On Comparing the Performance of Dynamic Multi-Network Optimizations

G.J. Hoekstra[1,2], R.D. van der Mei[2,3] and J.W. Bosman[2]

[1]Innovation Research & Technology, Thales Nederland B.V., Huizen, The Netherlands
[2]CWI, Probability and Stochastic Networks, Amsterdam, The Netherlands
[3]VU University Amsterdam, Department of Mathematics, The Netherlands

***Abstract*—With a large variety of wireless access technologies available, multi-homed devices may strongly improve the performance and reliability of communication when using multiple networks simultaneously. A key question for the practical application of multi-path strategies is the granularity at which the traffic streams should be dispersed among the available networks. This level of granularity may be expected to have a major impact on both the efficiency and complexity of practical realizations. Motivated by this, we compare two dynamic strategies that operate at different levels of granularity. The first strategy, which we call network selection, requires little operational complexity and dynamically *assigns* an arriving application data transfer to the network that delivers the highest expected performance. Our second strategy, which we call traffic-splitting, is of higher complexity and aims to optimally *split* individual data transfers among the available networks. To this end, we (1) develop quantitative models that describe the performance of both strategies, (2) determine the (near-)optimal algorithms for both strategies, and (3) validate the efficiency and practical usefulness of the algorithms via extensive network simulations and experiments in a real-life testbed environment. These experimental results show that the optimal strategies obtained from the theoretical models lead to extremely well-performing solutions in practical circumstances. Moreover, the results show that the splitting of data transfers, which is easy to embed in the network requiring no information on the number of flows in the system, leads to a much better performance compared to dynamic network selection.**

***Index Terms*—Multi-homed systems, capacity aggregation, Processor Sharing queues, file splitting**

I. Introduction

TODAY'S wireless Internet users are accustomed to the ever increasing data rates offered by the latest generation wireless access technologies. Many contemporary wireless networks have already closely approached the Shannon limit on channel capacity, leaving complex signal processing techniques room for only modest improvements in the data transmission rate [1]. The research efforts on the physical layer of wireless systems have matured this field, such that the theoretical limits are known and are closely approached by practical systems. The data rates available to the applications on mobile multi-homed devices may strongly benefit from the overlapping coverage of a wide range of wireless access technologies that operate in different frequency bands and already achieve very high spectral efficiencies. Using these networks in parallel creates a huge potential for performance and reliability improvements.

Over the years, many approaches have been proposed on aggregating capacity on multi-homed devices in wireless networks. As opposed to the research on the physical layer of wireless systems, the application performance in multi-network environments is not well understood with respect to its fundamental limits and how to approach these by practical systems.

Motivated by this, in the present paper we study both the theoretical modeling and the practical viability of the following two performance improvement strategies for file transfers over multiple wireless access networks: (1) a *dynamic network selection* strategy that assigns an entire file transfer to just one of the available networks, based on the number of ongoing transfers in each network, and (2) a *traffic-split* strategy that dynamically splits the file contents into segments each of which is assigned to one of the networks on the basis of TCP-level information (e.g. measured throughput and round trip times) from the connections.

In the literature, research efforts on aggregating capacity on mobile multi-homed devices multiple networks concentrate primarily on the use of SCTP (see for example [2], [3], [4]) or TCP. It needs to be pointed out that the functionality for efficiently using multiple network paths is not considered by the SCTP standard [5], meaning that distributing and re-sequencing the data should be implemented separately. In addition, SCTP applies the same flow- and congestion-control mechanism for the possibly different networks used in parallel, which is not in the interest of overall efficient link utilization nor application performance. Others have proposed to modify TCP [6], [7], [8] to aggregate capacity on multi-homed devices. In another area of research, the application performance of file transfers is often modeled by Processor Sharing (PS)-based models [9], [10], [11] that have shown to be applicable to a wide variety of wireless access networks, including CDMA 1xEV-DO, Wireless LAN (WLAN), and UMTS-HSDPA. In fact, PS models may accurately predict the performance of file transfers over WLANs [12] by taking into account the complex dynamics of the application and its underlying protocol-stack. In a queuing-theoretical context, the distribution and re-assembly of tasks into subtasks are typically modeled by fork-join constructions [13]. In cases where the processing times of the subtasks are independent, exact or numerical analysis is relatively simple (e.g., [14]), whereas the inclusion of dependent processing times (e.g., due to queuing or job splitting) typically leads to very complex analysis (e.g., [15], [16]). For PS-based nodes that process the tasks of a job in parallel, the complex correlation structure between the sojourn times at the PS nodes makes an exact detailed mathematical analysis of the model impossible. As a result, the available literature on queuing models with regard to traffic-splitting is not widely adopted and hence leaves a gap between theory and practice.

