

Scheduling Issues in Multi-Channel Wireless Networks

Mingwei Gong
Department of Computer Science
University of Calgary
Calgary, AB, Canada T2N 1N4
Email: mgong@ucalgary.ca

Carey Williamson
Department of Computer Science
University of Calgary
Calgary, AB, Canada T2N 1N4
Email: carey@cpsc.ucalgary.ca

Abstract—Scheduling decisions can have a pronounced impact on the performance of multi-radio wireless systems. In this paper, we study the effects of dispatch policies and queue scheduling strategies on the user-perceived performance for Internet traffic flows in a multi-channel WLAN. Our work is carried out using simulation and an empirical Web workload trace, with mean response time as the primary performance metric. The simulation results demonstrate the good/bad combination of the dispatch policy with queue scheduling strategy, the advantages of deferred dispatch over immediate dispatch, and the sensitivity of dispatch policies to heavy-tailed workload characteristics. The results also highlight the pros and cons of a simple lookahead scheduling policy, particularly in the presence of high variability workloads on a heterogeneous multi-channel system with random losses. Our results provide insights into efficient and robust scheduling policies for multi-channel WLANs.

Keywords: Multi-channel wireless networks, Dispatch policy, Scheduling, Simulation

I. INTRODUCTION

Wireless LANs based on the IEEE 802.11 standard are prevalent today in corporate and campus environments as a flexible and cost-effective means to provide users with untethered access to the Internet. The IEEE 802.11a/b/g specifications provide a multi-rate capability at the physical layer to accommodate mobile users with diverse wireless channel conditions. For example, IEEE 802.11b allows transmissions at 1, 2, 5.5, or 11 Mbps [12], while IEEE 802.11a supports 8 rate options up to 54 Mbps. Furthermore, the IEEE 802.11 PHY specification permits simultaneous operation on multiple non-overlapping channels.

A recent trend in commercial wireless networks is toward *multi-radio* systems [4]. That is, wireless access points and wireless mesh routing nodes have *multiple* wireless radios, each operating independently (and concurrently) on different wireless channels. Even end-user devices can have multiple radios, enabling dynamic selection between (for example) WiFi, WiMAX, Bluetooth, and GSM as a preferred wireless access technology. We refer to such systems generically as *multi-channel wireless networks* in this paper.

The multi-radio and multi-rate features of multi-channel systems pose new challenges for flow-level and packet-level scheduling [17]. If used properly, these new features can improve system throughput, reduce mean response time, and ef-

fectively support both throughput-oriented and delay-sensitive applications.

Our work focuses on flow-level scheduling in multi-channel wireless systems. Scheduling has long been a popular means to provide quality of service (QoS) support [10]. However, previous wireless scheduling policies typically assume a single, fixed transmission rate for all users. As a result, channel scheduling algorithms designed for a single-rate environment may not be appropriate in multi-channel WLANs.

In this paper, we use simulation to evaluate how dispatch policies and queue scheduling policies affect system performance in multi-channel WLANs. The dispatch policies can be categorized into *immediate dispatch* and *deferred dispatch*, depending on whether the dispatcher has a central queue or not. The simulator considers both job-level scheduling issues as well as wireless channel characteristics, such as heterogeneous channel rates and wireless losses.

Our paper makes three main contributions. First, extensive simulations are conducted for multi-channel wireless systems with different channel assignment (dispatch) policies and queue scheduling policies. Second, a non-work-conserving Lookahead policy is evaluated for the multi-channel systems. Finally, different effects on the multi-channel systems are analyzed, including effect of job size distribution, effect of capacity ratio, and effect of wireless channel loss. Our simulation results demonstrate the (expected) superiority of deferred dispatch over immediate dispatch, as well as the benefits of preemptive size-based scheduling with heavy-tailed job size distributions, and the pros and cons of a simple Lookahead policy.

The rest of the paper is organized as follows. Section II presents background material and related work. Section III discusses simulation methodology, including assumptions, network configuration, experimental design, and performance metrics. Section IV presents our simulation results for immediate dispatch policies, while Section V presents results for deferred dispatch policies. Finally, Section VI concludes the paper.

II. BACKGROUND AND RELATED WORK

This section presents the background material required to understand the design of dispatch policies in a multi-channel

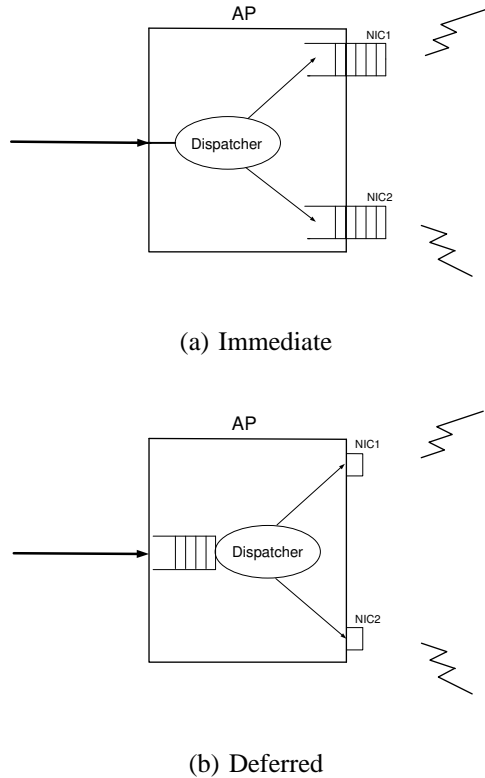


Fig. 1. Dispatch Policies (2 channels)

wireless system. The dispatch policies decide to which channel an incoming job should be sent for service, and when. Due to the conceptual similarity between channel assignment policies in WLANs and dispatch policies in multi-server systems, we begin with a brief overview of these systems.

