Understanding Slowdown in Large-Scale Heterogeneous Systems

William Turchetta[1] and Kristen Gardner[1]

Amherst College, Amherst MA 01002, USA
williamturchetta@gmail.com, kgardner@amherst.edu

Abstract. Modern computer systems are both large-scale, consisting of hundreds or thousands of servers, and heterogeneous, meaning that not all servers have the same speed. In such systems, slowdown—the ratio of a job's response time to its size—is an important performance metric that has not yet received significant attention. We propose a new definition of slowdown that is well-suited to large-scale, heterogeneous systems. We analyze mean slowdown and mean response time under the Probabilistic SITA family of dispatching policies, and use our analysis to present a numerical study of the tradeoff between these two performance metrics.

Keywords: Dispatching · Heterogeneity · Slowdown.

1 Introduction

Today's computer systems typically consist of hundreds or thousands of servers that are *heterogeneous* in the sense that not all servers have the same speed. Deciding how to dispatch jobs to servers is a question of critical importance in achieving good performance in such systems; policies designed for homogeneous systems often perform poorly in the presence of heterogeneity [1]. Here "performance" typically refers to the size of the stability region or to mean response time; most of the work on large-scale heterogeneous systems focuses on the design, analysis, and evaluation of dispatching policies with respect to these metrics [1, 6].

In this paper we consider an alternative performance metric: mean *slowdown*, where the slowdown of a job captures the ratio between the job's response time and its size. Slowdown is a well-studied metric in single-server systems; the Shortest-Processing-Time-Product (SPTP, also referred to as RS) scheduling policy is known to minimize mean slowdown [5, 7], while Processor Sharing (PS) and Preemptive Last-Come First-Served (PLCFS) have the desirable property that a job's expected slowdown is independent of its size [3]. However, there is relatively little work studying slowdown in heterogeneous systems, and this work tends to focus on systems with only two or three servers [5].

We propose a new definition of slowdown for large-scale heterogeneous systems. We introduce a generalization of the Size Interval Task Assignment (SITA)

policy [4], which we call Probabilistic SITA (PSITA). We analyze the mean system slowdown and mean response time under PSITA and use our analysis to identify optimal parameterizations of PSITA. Finally, using our analytical results, we gain insight about the structure of optimal PSITA policies and the relationship between mean response time and mean slowdown.

2 Model and Preliminaries

We consider a system with k servers that are heterogeneous in that different servers may have different speeds. We assume that there are s server speed classes and let $\mathcal{S} \equiv \{1, \ldots, s\}$ denote the set of all server classes. We denote the number of servers in class i as k_i and the fraction of servers in class i as $q_i = k_i/k$, $i \in \mathcal{S}$. Class-i servers have speed μ_i, where $\mu_1 > \cdots > \mu_s$; all servers operate under the Processor Sharing (PS) scheduling discipline. We assume that each job has a size X drawn from a general distribution. Following [2], the time that a job with size X spends in service on a server with speed μ_i is X/μ_i. Arrivals to the system follow a Poisson process with rate λk. Without loss of generality, we assume that the total capacity of the system is k, i.e., we require $\sum_{i=1}^{s} q_i \mu_i = 1$. To ensure stability, we assume that $\lambda < 1$.

Our primary metric of interest is the mean system *slowdown*. In a homogeneous system, i.e., one in which all servers have the same speed, the slowdown of a job with size x is $S(x) \equiv \frac{T(x)}{x}$, where $T(x)$ is the job's response time. Notably, this definition of slowdown implies that $S(x) \geq 1$, and hence captures the extent to which the job is "slowed down" by the presence of other jobs, relative to the response time it would experience running in isolation. Extending this definition to heterogeneous systems presents a challenge: how should one account for the differences in server speeds? One approach, used in [5], defines the slowdown of a job with size x running on a class-i server (i.e., a server with speed μ_i) as $S(x) \equiv \frac{T(x)}{x/\sum_j \mu_j}$. That is, the slowdown is normalized relative to the overall capacity of the system. The disadvantage of this approach is that, as the number of servers in the system grows, slowdowns can become extremely large, making it difficult to draw meaningful comparisons between different system configurations.

To overcome this disadvantage, we propose a new definition of slowdown in heterogeneous systems.