In this paper, we study two strategies that aggregate the capacity at a wireless, multi-homed device to achieve near-optimal performance for file transfers. The network selection strategy dynamically assigns an arriving transfer to the network that delivers the best expected performance, whereas the traffic-splitting strategy actually splits the file transfer into segments and distributes those among the available networks. For these two strategies we evaluate their efficiency and practical usefulness in a mutual comparison.

PS models form an attractive class of models that on the one hand accurately describe the resource sharing-behavior among the flows in a network, and on the other hand often allow for an exact mathematical analysis. Therefore, we use PS models to describe the performance of our strategies, to subsequently determine (near-) optimal algorithms, and validate the efficiency and ease-of-use of these algorithms via extensive network simulations and experiments in a real-life testbed. The results show that the traffic-splitting strategy leads to a much better performance compared to dynamic network selection and is easier to embed in multi-homed devices operating in wireless networking environments.

The novelty of our study is threefold. First, in the literature today there is no satisfactory quantitative model that accurately describes the traffic-splitting performance under practical circumstances for multi-homed devices. The model presented here fills the gap between theory and practice. Second, a comparison of traffic-splitting against a practically embedded Markov Decision Process (MDP)-based dynamic network selection scheme is new. Third, we show through practical experiments that extremely high efficiencies and network performance can be obtained in a testbed environment using real networks.

II. Model

In this paper, we consider the multi-network environment in Figure 1, where several wireless access devices may use a number of networks. We analyze the flow-level performance of this system and model each of the networks by PS nodes. The model consists of N parallel PS nodes that represent wireless access networks. There are $N+1$ traffic streams: a single stream of foreground jobs (called class-0 jobs) and N streams of background jobs (called class-i jobs, for $i = 1, \ldots, N$). Class-i jobs arrive according to independent Poisson processes with rates λ_i, the service times are exponentially distributed with mean $\beta_i = 1/\mu_i$, and the corresponding load offered to the system is $\rho_i = \lambda_i \beta_i$, $i = 0, 1, \ldots, N$. Throughout this

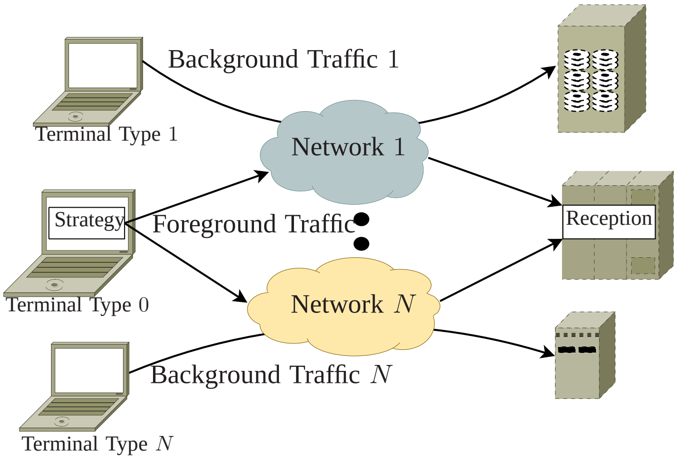

Fig. 1: Multi-network environment.

paper the terms jobs, flows and transfers are used interchangeably.

The decision logic to decide upon the optimum traffic distribution of foreground streams depends on the chosen strategy. We consider the following two strategies. In the *dynamic network selection strategy* foreground jobs are assigned to one network, based on the number of foreground and background flows in each of the networks. In the *dynamic traffic-splitting strategy*, jobs are dynamically split into N portions, each of which is assigned to one of the N networks, based on statistics obtained from the TCP connections in each of the networks.