A. Multi-Server Systems

A dispatcher in a server farm receives all incoming jobs and sends each job to one of the servers for processing. This type of scenario includes grid computing systems, call centers, and Web servers.

There are two general categories of dispatch policies in multi-server systems, depending on whether there is a central dispatch queue or not. Figure 1(a) illustrates immediate dispatch. Incoming jobs are forwarded immediately to a chosen server (channel) for processing, based on a dispatch decision-making criterion. Scheduling policies are used at each queue to decide the service order for pending jobs. Figure 1(b) illustrates deferred dispatch. The main difference is that there is a central queue at the dispatcher, and the dispatch decision is only made when a server (channel) is available. Jobs are indivisible. Furthermore, once a job is assigned, it cannot be reassigned.

Immediate dispatch policies have been studied extensively in server farm scenarios. The most common ones are *Random* and *Round-Robin* [20]. The Random policy assigns jobs to each server equiprobably at random, while the Round-Robin

policy assigns jobs to servers in a cyclical fashion. Both policies aim to balance the expected number of jobs at each server, and are frequently used as a baseline to compare with other job dispatch policies. Previous studies show that Random and Round-Robin have similar performance. We thus use Random as our baseline policy in the paper.

One of the best immediate dispatch policies in the literature is Size Interval Task Assignment (SITA). SITA with Variable Load (SITA-V) [7] defines a (static) size threshold, such that small jobs are dispatched to the least-loaded server, while larger jobs are sent to the busier server. SITA with Equal Load (SITA-E) [13] assigns distinct (static) ranges of job sizes to each server so that the total expected load for each server is the same. Upon arrival, each job is assigned to the server that handles jobs in that size range. This policy isolates small jobs from large jobs.

Some immediate dispatch policies consider the current load at each server, in order to distribute jobs among servers more judiciously. It is known that balancing the load minimizes the mean response time [7], [19]. A Least Loaded First (LLF) approach assigns a job to the server with the least work remaining. The remaining work can be interpreted with two possible metrics: the number of jobs remaining (Fewest Jobs), or the total bytes remaining (Fewest Bytes).

B. Multi-Channel Wireless Systems

Recently, multi-channel wireless networks have been the subject of extensive research [2], [4], [8], [14], [22]. Previous work for multi-channel networks can be broadly categorized into single-radio and multi-radio approaches. Typically, multi-radio approaches achieve a higher throughput than single-radio approaches, though they are more expensive to implement.

The single-radio approaches assume that the radio can be rapidly tuned from one channel to another, on a per-packet basis [5], [21]. However, this approach has several drawbacks. It requires each node to tune periodically to a common control channel to communicate. Such coordination requires tight time synchronization among nodes. Mo *et al.* [16] provide a taxonomy of single-radio approaches.

Other research on multi-channel networks assumes a multi-radio architecture. These approaches use inexpensive commodity 802.11 hardware, and can overcome the deficiencies of single-radio solutions. The general goal is to assign channels to radio interfaces to achieve efficient utilization. Adya *et al.* [1] propose a simple approach using common channel assignment (CCA). In their approach, radio interfaces at each node are assigned to the same set of channels. However, inefficient channel utilization can occur if the number of channels exceeds the number of interfaces per node. Rama *et al.* [18] develop a measurement-based approach for channel assignment. A central server periodically collects information about dynamically-varying channel interference. The approach assigns channels in order to minimize interference. However, the centralized architecture makes this approach difficult to implement in practice. Ko *et al.* [15] present a distributed algorithm for channel assignment. Interference is minimized

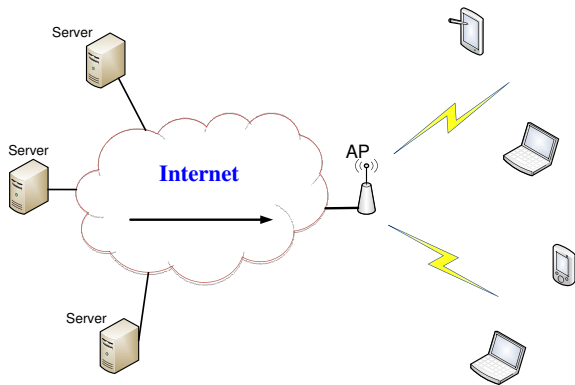


Fig. 2. Wireless Network Scenario

by using partially overlapping channels. However, channel assignment is independent of network load.

Our work applies at the flow level in multi-radio wireless networks, such as wireless mesh networks or WLANs. We explore the performance tradeoffs for a range of channel assignment and queue scheduling strategies, assuming heavy-tailed Internet flow sizes and heterogeneous wireless channels.