Definition 1. *The* ***slowdown*** *of a job with size x running on a class-i server is $S_i(x) \equiv \frac{T_i(x)}{x}$.*

At first glance, this definition looks identical to the definition of slowdown in a single-server (or homogeneous) system. However, we note that the term x in the denominator does *not* include the server's speed: this is the job's "inherent size," and not the time it spends in service. Hence, using this definition of slowdown, a job that runs in isolation on a class-i server will experience slowdown $1/\mu_i$, and in general a job's slowdown is scaled by a factor of $1/\mu_i$ relative to its slowdown on

a speed-1 server (i.e., when a job's service time is equal to its size). For example, in an $M/G/1/\text{PS}$ system in which the server has speed 1, the mean slowdown is $\mathbb{E}[S] = \mathbb{E}[S|X] = \frac{1}{1-\rho_i}$ [3]. Using our heterogeneity-based definition, the mean slowdown at a class-i server under PS is $\mathbb{E}[S] = \frac{1}{1-\rho_i} \cdot \frac{1}{\mu_i}$, where in both cases ρ_i denotes the fraction of time that a class-i server is busy.

There are several key advantages to our definition of slowdown. First, our definition is easily interpretable: if a job experiences a slowdown less than 1, we know that the job ran on a fast server and did not compete with many other jobs. Meanwhile, if a job experiences a slowdown greater than 1, we know that it ran on a slow server or on a heavily-loaded server. This clean interpretation is in contrast to the definition used in [5], under which a job that runs in isolation on a fast server may nonetheless experience a very high slowdown. Second, our definition does not explicitly depend on the total system capacity, meaning that, unlike in [5], adding servers to the system will not artificially inflate slowdown values. Finally, our definition facilitates comparisons between the slowdowns obtained under different system configurations, given a fixed total system capacity.

With the goal of achieving low mean slowdown in mind, we propose a new dispatching policy called Probabilistic SITA (PSITA) that generalizes the SITA [4] policy. SITA is typically defined for *homogeneous* systems (i.e., systems in which all servers have the same speed). Under SITA, jobs are divided into k bins based on their sizes; recall that k is the number of servers in the system. One server is allocated to each bin; when a job arrives, it is dispatched to the server corresponding to its bin. In this way, small jobs are isolated from big jobs, thereby protecting the small jobs from experiencing high response times and high slowdowns.

The PSITA policy generalizes SITA by (i) allowing the number of bins to differ from the number of servers, and (ii) making dispatching decisions probabilistically, rather than deterministically. In particular, under PSITA, jobs are divided into b bins based on their sizes. When a job from bin j arrives, the job is dispatched to one of the server classes, where this class is chosen *probabilistically*. These two extensions render PSITA appropriate for use in heterogeneous systems.

Definition 2. *Let $0 \equiv x_1 \leq x_2 \leq \cdots \leq x_b < x_{b+1} \equiv \infty$. A job is in bin j if it has size x such that $x_j \leq x < x_{j+1}$, $j \in \{1, \ldots, b\} \equiv \mathcal{B}$. Under the **Probabilistic SITA (PSITA)** dispatching policy, when a job from bin j arrives it is sent to some class-i server with probability $p_{j,i}$, for all $i \in \mathcal{S}$. The specific server is chosen uniformly at random among all class-i servers.*

3 Analysis

In this section we derive mean slowdown, $\mathbb{E}[S]$, and mean response time, $\mathbb{E}[T]$, under PSITA dispatching and PS scheduling for fixed values of x_j, $j \in \mathcal{B}$, and $p_{j,i}$, $j \in \mathcal{B}$, $i \in \mathcal{S}$.

We begin by observing that

$$\mathbb{E}[S] = \sum_{i=1}^{s} p_i \mathbb{E}[S_i]$$
$$\mathbb{E}[T] = \sum_{i=1}^{s} p_i \mathbb{E}[T_i],$$

where $\mathbb{E}[S_i]$ and $\mathbb{E}[T_i]$ are respectively the mean slowdown and the mean response time on a class-i server, and p_i is the probability that an arbitrary job is dispatched to a class-i server. Note that the per-class mean slowdowns are weighted by the probability that an arbitrary *job* runs on a class-i server, not the probability that a *server* is of class i.