A. Dynamic Network-Selection Strategy

The network selection strategy is modeled by a Markov Decision Process (MDP) and allows the formulation of an optimum solution. It is assumed that the reader is familiar with the basic concepts of MDPs, see [17] for a standard text book on MDPs. The state space of this model is $S = {\mathbb{N}_0}^{2N}$, where in $s = (b_1, f_1, b_2, f_2, \ldots, b_N, f_N) \in \mathcal{S}$. Here f_i denotes the number of foreground transfers in network i, and b_i denotes the number of foreground transfers in network i. For each arriving foreground flow, a network has to be selected for transmission. This selection is modeled in the action space $a \in \mathcal{A} = \{1, 2, \ldots, N\}$, representing the index of the PS node that should be selected. The expected download time can be derived from the expected number of flows using Little's formula $\lambda_0 \mathbb{E}[S] = \mathbb{E}[L]$ with λ_0 the (foreground traffic) arrival rate, $\mathbb{E}[S]$ the expected download time, and $\mathbb{E}[L]$ the steady state expected number of foreground transfers.

Therefore, in the MDP formulation the reward is chosen equal to the number of foreground transfers (i.e. $\sum_{i=1}^{N} f_i$), so that the total expected reward, denoted g (to be obtained from (1) below), corresponds to the expected number of foreground flows in the system. As the average number of flows should be minimized, the value iteration will become a minimization problem. This description can be converted into an MDP. Using value iteration, the backward recursion equations can be defined as follows for $s \in \mathcal{S}$, $t = 0, 1, \cdots$ and $\tau = \frac{1}{\gamma}$:

$$V_{(t+1)\tau}(s) = \frac{\sum_{i=1}^{N} f_i}{\gamma} + \min_{a\in\mathcal{A}} \left[\frac{\lambda_0}{\gamma} V_{t\tau}(s+e_{2a})\right] \quad (1)$$
$$+ \sum_{i\in\mathcal{A}} \left[\frac{\lambda_i}{\gamma} V_{t\tau}(s+e_{2i-1})\right]$$
$$+ \sum_{i\in\mathcal{A}} \left[d_{b,i}\frac{\mu_i}{\gamma} V_{t\tau}\left([s-e_{2i-1}]^+\right)\right]$$
$$+ \sum_{i\in\mathcal{A}} \left[d_{f,i}\frac{\mu_0}{\gamma} V_{t\tau}\left([s-e_{2i}]^+\right)\right]$$
$$+ \sum_{i\in\mathcal{A}} \left[(1-d_{b,i})\frac{\mu_i}{\gamma} + (1-d_{f,i})\frac{\mu_0}{\gamma}\right] V_{t\tau}(s),$$

with

$$d_{b,i} = \begin{cases} \dfrac{b_i}{b_i+f_i} & \text{if} \quad b_i+f_i>0 \\ 0 & \text{otherwise} \end{cases},$$

$$d_{f,i} = \begin{cases} \dfrac{f_i}{b_i+f_i} & \text{if} \quad b_i+f_i>0 \\ 0 & \text{otherwise} \end{cases},$$

and

$$\gamma = \lambda_0 + \sum_{i\in\mathcal{A}} [\lambda_i + (\mu_i+\mu_0)].$$

In (1), the terms correspond to the number of foreground transfers in state s, the event of a foreground transfer arrival transition, a background arrival transition, a background transfer completion, a foreground completion and a dummy transition respectively. Furthermore, e_{2i-1} is the unit vector that has zeros on all dimensions except on the dimension that corresponds to the number of background transfers on network i, e_{2i} does the same for foreground traffic. This vector is used in the equations for identifying the transitions of the Markov chain. The fractions $\frac{b_i}{b_i+f_i}$ and $\frac{f_i}{b_i+f_i}$ are a result of the fact that the networks are modeled as PS nodes. Each transfer will receive a fraction $\frac{1}{b_i+f_i}$ of service (the number of transfers in a network equals the sum of background and foreground). From this, the total fraction of service to transfers on node i will be $\frac{b_i}{b_i+f_i}$ for background traffic and $\frac{f_i}{b_i+f_i}$ for foreground traffic. As the total expected reward g corresponds to the expected number of foreground transfers, Little's formula can now be applied in order to obtain the expected sojourn time of the foreground traffic stream: $\mathbb{E}[S_0] = \frac{\mathbb{E}[L_f]}{\lambda_0} = \frac{g}{\lambda_0}$, where S_0 represents the foreground download time, L_f is the number of foreground transfers, λ_0 is the foreground job arrival rate, and g is the long-term average expected reward (in this case the expected number of foreground flows).

