even in this multicommodity dynamic-graph context [15]. Another contribution of [15] was an improvement in the running time analysis of the Awerbuch-Leighton [35] multicommodity-flow algorithm that is based on MaxWeight.

Cellular wireless networks. The basic scheduling problem in cellular networks is best described in reference to a system known as EVDO. Consider a basestation transmitting to a set of users in a cell. In the current EVDO system time is slotted and in each time slot the basestation can transmit data to exactly one user. At all times each user measures the quality of the channel from the basestation and reports back the data rate that the channel can support. Note that this data rate will change over time and be different for different users. The basestation uses these data rates to determine which user should receive data at each time step. The scheduling algorithm needs to balance two conflicting goals. First, it should in general serve a user that can support a high data rate. This ensures good overall system performance. On the other hand the system should not starve a user that has a persistently poor channel.

Wireless scheduling is an extremely active research area. However, many papers use different models and so it is not always easy to compare results. One of the topics that I have studied is how the results that can be proved are different in different models. For example, one can assume that for each user there is always data to transmit (the infinitely backlogged model) or one can assume that data is generated according to some external arrival process. Another distinction is that some work assumes that the wireless channel process is governed by a stationary stochastic process. Other work assumes that the wireless channel process is a "worst-case" process generated by an adversary.

The above definitions give rise to four different models, each of which I have worked on. Some of this work has motivated practical algorithms that have been implemented in Lucent's wireless systems.

- **Infinitely backlogged data. Stationary channel process.** This is the model that motivated the widely used *Proportional Fair* (PF) scheduling algorithm (due to Tse [46]). At each time step PF serves the user with the highest value of instantaneous channel rate divided by long-term service rate. This algorithm is known to maximize the sum of the logarithms of the long-term service rates of each user (the so-called Proportional Fair criterion). However, one of the drawbacks of PF is that it does not provide any guarantees on service to individual users. In [18] we proposed an extension of PF, called Proportional Fair with Minimum Rate guarantees (PFMR) that allows each user to specify a minimum bandwidth that it requires from the system. The ability to do this is critical for real-time traffic such as streaming video. The PFMR algorithm, along with a number of other scheduler enhancements that we have proposed, have been added to Lucent's third-generation wireless offerings.

- **External data arrival process. Stationary channel process.** In the case that data arrives according to a external arrival process, the most important goal is to make sure that the queues remain stable (i.e. that they do not grow without bound). In [6] I was able to show that surprisingly, Proportional Fair is *not* stable. There are simple arrival patterns such that the service rates produced by Proportional Fair

do not match the arrival rates. Hence some queues can become arbitrarily large. As an alternative, in [16] we showed that we can ensure stability if we always try to serve a user that has both a high instantaneous channel rate and a large queue.

- **External data arrival process. Adversarial channel process.** In this context we again want the queues to be stable. However, we now assume that the channel rates are governed by an adversary, rather than a stationary stochastic process. In this case we showed in [24] that if the non-zero channel rates can be arbitrarily small, then in fact there is no online scheduling algorithm that is stable. For the case that the non-zero channel rates are bounded (e.g. if there is a finite set of channel rates), we presented in [24, 26] a *Tracking Algorithm* that keeps track of the adversary's past schedule and tries to emulate it in the future. We showed that this algorithm is stable and always has polynomially bounded queues. In [25] we generalized the Tracking Algorithm to the case of multi-hop wireless networks.

- **Infinitely backlogged data. Adversarial channel process.** I analyzed this scenario in [7] and showed that in contrast to the stationary channel case, the scheduling algorithm does not benefit from knowledge of the channel. In particular, it is impossible to beat an algorithm that simply serves a random user at each time step.

Other work I have done includes measurement of wireless networks. In [28] we describe a tool called DQoS (for "Data Quality-of-Service") that measures CDMA data networks at multiple points in the network and at multiple network layers and synthesizes all the information to give a unified view of how the network is performing. I was also part of a project that used human subjects to determine what aspects of network performance are most important in providing a satisfactory data experience. A report on this work is contained in [11].

As part of my Bell Labs project work I have also studied a number of questions related to admission control and overload control. The goal of much of my scheduling work has been to provide greater control over the Quality-of-Service provided by the network. However, the scheduler needs to be sure that the Quality-of-Service requested by the users in the system can be provided given the current channel conditions. This means that we need an admission control mechanism that prevents new users joining when the system is at full capacity. We also need an overload control mechanism that downgrades the Quality-of-Service level for a user in the system in the case that the channel conditions deteriorate and the current set of users can no longer be supported.

Plans for the future

There are a number of wireless problem that I plan to work on. First, the Tracking algorithm that provides stability in the case of adversarial channels is a fairly complex algorithm. I would like to know if there are any simpler algorithms that have similar properties.

A second problem concerns the best way to schedule data with tight delay requirements in wireless networks. From a practical standpoint, this will be an important issue for the future as carriers try to merge Voice-over-IP (VoIP) networks with wireless networks and hence there will be a need to carry VoIP traffic over wireless.