To find p_i, we will condition on the job's size, X; recall that, under PSITA, jobs in size bin j are dispatched to a class-i server with probability $p_{j,i}$. We find:

$$p_i = \sum_{j=1}^{b} p_{j,i} \mathbb{P}(X \in [x_j, x_{j+1})). \tag{1}$$

The PSITA dispatching policy amounts to Poisson splitting of the jobs within each bin, hence each class-i server behaves like an $M/G_i/1$. The total arrival rate to class-i servers is $\lambda k p_i$; because we dispatch uniformly within a server class, the arrival rate to an individual class-i server is

$$\lambda_i = \frac{\lambda k p_i}{k_i} = \frac{\lambda p_i}{q_i}. \tag{2}$$

The service time at a class-i server is distributed as $Y_i \sim X_i/\mu_i$. Here X_i denotes the size distribution of jobs that are dispatched to class-i servers. In an $M/G_i/1/\text{PS}$ with arrival rate λ_i and service times Y_i, the mean slowdown and mean response time are given by:

$$\mathbb{E}[S_i] = \frac{1}{(1-\rho_i)} \frac{1}{\mu_i}$$
$$\mathbb{E}[T_i] = \frac{\lambda_i \mathbb{E}[Y_i^2]}{2(1-\rho_i)} = \frac{\lambda_i \mathbb{E}[X_i^2]}{2(1-\rho_i)} \frac{1}{\mu_i^2}.$$

We now proceed to find ρ_i, $\mathbb{E}[X_i]$, and $\mathbb{E}[X_i^2]$. The load at each class-i server is

$$\rho_i = \lambda_i \mathbb{E}[Y_i] = \lambda_i \frac{\mathbb{E}[X_i]}{\mu_i}. \tag{3}$$

We find the expected job size on a class-i server by conditioning on the job size bin:

$$\mathbb{E}[X_i] = \sum_{j=1}^{b} \frac{p_{j,i} \mathbb{P}(X \in [x_j, x_{j+1}))}{p_i} \mathbb{E}[X | X \in [x_j, x_{j+1})], \tag{4}$$

where the term $\frac{p_{j,i}\mathbb{P}(X\in[x_j,x_{j+1}))}{p_i}$ gives the fraction of jobs at class-i servers that are from bin j. Similarly, we have:

$$\mathbb{E}[X_i^2] = \sum_{j=1}^{b} \frac{p_{j,i}\mathbb{P}(X \in [x_j, x_{j+1}))}{p_i} \mathbb{E}[X^2 | X \in [x_j, x_{j+1})]. \tag{5}$$

At this point we have derived all of the components necessary to obtain the following results.

Proposition 1. *The mean slowdown and mean response time under PSITA dispatching and PS scheduling are:*

$$\mathbb{E}[S] = \sum_{i=1}^{s} p_i \frac{1}{(1-\rho_i)} \frac{1}{\mu_i}$$
$$\mathbb{E}[T] = \sum_{i=1}^{s} p_i \frac{\lambda_i \mathbb{E}[X_i^2]}{2(1-\rho_i)} \frac{1}{\mu_i^2}$$

where p_i, λ_i, ρ_i, $\mathbb{E}[X_i]$, and $\mathbb{E}[X_i^2]$ are given in (1)-(5).

Using our closed-form expressions for $\mathbb{E}[S]$ and $\mathbb{E}[T]$ given in Proposition 1, we can now find the PSITA policy that minimizes each metric by jointly optimizing over x_j, $j \in \mathcal{B}$, and $p_{j,i}$, $j \in \mathcal{B}$, $i \in \mathcal{S}$. Our optimization problem for mean slowdown is:

$$\min_{\substack{p_{j,i}, j\in\mathcal{B}, i\in\mathcal{S} \\ x_j, j\in\mathcal{B}}} \sum_{i=1}^{s} p_i \frac{1}{(1-\rho_i)} \frac{1}{\mu_i}$$

$$\begin{aligned}
\text{s.t.} \quad & p_i = \sum_{j=1}^{b} p_{j,i}\mathbb{P}(X \in [x_j, x_{j+1})) && \forall i \in \mathcal{S} \\
& \rho_i = \frac{\lambda p_i \mathbb{E}[X_i]}{q_i \mu_i} && \forall i \in \mathcal{S} \\
& \mathbb{E}[X_i] = \sum_{j=1}^{b} \frac{p_{j,i}\mathbb{P}(X \in [x_j, x_{j+1}))}{p_i} \cdot \mathbb{E}[X | X \in [x_j, x_{j+1})] && \forall i \in \mathcal{S} \\
& \sum_{i=1}^{s} p_{j,i} = 1 && \forall j \in \mathcal{B} \\
& \sum_{i=1}^{s} p_i = 1 \\
& 0 \le p_{j,i}, p_i \le 1 && \forall j \in \mathcal{B},\ i \in \mathcal{S} \\
& 0 \le \rho_i < 1 && \forall j \in \mathcal{B}
\end{aligned}$$