B. Dynamic Traffic-Split Strategy

In contrast to the network selection strategy, where *entire* files are scheduled for transmission over one specific network, the traffic-splitting strategy actually splits a file into N parts, based on the total number of foreground and background flows in each of the PS nodes. The i-th part is transferred to PS node i. At the receiving end, the N parts are reassembled, which concludes the transfer of a file. The objective is to minimize the expected transfer-and-reassembly time of the files from the foreground stream.

We assume that the splitting strategy dynamically splits the file with infinitely small granularity into parts that are assigned to the nodes, such that all nodes finish servicing the tasks belonging to the same job exactly at the same time. We also assume that the splitting strategy operates optimally in the sense there is no synchronization time needed at the receiving end to reassemble the file, which is a realistic assumption to model the performance in TCP-based networks. This assumption allows us to formulate the model as a continuous time Markov chain for with an $N+1$-dimensional state space, $\tilde{S} = \mathbb{N}_0^{N+1}$, where a state $\tilde{s} \in \tilde{S}$ is of the form $\tilde{s} = (f, b_1, \ldots, b_N) \in \tilde{S}$ with f the number of foreground flows in the system (which is the same for all N PS nodes) and b_i is the number of background flows in PS node i. The transition rates of this Markov chain are then as follows:

$$q(\tilde{s}, \tilde{s}+\tilde{e}_i) = \lambda_i \quad (i=0,\ldots,N), \tag{2}$$

$$q(\tilde{s}, \tilde{s}-\tilde{e}_0) = \sum_{i=1}^{N} \frac{f}{d_i}\mu_0, \tag{3}$$

$$q(\tilde{s}, \tilde{s}-\tilde{e}_i) = \frac{b_i}{d_i}\mu_i \quad (i=1,\ldots,N), \tag{4}$$

with

$$d_i = \begin{cases} f+b_i & \text{if} \quad f+b_i>0, \\ 1 & \text{if} \quad \text{otherwise.} \end{cases}$$

Here, $\tilde{e}_i$ represents the unit vector that has zeros on all dimensions except the dimension that corresponds to the total number of foreground jobs (by taking $i=0$) or to the number of background jobs (for $i=1,\ldots,N$). Based on this result we denote the expected number of foreground transfers in our multi-network environment as $\mathbb{E}[N_0] = \mathbb{E}\left[\tilde{e}_0^T \pi(.)\right]$, where $\tilde{e}_0^T$ represents the transposed unit vector of $\tilde{e}_0$, and $\pi(.)$ stands for the marginal distribution of our Markov chain. Using Little's formula we obtain the expected download time of the foreground traffic.

III. Experimental setting

We have implemented the dynamic network-selection strategy in an OPNET network simulation and the traffic-split strategy in a testbed environment to assess the effectiveness of both strategies in a multi-network environment of two networks, $N=2$. Both strategies are mutually compared against their theoretical model outcomes to assess their practical efficiency. The simulations and testbed are configured in accordance to the conditions in [12], where it is validated that the file download performance in WLAN can be modeled by a PS model under specific circumstances that were also adopted for our experiments in this paper. Due to space constraints we refer to [12] for more details. In our experiments we have used the following parameterization:

$$\rho = \frac{\rho_0+\rho_1+\rho_2}{2}, \quad \kappa = \frac{1-\rho_1}{1-\rho_2}, \quad \eta = \frac{\rho_0}{\rho_0+\rho_1+\rho_2},$$

where ρ is the average of all network loads, κ is the ratio of mean unutilized capacity on both networks and η is the ratio of foreground load to total traffic load offered to the system.