III. EXPERIMENTAL METHODOLOGY

Simulation is used to evaluate dispatch policies and their impacts on the multi-channel system. This section describes the design of the simulation experiments.

A. Network and Workload Model

We consider a generic model of a multi-channel wireless network, as indicated in Figure 2. Network traffic flows originate from the Internet on the left. These flows arrive at the wireless Access Point (AP), and must be scheduled for transmission to WLAN users (on the right).

The AP has two channels (slow and fast) on which to schedule these flow transmissions. The fast link is twice as fast as the slow link. Each flow is mapped onto exactly one channel.

The workload model used in our work is an empirical World Cup trace [3]. This trace represents 1 million requests to a sports Web server over a 15-minute interval. The mean transfer size is 3.5 KB, and the coefficient of variation is 5.4. Table I provides a statistical summary of the empirical Web server workloads used in our simulations. This same trace was used in our earlier work on size-based scheduling in Web servers [9], facilitating cross-validation of our simulation results against single-server scenarios.

In all simulations, the system load (utilization) is controlled by setting the channel link capacities. We first determine the link capacity at saturated system load ($\rho = 1$), based on the total bytes transferred during the trace duration.

$$\text{Link capacity} = \frac{\text{Total bytes transferred}}{\text{Trace duration}}$$

TABLE I
SUMMARY OF TRACE CHARACTERISTICS

Item Description	Empirical Trace
Trace Duration	860.9 sec
Total Requests	1,000,000
Unique Web Objects	5,549
Total Transferred Bytes	3.5 GB
Smallest Transfer Size	4
Median Transfer Size	889
Largest Transfer Size	2,891,887
Mean Transfer Size	3,498
Standard Deviation	18,815

System load for $\rho < 1$ is adjusted by scaling up the link capacity accordingly. The capacities of the heterogeneous channels are determined using the following expressions:

$$\text{Slow link} = \frac{1}{\text{Link ratio} + 1} * \text{Link capacity}$$

$$\text{Fast link} = \frac{\text{Link ratio}}{\text{Link ratio} + 1} * \text{Link capacity}$$

The default link capacity ratio is 2.

B. Dispatch Policies

A brief description of the four immediate dispatch policies considered in our work appears in Table II. Note that we generalize SITA to a Size Interval (SI) policy for heterogeneous channels (servers). With heterogeneous channel capacity, the size threshold is defined so that the total workload assigned to the fast channel is proportional to the channel rate. The Random policy also chooses channels in proportion to their rates, instead of equiprobably at random (i.e., the probability of assigning jobs to the fast channel is twice as much as assigning them to the slow channel when the service ratio is two).

TABLE II
IMMEDIATE DISPATCH POLICIES

Random	Send the incoming job to one of the two channels at random.
Size Interval	Send small jobs to the fast channel, and large jobs to the slow channel.
Fewest Jobs	Send incoming job to the channel with the fewest jobs pending.
Fewest Bytes	Send incoming job to the channel with the fewest bytes pending.

Three different deferred dispatch policies are considered: Random, Fastest Channel, and Lookahead. Table III provides a brief description of these three policies.

C. Experimental Design

The simulation experiments use a multi-factor experimental design. The primary factors are the dispatch policy and queue scheduling policy. The system load ranges from 30% to 95% in all of the experiments.

Table IV summarizes the factors and levels used in the trace-driven simulation experiments for immediate dispatch policies. Each new arrival is immediately dispatched to one

TABLE III
DEFERRED DISPATCH POLICIES

Random	If both channels are available, then send the job to a channel at random, else send it to the available channel.
Fastest	If both channels are available, then send the job to the fastest channel, else send it to the available channel.
Lookahead	Calculate expected completion time for each channel. Send the job to the channel with the earliest completion time.

of the channel queues. Four different immediate dispatch policies are modeled in the simulator. The queue scheduling policy at the channel is one of FIFO (First-In-First-Out), PS (Processor-Sharing), SJF (Shortest-Job-First), or SRPT (Shortest-Remaining-Processing-Time). The dispatcher has global knowledge regarding the queue status at each channel (a common assumption in multi-server scenarios). This assumption allows the dispatcher to make correct dispatch decisions (i.e., assigning a job to the queue with fewest jobs or fewest bytes).

TABLE IV
EXPERIMENTAL FACTORS AND LEVELS FOR IMMEDIATE DISPATCH POLICIES

Factor	Levels
System Load	30% to 95%
Dispatch Policy	Random, Size Interval (SI), Fewest Jobs (FJ), Fewest Bytes (FB)
Scheduling	FIFO, PS, SJF, SRPT

Table V summarizes the design of the deferred dispatch experiments. Each new job arrival is stored at the central queue, and is sent to one of the two channels when available. The dispatch policies considered include Random, Fastest channel, and Lookahead. Two queue scheduling policies are used: FIFO and SJF.

The primary performance metric in our simulation experiments is *mean response time*, defined as the elapsed time from when the job arrives to the system until it departs from the system. This metric is used to compare different dispatch policies and queue scheduling policies under different system loads.