The optimization problem for mean response time is similar. While we opt to take $b = |\mathcal{B}|$ (the number of job size bins) as fixed, note that one could also leave b as a parameter of the optimization problem.

λ	$p^*_{2,1}$	$p^*_{1,1}$	x^*_2	$\mathbb{E}[S]$	ρ_1	ρ_2
0.1	0	1	3.96	0.88	0.14	0.03
0.2	0	1	2.76	1.02	0.23	0.14
0.3	0	1	2.4	1.18	0.31	0.28
0.4	0	1	2.28	1.38	0.4	0.4
0.5	0	1	2.2	1.65	0.48	0.53
0.6	0	1	2.2	2.05	0.58	0.64
0.7	0	1	2.2	2.72	0.68	0.74
0.8	0	1	2.24	4.06	0.79	0.83
0.9	0.01	1	2.24	8.07	0.89	0.92

Table 1. Optimal policy parameters and corresponding performance metrics when $q_1 = q_2 = 0.5$ and $r = 2$.

4 Results and Discussion

In this section, we present a brief numerical study of systems with $s = 2$ server speeds and $b = 2$ job size bins; we assume that $X \sim \text{Exp}(1)$. We let $r \equiv \frac{\mu_1}{\mu_2}$ denote the speed ratio between fast and slow servers.

Both the objective function and the constraints of the optimization problem given in Section 3 are non-convex in multiple dimensions, hence standard optimization algorithms are not guaranteed to yield globally optimal solutions. We thus carry out our optimization using a grid search, considering all 4 080 501 combinations of $x_2 \in \{0, 0.04, \ldots, 3.96, 4\}$ and $p_{1,1}, p_{2,1} \in \{0, 0.005, \ldots, 0.995, 1\}$. We then select the parameters $(x^*_2, p^*_{1,1}, p^*_{2,1})$ that yield the lowest mean slowdown and refer to this as the optimal solution for the given system settings.

4.1 Structure of Optimal PSITA Policies

In this section, we study the structure of PSITA policies that minimize mean slowdown. Table 1 shows the optimal policy parameters ($p^*_{1,1}$, $p^*_{2,1}$, and x^*_2) and corresponding performance metrics ($\mathbb{E}[S]$, ρ_1, and ρ_2) in a system in which $q_1 = q_2 = 0.5$, $r = 2$, and λ varies. (While Table 1 shows results only for one specific pair of q_1 and r values, we observed similar results for other parameter settings.) Interestingly, the optimal PSITA policies are in fact nearly always SITA policies. That is, the optimal policy nearly always has $p^*_{1,1} = 1$ and $p^*_{2,1} = 0$, meaning that all small jobs are dispatched to fast servers and all large jobs are dispatched to slow servers. This indicates that the primary factor that leads to low slowdown is the isolation of small jobs. Indeed, at all but the highest values of λ, the only policy parameter that changes with λ is the size cutoff above which jobs are considered large: as λ increases the optimal size cutoff decreases. This is because at higher values of λ, isolating the very smallest jobs requires us to offload more moderately-sized jobs to the slow servers. When λ is very high, the optimal policies change structurally in two ways. First, the optimal cutoff increases slightly. Second, the optimal PSITA policy is no longer a SITA