A. Dynamic Network-Selection Strategy

Using (1), an optimal assignment strategy for our network selection problem can be obtained, and used in a network simulation as the policy that should be enforced. Our approach is based on the PS-model presented in [12] that was used to obtain the decision policies for a number of parameter and load combinations and later to parameterize the simulation scenarios and the theoretical model. Our simulated network contains two WLAN APs, operating on non-overlapping frequency channels, that serve the download requests arriving from three types of terminal types. There are ten multi-homed terminals of type 0 generating with arrival rate λ_0 the foreground traffic within close (35 meter) range of both available networks and may use them simultaneously accordingly. In addition there are ten single-homed terminals of type 1 at an equal distance of fifteen meters from AP1 that generate the background traffic with rate λ_1 on the first network. The remaining ten single-homed terminals generate the background traffic at rate λ_2 on AP2 in a similar fashion. We have parameterized all WLAN stations such, that for a mean file size of 200kByte we obtain an expected download time of one such transfer in an empty network of $\beta = 0.36$ seconds.

B. Dynamic Traffic-Split Strategy

To evaluate the performance of the dynamic traffic-splitting strategy an application was developed that uses the standard sockets API to distribute its FTP application traffic among the multiple networks present in accordance to the Arrival-Time matching Load-Balancing (ATLB) method from [8]. In this method the time of arrival of TCP segments from different networks is matched at the receiver's resequencing point, and takes into account the two most important delay factors; the queuing delay at the sender and the transmission delay in the network. The first factor can be obtained by maintaining the TCP throughput of each connection over time, whereas the latter factor the Round Trip Time (RTT) estimation from TCP (smoothed RTT) can be used. The effectiveness of our traffic distribution application is subsequently evaluated in a testbed consisting of two powerful multi-homed PCs that are connected by two independent and identically configured WLAN access networks. Similar to the foreground traffic, the background traffic in each network consists of independent Poisson arrivals of i.i.d. files with a mean size of 2MByte. As our traffic-splitting solution requires some 'warm-up' time to operate effectively, the files that are transferred consist of a fixed portion of 1MByte that is added on top of an exponentially distributed file with a mean size of 1MByte. The WLAN equipment in our testbed was configured such that we obtain an expected download time for our files in an empty network of $\beta = 3.03$ seconds. Note that the resulting file size distribution does not entirely match the exponential assumptions in our theoretical model but further experiments revealed near-insensitivity of the splitting performance to the file size distribution, which is in line with similar observations that were made in [12] for single network scenarios.

IV. Experimental Results

In this section we compare the model, simulation and experimental outcomes of both strategies and have included a third strategy, the so-called *static selection model*, in which the network is selected with the lowest background load to represent a commonly used approach. We have conducted many experiments and the results are outlined below (for 95% Confidence Intervals of less than two percent).

Figure 2 plots the mean download times for $\beta = 1$, obtained from the models and the experiments (measured download times are divided by β), of the foreground traffic as a function of ρ for the different network selection and traffic-splitting strategies discussed above.

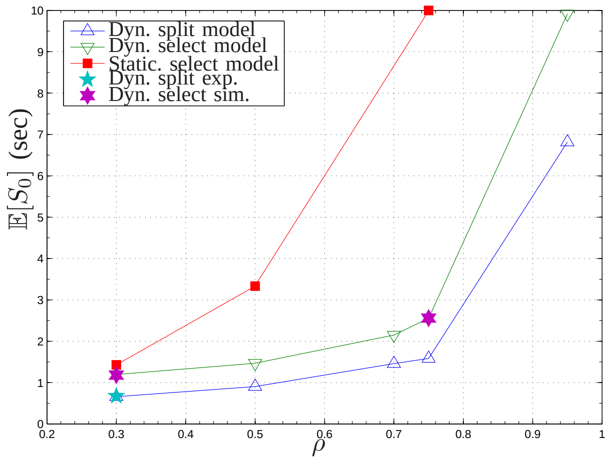

Fig. 2: Mean foreground download time, $\mathbb{E}[S_0]$, as a function of ρ, for $\kappa = 2/3$.