IV. RESULTS FOR IMMEDIATE DISPATCH

This section presents the simulation results for immediate dispatch policies in our two-channel WLAN scenario. We

TABLE V
EXPERIMENTAL FACTORS AND LEVELS FOR DEFERRED DISPATCH POLICIES

Factor	Levels
System Load	30% to 95%
Dispatch Policy	Random, Lookahead, Fastest Channel
Scheduling	FIFO, SJF
Channel Ratio	1:1, 2:1, 10:1
Channel Loss Ratio	0%, 1%, 10%

consider separately the effects of dispatch decision and the scheduling policy at the individual transmission queues.

A. Effect of Dispatch Policy

Figure 3 shows the mean response time results for different dispatch policies. The graphs show mean response time as a function of system load, using a logarithmic scale on the vertical axis. Each graph shows results for one particular dispatch policy, with one line plotted for each of the four different queue scheduling policies. In general, the mean response time increases with load for all scheduling policies, as expected. However, the shapes of the curves and their relative placement vary for different policies.

Figure 3(a) shows the results for Random dispatch. As expected, SRPT scheduling shows the lowest mean response time for all system loads among the four scheduling policies compared, while FIFO scheduling shows the worst. These trends are consistent across other dispatch policies as well. PS produces lower response time than SJF. This result shows that preemption plays an important role in scheduling: preemptive policies consistently outperform non-preemptive policies under Random dispatch.

Figure 3(b) shows the results for the (static) Size Interval policy. For system loads lower than 80%, the same trends can be observed as for the Random dispatch policy. All four scheduling policies show a sharp increase in mean response time for system load above 80%. In general, the SI dispatch policy has the advantage of reducing the chance that small jobs wait behind large jobs, since they are dispatched to different queues. However, it assigns an uneven number of jobs to each channel, which affects the overall performance, especially for the PS scheduling policy. PS is inefficient when the number of concurrent jobs grows large.

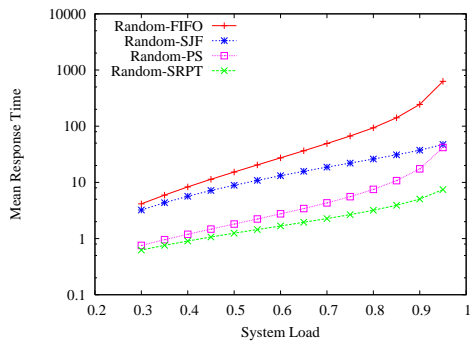
Figure 3(c) shows the results for the Fewest Jobs dispatch policy. Generally, the simulation results follow the trends observed for the Random dispatch policy. That is, the mean response time increases from SRPT to PS to SJF to FIFO under most system loads. However, at high system load, PS has slightly worse performance than the SJF policy.

Figure 3(d) shows the results for the Fewest Bytes policy. Again, SRPT has the best mean response time, while FIFO performs the worst. PS generally exhibits lower mean response time than SJF, except at high system load, when the number of jobs becomes unbalanced, degrading the performance of the PS policy.

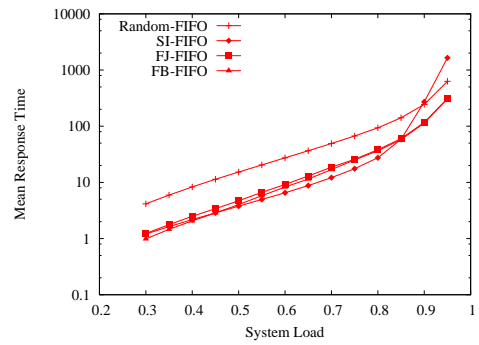
B. Effect of Queue Scheduling Policy

Figure 4 presents a different view of the foregoing results. The queue scheduling policy is fixed for each graph, while each line shows the results for a different immediate dispatch policy.

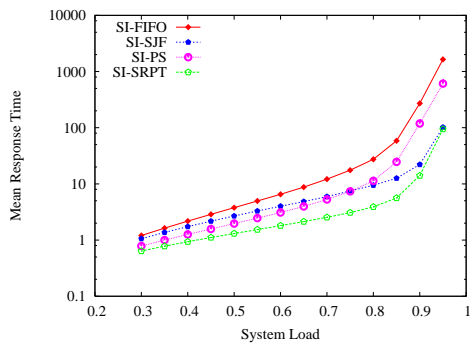
Figure 4(a) shows the simulation results with FIFO queue scheduling. Random dispatch performs poorly in this context, as expected, while the other three policies perform comparably. At low to moderate system load, SI performs slightly better than the other dispatch policies, as indicated in the prior