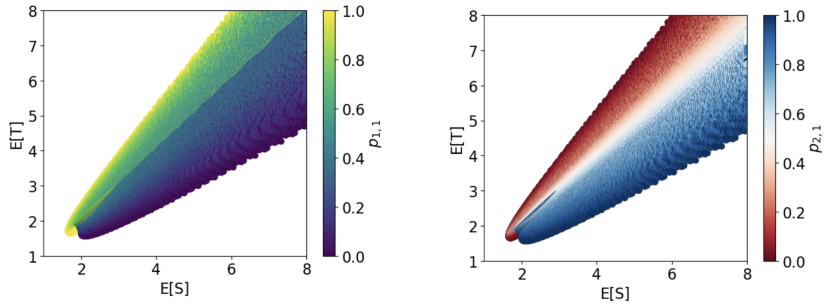


Fig. 1. Mean response time vs. mean slowdown for $\lambda = 0.5$, $q_1 = q_2 = 0.5$, $\mu_1 = \frac{4}{3}$, and $\mu_2 = \frac{2}{3}$, for all values of $x_2 \in \{0, 0.04, \ldots, 3.96, 4\}$ and $p_{1,1}, p_{2,1} \in \{0, 0.005, \ldots, 0.995, 1\}$. Colors represent the value of $p_{1,1}$ (left) and $p_{2,1}$ (right).

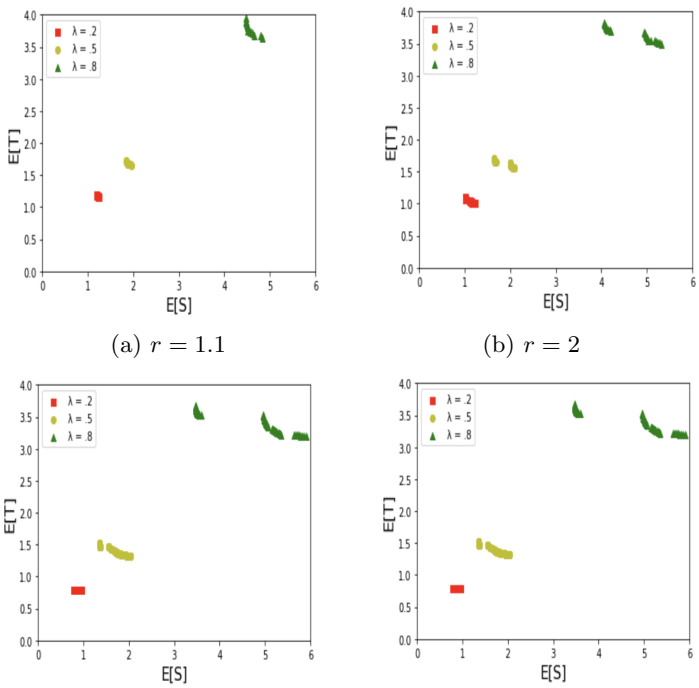
policy. Together, these structural changes ensure that the slow servers remain stable. This highlights the additional flexibility that PSITA offers over SITA: under SITA, it would only be possible to maintain stability at the slow servers by increasing the size cutoff between small and large jobs.

4.2 Tradeoff between $\mathbb{E}[S]$ and $\mathbb{E}[T]$

We now turn to the relationship between mean slowdown and mean response time. Figure 1 shows $\mathbb{E}[T]$ as a function of $\mathbb{E}[S]$ in a system where $q_1 = q_2 = 0.5$, $r = 2$, and $\lambda = 0.5$. Each point represents a different parameterization of the PSITA policy; we show all parameterizations considered in the grid search, giving a total of 4 080 501 points. In general, there is a roughly linear relationship between mean response time and mean slowdown. Both metrics depend strongly on the per-class loads, so a policy parameterization that leads to a high load at one or both server classes likely yields poor performance for both metrics.

Despite the correlation between $\mathbb{E}[S]$ and $\mathbb{E}[T]$, it is clear that the optimal policies for slowdown are not necessarily optimal for mean response time. Two distinct regions emerge in Figure 1: a "cone" with slightly lower $\mathbb{E}[S]$ (shown by yellow points in the left subfigure and by red points in the right subfigure) consisting of policies that send most of the small jobs to fast servers and most of the big jobs to slow servers, and a "cone" with slightly higher $\mathbb{E}[S]$ (shown by blue points in both subfigures) consisting of policies that do the opposite. Broadly, policies in the former cone obtain similar mean response times to their counterparts in the latter cone, but lower mean slowdowns. This pattern emphasizes the importance not only of isolating the small jobs, but of isolating the small jobs *on fast servers* in order to achieve low mean slowdown.