The results lead to a number of observations. First, the dynamic traffic-splitting model consistently outperforms the dynamic network selection strategy, for all values of the load with performance improvements typically in the range of $30 - 40\%$. Second, the static selection strategy is strongly outperformed by both dynamic strategies, as expected.

Figures 3 and 4 show the results (again normalized to $\beta = 1$) from the experiments, simulations and those from the analytic models. In these figures, the mean download time of the foreground traffic is plotted as a function of the asymmetry factor κ.

These results lead to a number of observations. First, the dynamic strategies deliver significantly different performance. The dynamic traffic-split model consistently outperforms the dynamic network selection strategy for all values of κ. Second, the experimental results obtained from our testbed match extremely well with the dynamic traffic-splitting model, which implies that traffic-splitting can be achieved under practical circumstances with nearly-optimal performance (differences between one and three percent). Third, the simulation outcomes of the dynamic network selection strategy also match very closely to the quantitative model. Fourth, again the

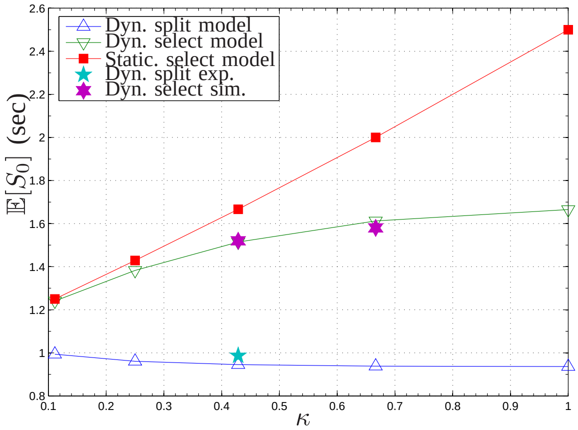

Fig. 3: Mean foreground download time, $\mathbb{E}[S_0]$, as a function of κ for a constant total system load, $\rho = 0.55$ and $\eta = 0.09$.

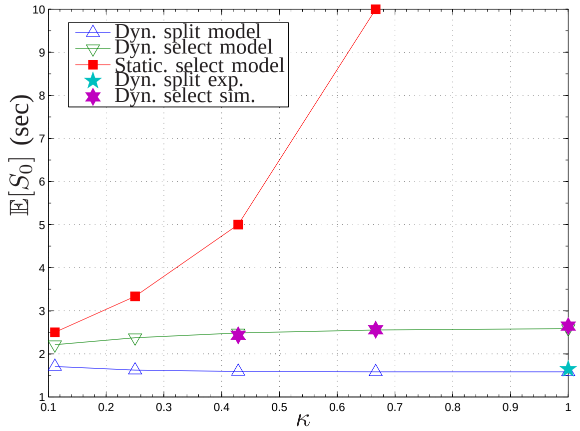

Fig. 4: Mean foreground download time, $\mathbb{E}[S_0]$, as a function of κ for a constant total system load, $\rho = 0.75$ and $\eta = 0.33$.

static network selection strategy far less effective than the dynamic strategies, particularly if the network loads are fairly symmetric (i.e. κ close to 1). This effect is vanishes for higher asymmetry in the load values (i.e. κ close to 0.1).

Based on the experimental results it can be concluded that the dynamic traffic-splitting strategy consistently delivers better performance in terms of lower mean download response times when compared to dynamic network selection. Best performance (twice as fast) is achieved under low load conditions and under the least favorable conditions of having extreme load asymmetries (i.e. $\kappa = 0$) and low foreground traffic loads (i.e. $\eta = 0.1$) mean download times may still be reduced by more than eighteen percent. This high performance comes at a cost that primarily lies in the implementation complexity (see [6]), but need not necessarily imply that this functionality should be adopted in a TCP-stack replacement [6], [7] or even in a separate device [8]. In contrast to the proposed dynamic network selection strategy, there is no need for any additional network status information, other than what may be calculated from the TCP connection's statistics. Clearly, relatively short file transfers may benefit from dynamic server selection, as there is no 'warm-up' period required. However, embedding dynamic network selection strategies in network devices involves no sophisticated splitting functionality but does require that there is a front-end router/dispatcher that has maintains the information on the number of foreground and background flows in the network. Provided that this is the case, the traffic load values should be known or estimated using for example queue-learning techniques.