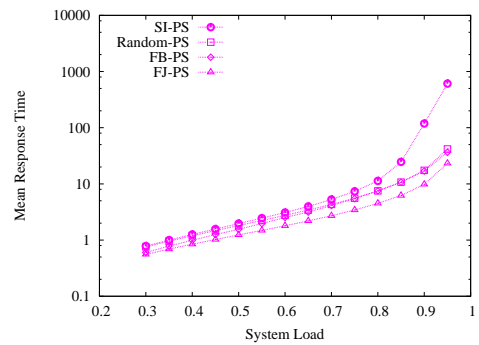
(a) Random



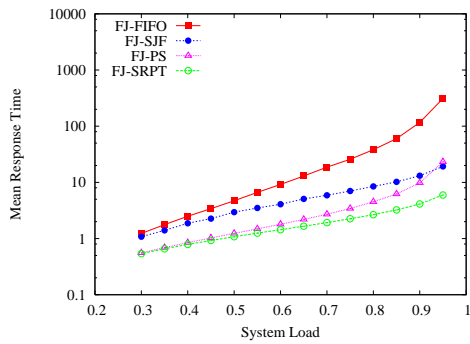
(a) FIFO



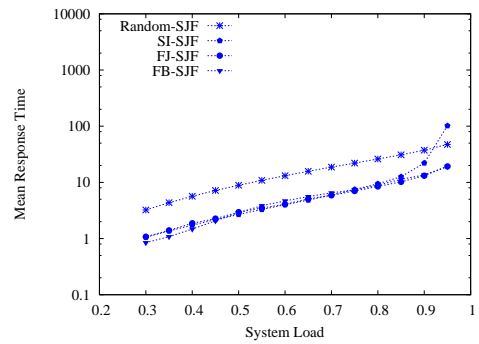
(b) Size Interval



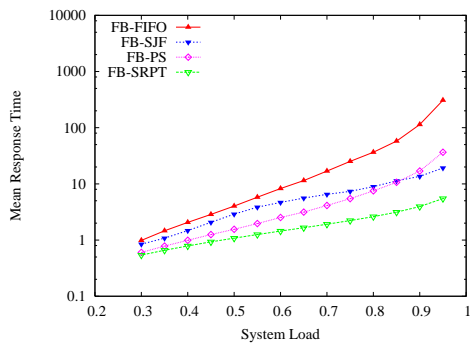
(b) PS



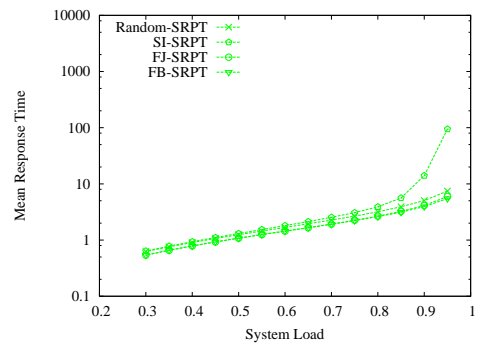
(c) Fewest Jobs



(c) SJF



(d) Fewest Bytes



(d) SRPT

Fig. 3. Simulation Results for Immediate Dispatch Policies with Different Queue Scheduling Policies

Fig. 4. Simulation Results for Different Queue Scheduling Policies with Immediate Dispatch

literature [6], [13]. However, when system load is high, the SI policy performs worse than the other policies. This surprising result for SI has not been reported before. The inconsistent performance of SI can be explained as follows. At low system load, the SI policy can effectively reduce the chance that a small job is stuck behind a large job, by dispatching them to different queues. However, at high system load, many of the jobs are small, and enter the same queue. Forcing many small jobs to compete with each other adversely affects the overall performance.

Figure 4(b) shows results for the PS scheduling policy. Three observations are evident. First, the Fewest Jobs (FJ) dispatch policy performs better than the other three dispatch policies. This is consistent with intuition, as well as previous research [11]. With the FJ policy, the number of jobs in each queue is proportional to the service capacity. Second, the PS scheduling policy softens the adverse effects of Random dispatch, which performs no worse than FB in this example. Third, the drawbacks of the SI policy are even more pronounced. The SI dispatch policy sends most jobs to the fast channel, because of the empirical trace characteristics. The many small jobs in the queue have to share the link using PS, while the relatively few large jobs have their own dedicated channel. This combination of SI dispatch and PS scheduling is the worst choice among all four policies under all system loads.

The remaining graphs show results for the two size-based scheduling policies. Figure 4(c) shows the results for non-preemptive SJF scheduling. Random dispatch is the worst for most system loads considered; the other three load-balancing policies show a distinct advantage. However, the SI policy is slightly worse at very high load, for the reasons stated previously. Figure 4(d) shows results for the preemptive SRPT policy. Preemption improves the performance for all dispatch policies, as noted earlier. The mean response time with the SRPT policy is quite similar for all dispatch policies, except for the SI policy at very high load.

C. Summary

In summary, the simulation results lead to the following observations regarding immediate dispatch:

- Queue scheduling policies have a large impact on the performance of a multi-channel wireless system, as evident by comparing Figure 3 and Figure 4. For example, the SRPT scheduling policy can reduce mean response time by a factor of 2 to 100 compared to other scheduling policies, depending on the dispatch policy and system load.
- Size-based scheduling is advantageous. SJF outperforms FIFO, and SRPT outperforms PS.
- Preemption plays an important role in queue scheduling policies. Preemptive policies (PS and SRPT) typically outperform non-preemptive policies (FIFO and SJF).
- Dispatch policies have only a modest impact on overall system performance, and their effects are only evident at high load. For example, Random dispatch is often poor,

but good scheduling can overcome this. Fewest Bytes and Fewest Jobs show comparable performance for all four queue scheduling policies considered.