Ultimately, in a system where both metrics are of equal importance, one would want to select a policy along the Pareto-dominating envelope of the region shown in Figure 1. Figure 2 shows the tradeoff between $\mathbb{E}[S]$ and $\mathbb{E}[T]$ for these dominating policies, for three values of λ and four values of r. That is, each point



(a) $r = 1.1$ (b) $r = 2$

(c) $r = 5$ (d) $r = 10$

Fig. 2. Mean response time vs. mean slowdown for three values of λ and four values of r, where $q_1 = q_2 = 0.5$. Each point represents a non-dominated policy, i.e., a policy for which it is not possible to simultaneously reduce both $\mathbb{E}[S]$ and $\mathbb{E}[T]$.

represents a policy for which it is not possible to simultaneously reduce both $\mathbb{E}[S]$ and $\mathbb{E}[T]$. When λ is low, the Pareto-optimal policies are tightly clustered: the policies that are optimal with respect to $\mathbb{E}[T]$ are also nearly optimal with respect to $\mathbb{E}[S]$, and vice versa. As λ increases the tradeoff between $\mathbb{E}[S]$ and $\mathbb{E}[T]$ becomes more pronounced: there emerges a set of distinct Pareto-optimal policies. All of these policies behave similarly in the sense that all operate by isolating small jobs. However, the policies differ in *where* they isolate the small jobs. As previously observed, slowdown-optimal policies send small jobs to fast servers and large jobs to slow servers; meanwhile, the policies that minimize mean response time tend to do the opposite. While different policies are optimal for each metric, we note that $\mathbb{E}[T]$ is significantly less sensitive than $\mathbb{E}[S]$ to the choice of (Pareto-optimal) policy.

5 Conclusion

Our work represents an important first step towards understanding slowdown in large-scale, heterogeneous systems. Our key contribution is our definition of

slowdown, which is interpretable, scalable, and allows for meaningful comparisons between system configurations. We analyze mean slowdown and mean response time under the PSITA dispatching policy and PS scheduling, and present a numerical study of the tradeoff between these two performance metrics. Our results indicate that, while mean slowdown and mean response time are generally correlated, minimizing one metric does not necessarily result in minimizing the other. In particular, slowdown is far more sensitive than response time to the particular choice of policy parameters, suggesting that when both metrics are of equal importance, it may be more valuable to select a policy aimed at minimizing slowdown.

While our numerical study considers only systems with two server speeds and exponentially distributed job sizes, and we allow our PSITA policies to include only two job size bins, we expect that the insights learned from this study are likely to translate to more general settings. In particular, we anticipate that isolating small jobs at fast servers will remain the most important factor in achieving low mean slowdown. Furthermore, preliminary results suggest that there are diminishing marginal returns when increasing the number of jobs size bins beyond two.

In this paper we focus on identifying optimal dispatching policies within the PSITA family; however these policies likely are not optimal more generally. For example, policies that yield lower mean response times, such as JSQ-d, JIQ, and the policies studied in [1], will likely also yield lower mean slowdowns; analyzing slowdown under such policies represents an important direction for future work.

References

1. J. Abdul Jaleel, S. Doroudi, K. Gardner, and A. Wickeham. A general "power-of-d" dispatching framework for heterogeneous systems. *Queueing Systems*, 2022.
2. K. Gardner, M. Harchol-Balter, A. Scheller-Wolf, and B. Van Houdt. A better model for job redundancy: Decoupling server slowdown and job size. *IEEE/ACM transactions on networking*, 25(6):3353–3367, 2017.
3. M. Harchol-Balter. *Performance modeling and design of computer systems: queueing theory in action*. Cambridge University Press, 2013.
4. M. Harchol-Balter, M. E. Crovella, and C. D. Murta. On choosing a task assignment policy for a distributed server system. *Journal of Parallel and Distributed Computing*, 59(2):204–228, 1999.
5. E. Hyytiä, S. Aalto, and A. Penttinen. Minimizing slowdown in heterogeneous size-aware dispatching systems. *ACM SIGMETRICS Performance Evaluation Review*, 40(1):29–40, 2012.
6. A. Stolyar. Pull-based load distribution in large-scale heterogeneous service systems. *Queueing Systems*, 80(4):341–361, 2015.
7. S. Yang and G. De Veciana. Size-based adaptive bandwidth allocation: Optimizing the average QoS for elastic flows. In *Proceedings. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 657–666. IEEE, 2002.