Third, I would like to continue my study of optimization issues in mobile ad-hoc networks. Much of the work in the CBMANET project is still in its early stages and there are many interesting theoretical questions that are still unresolved, especially with respect to multicast transmissions. For example, a nontrivial problem is to design algorithms for combined congestion control and scheduling in a multicast context that ensure network optimality but which do not require excessive signaling.

4 Other Networking Research

The emergence of content delivery networks during the past few years has greatly improved the performance of the World Wide Web. From 1999 to 2001 I was involved with a large effort at Bell Labs to break into the market for content delivery appliances. In particular, I helped design Lucent's webDNS product which is a DNS server that can route users' web requests to the nearest content server based on proximity. Unlike other devices, our solution is entirely passive and does not require any access to routing tables and does not require any active "pings" between clients and servers.

The algorithms behind the webDNS are presented in [19]. The main idea is to measure round-trip time for the TCP handshake whenever a TCP connection is set up between a client and server. (Hence we generate no extra traffic to a client). These measurements are then sent to the DNS server which uses them to perform a clustering of the IP address space and then assign each cluster to the most appropriate content server. Our assignment algorithm is derived from the notion of *Gittins index* which arises from the theory of "multi-armed bandit problems" in stochastic processes.

I also looked at the problem of caching multimedia streams. This differs from the standard online caching problem in that the streamed object is typically too large to fit into the cache. Our aim is to cache small pieces of a stream in such a way that as many users as possible receive data from the cache rather than the original server. We presented our solutions to this problem in [17].

5 Research summary

In this statement I have described some of my research interests. These largely fall into the field of network optimization, either in the area of network design or in the area of design of online network algorithms such as packet schedulers. I would like to continue these efforts as a professor. In addition, since these are such fertile areas for open problems, I believe that they can provide students at all levels with interesting topics for their own research.

I would also like to take advantage of the greater freedom afforded by a university position to explore other areas of computer science and discrete mathematics. My current position with a telecommunications equipment company has meant that my research has been dominated by problems in networking. I am excited by the possibility that as an academic I could work in new research areas.

References

[1] R. Ahlswede, N. Cai, S. Li, and R. Yeung. Network information flow. *IEEE Transactions on Information Theory*, 46(4):1204–1216, 2000.

[2] W. Aiello, E. Kushilevitz, R. Ostrovsky, and A. Rosen. Adaptive packet routing for bursty adversarial traffic. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 359 – 368, Dallas, TX, May 1998.

[3] M. Andrews. Probabilistic end-to-end delay bounds for Earliest Deadline First scheduling. In *Proceedings of IEEE INFOCOM '00*, pages 603 – 612, Tel Aviv, Israel, March 2000.

[4] M. Andrews. Hardness of buy-at-bulk network design. In *Proceedings of the 45th Annual Symposium on Foundations of Computer Science*, pages 115 – 124, Rome, Italy, October 2004.

[5] M. Andrews. Instability of FIFO in session-oriented networks. *Journal of Algorithms*, 50(2):232–245, 2004.

[6] M. Andrews. Instability of the proportional fair scheduling algorithm for HDR. *IEEE Transactions on Wireless Communications*, 3(5), 2004.

[7] M. Andrews. Maximizing profit in overloaded networks. In *Proceedings of IEEE INFOCOM '05*, Miami, FL, March 2005.

[8] M. Andrews. Instability of FIFO in the permanent sessions model at arbitrarily small network loads. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2007.

[9] M. Andrews. Load balancing in the internet with strict delay constraints. In *Proceedings of IEEE INFOCOM '07*, Anchorage, Alaska, May 2007.

[10] M. Andrews, B. Awerbuch, A. Fernández, J. Kleinberg, T. Leighton, and Z. Liu. Universal stability results and performance bounds for greedy contention-resolution protocols. *Journal of the ACM*, 48(1):39–69, January 2001.

[11] M. Andrews, J. Cao, and J. McGowan. Measuring human satisfaction in data networks. In *Proceedings of IEEE INFOCOM '06*, Barcelona, Spain, April 2006.

[12] M. Andrews, J. Chuzhoy, S. Khanna, and L. Zhang. Hardness of edge-disjoint paths with congestion on undirected graphs. In *Proceedings of the 46th Annual Symposium on Foundations of Computer Science*, Pittsburgh, PA, October 2005.

[13] M. Andrews, A. Fernández, A. Goel, and L. Zhang. Source routing and scheduling in packet networks. In *Proceedings of the 42nd Annual Symposium on Foundations of Computer Science*, pages 168–177, Las Vegas, NV, October 2001.

[14] M. Andrews, A. Fernández, M. Harchol-Balter, T. Leighton, and L. Zhang. General dynamic routing with per-packet delay guarantees of O(distance + 1/session rate). *SIAM Journal on Computing*, 30(5):1594–1623, 2000.