- The *combination* of a dispatch policy with the right queue scheduling policy is important. For example, FJ fits well with PS, while SI performs poorly with PS scheduling, especially at high load. While the SI policy isolates small jobs from large jobs, this strategy has a drawback at high load, when many small jobs compete with each other for a shared channel.

V. RESULTS FOR DEFERRED DISPATCH

This section presents the simulation results for deferred dispatch policies. As shown in Figure 1(b), the dispatcher has a central queue that holds incoming jobs before they are sent to one of the two channels. The dispatch decision only applies for the first job waiting in the queue, and dispatch decisions are irreversible. That is, once a job is assigned to a channel, it cannot be interrupted, or switched to another channel.

The presence of heterogeneous channels complicates the dispatch problem. A dispatch decision is needed when a new job arrives to an empty system. In this case, both channels are available, and a decision must be made regarding which one to use. A dispatch decision is also needed whenever a transmission completes, if the central queue is non-empty. In this case, typically one channel is available (either slow or fast).

To maximize the system throughput, an obvious dispatch policy is to assign the job at the front of the queue to the fast channel whenever it is available. We call this policy Fastest Channel. However, it is not immediately clear what to do when only the slow channel is available. Should the job take the slow channel immediately, or wait briefly for the fast channel to be available? Moreover, how do other factors affect the dispatch decision? These questions are addressed in the following subsections.

The default settings for the simulation are as follows. The fast channel capacity is double that of the slow channel. The system load is varied from 30% to 95% for all the simulation results. For the deferred dispatch policies, we consider Random, Fastest, and Lookahead dispatch policies. The Lookahead policy is illustrated in Figure 5. It is a simple heuristic meant to avoid taking the slow channel when the fast channel will be available soon. Only non-preemptive scheduling policies (FIFO and SJF) are considered at the central queue, since pre-empting a traffic flow during transmission is not feasible.

A. Effect of Dispatch Decision

Figure 6 shows the baseline results for the deferred dispatch policies. In general, deferred dispatch is superior to immediate dispatch, often by an order of magnitude or more. This result is evident in Figure 6(a), which compares Random-FIFO (immediate Random dispatch, followed by FIFO queue) with FIFO-Random (FIFO central queue, followed by Random deferred

```

For the job at the head of the central queue do:
  If fast channel is idle
    Then assign job to fast channel
  If slow channel is idle
    Compute finish time  $T_{slow}$  for slow channel
    Compute finish time  $T_{fast}$  for fast channel
    If  $T_{slow} < T_{fast}$ 
      Then assign job to slow channel
    Else wait in the central queue for fast channel

```

Fig. 5. Algorithm for Lookahead Dispatch

dispatch), and Random-SJF (immediate dispatch) with SJF-Random (deferred dispatch).

The trends observed for both FIFO (Figure 6(b)) and SJF (Figure 6(c)) are similar, though clearly SJF produces much lower mean response time than FIFO. The Fastest channel dispatch policy (not shown here, to simplify the graphs) slightly outperforms Random dispatch. When system load is low, the Lookahead dispatch policy performs slightly better than both Random and Fastest dispatch policies. However, when system load is high, the Lookahead policy performs much worse than the other two dispatch policies.

There are two reasons why the simple Lookahead policy is poor, especially at high load. First, it is a *non-work-conserving* policy. As defined in queuing theory, a work-conserving policy never leaves a server idle if there are jobs waiting for service. A non-work-conserving policy relaxes this constraint. In our context, the Lookahead policy may leave the slow channel idle while waiting for the fast channel. This decision wastes some of the inherent capacity of the system. Second, the policy as defined suffers from Head-of-Line (HOL) blocking. That is, the decision made by the front job to wait for the fast channel may block¹ the next job (and subsequent jobs) in the queue from making progress (or possibly different decisions). These effects are most evident at high system load, with a continuous stream of arrivals. Again, system resources are wasted when the slow channel idles. SJF makes this situation even worse, since successive jobs in the queue tend to be comparable in size (making it even less likely that they choose separate channels).

B. Effect of Job Size Distribution

Previous studies have shown that Web workloads exhibit a heavy-tailed transfer size distribution. The heavy-tailed property refers to extremely high variability in job sizes. This is often manifested as many small values mixed with a few extremely large values. In particular, a small fraction of the jobs (e.g., 1%) accounts for most of the byte traffic volume (e.g., 50%).

¹A non-blocking version of the Lookahead policy can ameliorate this problem, by looking deeper into the queue for a job willing to use the slow channel.

The next experiment assesses the impact of the job size distribution on the performance of different dispatch policies. We compare the results for the empirical trace (with a heavy-tailed job size distribution) to the results for a synthetically-generated log-normal job size distribution (with the same mean job size, and the same arrival time sequence as the empirical trace).

Figure 7 shows the simulation results for FIFO (Figure 7(a)) and SJF (Figure 7(b)) queue scheduling policies. In general, the mean response time is lower for the log-normal trace than for the empirical trace. This observation applies for all three dispatch policies. However, at high system load, the Lookahead dispatch policy actually performs *worse* with the log-normal job size distribution. Simply stated, the Lookahead dispatch policy can exploit high job size variability, and dispatch a job to the slow channel to avoid waiting excessively for the fast channel. However, with reduced job size variability, the job at the head of the queue is *more likely* to wait for the fast channel, rather than take the slow channel. This selfish decision can affect other jobs, degrading the overall mean response time.