V. Acknowledgments

The work reported in this paper was supported by the Netherlands Organisation for Scientific Research (NWO) under the Casimir project: Analysis of Distribution Strategies for Concurrent Access in Wireless Communication Networks.

References

[1] D. Cox, "Fundamental limitations on the data rate in wireless systems," *IEEE Communications Magazine*, vol. 46, no. 12, pp. 16–17, 2008.

[2] J. Iyengar, P. Amer, and R. Stewart, "Concurrent multipath transfer using sctp multihoming over independent end-to-end paths," *IEEE/ACM Trans. Netw.*, vol. 14, no. 5, pp. 951–964, 2006.

[3] J. Iyengar, "End-to-end concurrent multipath transfer using transport layer multihoming," Ph.D. dissertation, Newark, DE, USA, 2006, adviser-Paul D. Amer.

[4] C. Huang and C. Tsai, "The handover control mechanism for multi-path transmission using stream control transmission protocol (sctp)," *Computer Communications*, vol. 30, no. 17, pp. 3239–3256, 2007.

[5] R. Stewart, "Stream control transmission protocol," Internet Engineering Task Force, RFC 2960, October 2000.

[6] H. Hsieh and R. Sivakumar, "A transport layer approach for achieving aggregate bandwidths on multi-homed mobile hosts," in *MobiCom '02: Proceedings of the 8th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2002, pp. 83–94.

[7] M. Zhang, J. Lai, A. Krishnamurthy, L. Peterson, and R. Wang, "A transport layer approach for improving end-to-end performance and robustness using redundant paths," in *ATEC '04: Proceedings of the annual conference on USENIX Annual Technical Conference*. Berkeley, CA, USA: USENIX Association, 2004, pp. 8–8.

[8] Y. Hasegawa, I. Yamaguchi, T. Hama, H. Shimonishi, and T. Murase, "Deployable multipath communication scheme with sufficient performance data distribution method," *Computer Communications*, vol. 30, no. 17, pp. 3285–3292, 2007.

[9] S. Borst, O. Boxma, and N. Hegde, "Sojourn times in finite-capacity processor-sharing queues," in *Proceedings NGI 2005 Conference*, 2005.

[10] R. Litjens, F. Roijers, J. V. den Berg, R. Boucherie, and M. Fleuren, "Performance analysis of wireless LANs: An integrated packet/flow level approach," in *Proceedings of the 18th International Teletraffic Congress - ITC18*, Berlin, Germany, 2003, pp. 931–940.

[11] Y. Wu, C. Williamson, and J. Luo, "On processor sharing and its applications to cellular data network provisioning," *Performance Evaluation*, vol. 64, no. 9-12, pp. 892–908, 2007.

[12] G. Hoekstra and R. van der Mei, "On the Processor Sharing of File Transfers in Wireless LANs," in *Proceedings of the 69th IEEE Vehicular Technology Conference, VTC Spring 2009*, Barcelona, Spain, 26-29 April 2009.

[13] S. Ko and R. Serfozo, "Response times in M/M/s fork-join networks," *Advances in Applied Probability*, vol. 36, no. 3, pp. 854–871, 2004.

[14] G. Choudhury and D. Houck, "Combined queuing and activity network based modeling of sojourn time distributions in distributed telecommunication systems," in *the 14th International Teletraffic Congress - ITC14*, J. Labetoulle and J.W.Roberts, Eds., Antibes, France, 1994, pp. 525–534.

[15] L. Flatto and S. Hahn, "Two parallel queues created by arrivals with two demands i," *SIAM Journal on Applied Mathematics*, vol. 44, no. 5, pp. 1041–1053, 1984. [Online]. Available: http://link.aip.org/link/?SMM/44/1041/1

[16] J. Lui, R. Muntz, and D. Towsley, "Computing performance bounds for fork-join queueing models," The Chinese University of Hong Kong, Tech. Rep., 1994.

[17] M. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 1994.