[15] M. Andrews, K. Jung, and A. Stolyar. Stability of the max-weight routing and scheduling protocol in dynamic networks and at critical loads. *Submitted*, 2006.

[16] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting. Scheduling in a queueing system with asynchronously varying service rates. *Probability in the Engineering and Informational Sciences*, 18, 2004.

[17] M. Andrews and K. Munagala. Online algorithms for caching multimedia streams. In *Proceedings of the 8th European Symposium on Algorithms*, 2000.

[18] M. Andrews, L. Qian, and A. Stolyar. Optimal utility based multi-user throughput allocation subject to throughput constraints. In *Proceedings of IEEE INFOCOM '05*, 2005.

[19] M. Andrews, B. Shepherd, A. Srinivasan, P. Winkler, and F. Zane. Clustering and server selection using passive monitoring. In *Proceedings of IEEE INFOCOM '02*, New York, NY, June 2002.

[20] M. Andrews and A. Slivkins. Oscillations with TCP-like flow control in networks of queues. In *Proceedings of IEEE INFOCOM '06*, 2006.

[21] M. Andrews and L. Zhang. Minimizing end-to-end delay in high-speed networks with a simple coordinated schedule. In *Proceedings of IEEE INFOCOM '99*, New York, NY, 1999.

[22] M. Andrews and L. Zhang. Packet routing with arbitrary end-to-end delay requirements. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pages 557–565, Atlanta, GA, 1999.

[23] M. Andrews and L. Zhang. Achieving stability in networks of input-queued switches. In *Proceedings of IEEE INFOCOM '01*, Anchorage, Alaska, April 2001.

[24] M. Andrews and L. Zhang. Scheduling over a time-varying user-dependent channel with applications to high speed wireless data. In *Proceedings of the 43nd Annual Symposium on Foundations of Computer Science*, Vancouver, Canada, November 2002.

[25] M. Andrews and L. Zhang. Routing and scheduling in multihop wireless networks with time-varying channels. In *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms*, New Orleans, LA, January 2004.

[26] M. Andrews and L. Zhang. Scheduling over non-stationary wireless channels with finite rate sets. In *Proceedings of IEEE INFOCOM '04*, Hong Kong, March 2004.

[27] M. Andrews and L. Zhang. Bounds on fiber minimization in optical networks with fixed fiber capacity. In *Proceedings of IEEE INFOCOM '05*, Miami, FL, March 2005.

[28] M. Andrews and L. Zhang. A CDMA data measurement and analysis tool. In *Proceedings of the 2nd International Workshop on Wireless Network Measurement (WiNMee)*, 2005.

[29] M. Andrews and L. Zhang. Hardness of the undirected congestion minimization problem. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, Baltimore, MD, May 2005.

[30] M. Andrews and L. Zhang. Hardness of the undirected edge-disjoint path problem. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, Baltimore, MD, May 2005.

[31] M. Andrews and L. Zhang. Complexity of wavelength assignment in optical network optimization. In *Proceedings of IEEE INFOCOM '06*, Barcelona, Spain, April 2006.

[32] M. Andrews and L. Zhang. Logarithmic hardness of the directed congestion minimization problem. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 517–526, Seattle, WA, 2006.

[33] E. Anshelevich, D. Kempe, and J. Kleinberg. Stability of load balancing algorithms in dynamic adversarial systems. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, Montreal, Canada, May 2002.

[34] B. Awerbuch and Y. Azar. Buy-at-bulk network design. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science*, pages 542 – 547, Miami Beach, FL, October 1997.

[35] B. Awerbuch and T. Leighton. Improved approximation algorithms for the multi-commodity flow problem and local competitive routing in dynamic networks. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, pages 487–496, 1994.

[36] Y. Bartal. On approximating arbitrary metrics by tree metrics. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 161 – 168, Dallas, TX, May 1998.

[37] A. Baveja and A. Srinivasan. Approximation algorithms for disjoint paths and related routing and packing problems. *Mathematics of Operations Research*, 25:255–280, 2000.

[38] J.-Y. Le Boudec and P. Thiran. *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*. Springer, 2001.

[39] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *Proceedings of the 35th Annual ACM Symposium of Theory of Computing*, pages 448 – 455, San Diego, CA, 2003.

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

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

[42] S. Low, L. Peterson, and L. Wang. Understanding Vegas: a duality model. *Journal of the ACM*, 49(2):207–235, 2002.

[43] P. Raghavan and C.D. Thompson. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7:365 – 374, 1991.

[44] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 37(12):1936 – 1948, December 1992.

[45] L. Tassiulas and A. Ephremides. Dynamic server allocation to parallel queues with randomly varying connectivity. *IEEE Transactions on Information Theory*, 30:466 – 478, 1993.

[46] D. Tse. Multiuser diversity in wireless networks. http://www.eecs.berkeley.edu/˜dtse/stanford416.ps.