Similar observations apply for SJF scheduling in Figure 7(b). In fact, SJF-Lookahead has even worse performance than FIFO-Lookahead. Since the SJF policy sorts jobs in the central queue by job size, this increases the chances that the front job decides to wait for the fast channel. Thus, overall system performance suffers.

C. Effect of Channel Capacity Ratio

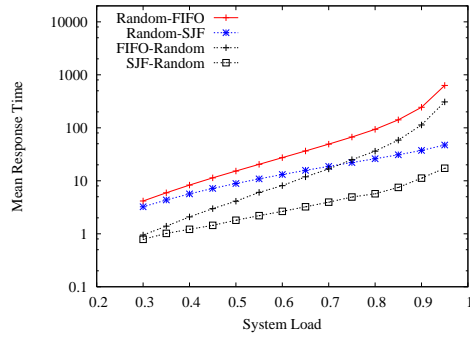
In the foregoing results, the fast channel was twice as fast as the slow channel. Our next experiment evaluates how the channel capacity ratio affects the performance of different channel dispatch policies.

Figure 8(a) presents the simulation results for homogeneous channels. The results separate into two groups, with the upper three lines for FIFO scheduling, and the lower three lines for SJF scheduling. These results are as expected. With homogeneous channels, the dispatch policy makes no difference. System performance is determined by the queue scheduling policy.

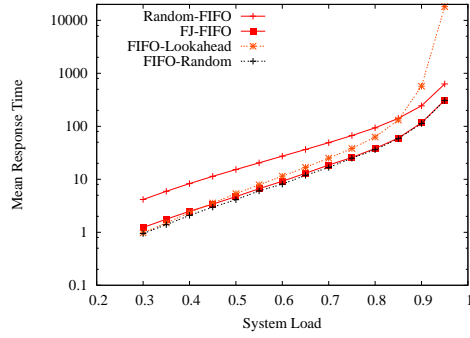
Figure 8(b) shows the results for a capacity ratio of 10. The results again split into two groups (FIFO and SJF), which shows that the queue scheduling policies have greater impact than the dispatch policies. The SJF-Lookahead policy provides the best performance for system loads up to 85%, and is only slightly worse than other policies at higher loads. This scenario is particularly well-suited for the Lookahead policy. The Lookahead heuristic tends to avoid assigning jobs to the slow channel, which is an order of magnitude slower in this case. In this case, leaving the slow channel idle wastes only about 10% of the overall system capacity.

D. Effect of Wireless Channel Loss

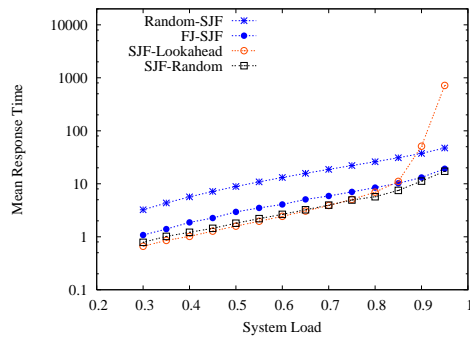
The final experiment is designed to study how wireless channel losses affect system performance. In this experiment, random Bernoulli losses are induced to illustrate the effect of



(a) Random

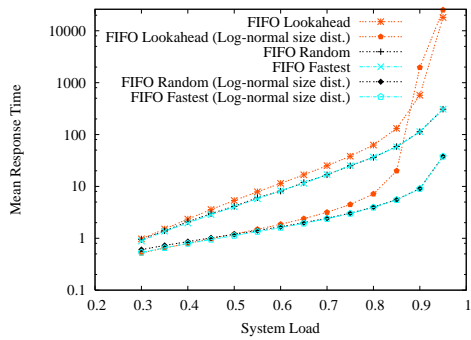


(b) FIFO

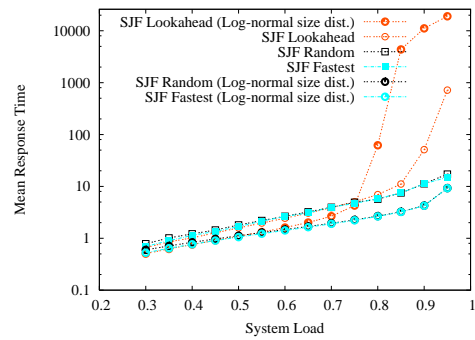


(c) SJF

Fig. 6. Comparison of Immediate and Deferred Dispatch Policies



(a) FIFO



(b) SJF

Fig. 7. Effect of Job Size Distribution

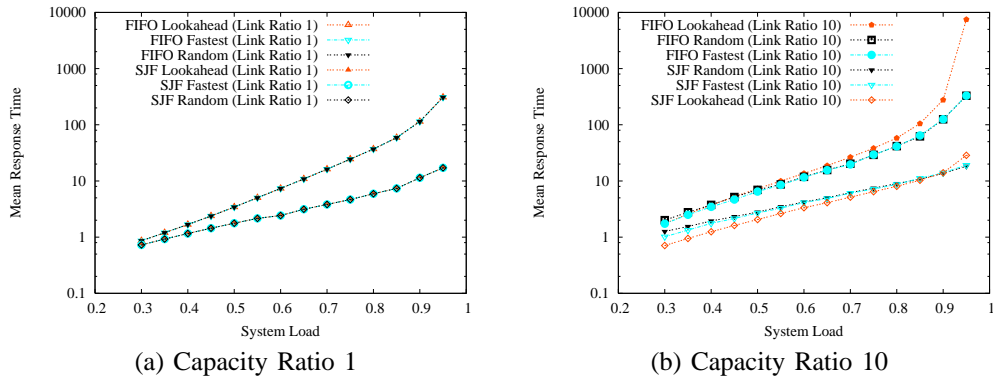


Fig. 8. Effect of Link Capacity Ratio

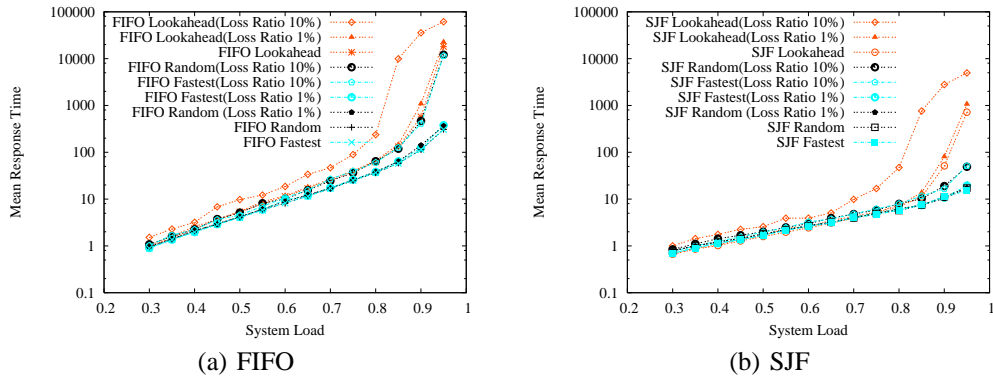


Fig. 9. Effect of Wireless Loss on Fast Channel

wireless channel loss on dispatch policies and queue scheduling strategies. For this set of results, only loss event on the fast channel is illustrated. The loss probability is set to 0%, 1%, or 10% according. Once a loss event happens on the fast channel, the affected job must be retransmitted (perhaps at a lower rate), which increases its effective service time accordingly.

Figure 9 presents the simulation results. Overall, the channel losses have a larger (adverse) impact on Lookahead than on the Random and Fastest dispatch policies. This result is as expected. When a loss event happens under the Lookahead dispatch policy, the retransmitted job is affected, along with those jobs behind it in the queue that are waiting for the fast channel. Uncertainty about job completion times makes Lookahead worse; this effect is evident even at 1% loss.

E. Summary

Four conclusions follow from our simulation results:

- The simple Lookahead dispatch policy performs poorly compared to the other dispatch policies, especially at high system load. Although this heuristic is intuitively appealing, its non-work-conserving property leads to HOL blocking, and sacrifices overall system performance.
- Workload characteristics matter. Size-based scheduling is advantageous for heavy-tailed job sizes, because it can exploit the high variability.
- The channel capacity ratio affects the performance of dispatch policies. The Lookahead policy performs at its best

when the channels differ greatly, since the HOL problem is less pronounced. In other scenarios, its performance is disappointing.

- Wireless channel losses can have a large impact on the performance of dispatch policies. Uncertainty in job completion times reduces the effectiveness of the Lookahead policy.

VI. CONCLUSIONS

This paper presents a detailed simulation study of dispatch policies and scheduling strategies for multi-channel wireless systems. The simulation results show that deferred dispatch policies are generally preferable to immediate dispatch policies, though the queue scheduling policy has far greater impact than the dispatch policy. In particular, the simulation results reveal some good/bad combinations of dispatch policy with queue scheduling policy, such as FJ fits well with PS, while SI performs poorly with PS scheduling. A simple Lookahead dispatch policy is advantageous at low to moderate load in multi-channel systems with vastly different channel capacities, but can be detrimental at high load due to HOL blocking effects. Finally, different effects on the multi-channel system are analyzed, including the effects of job size distribution, link capacity ratio, and wireless channel loss. Deferred dispatch with SJF scheduling at the central queue provides fairly robust performance, regardless of the specific dispatch policy used.

ACKNOWLEDGEMENTS

Financial support for this research was provided by iCORE (Informatics Circle of Research Excellence) in the Province of Alberta, as well as NSERC (Natural Sciences and Engineering Research Council) in Canada. The authors thank the anonymous reviewers for their constructive comments on our work.

REFERENCES

- [1] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou, "A Multi-Radio Unification Protocol for IEEE 802.11 Wireless Networks", *Proceedings of BROADNETS*, pp. 344-354, 2004.
- [2] M. Alicherry, R. Bhatia, and L. Li, "Joint Channel Assignment and Routing for Throughput Optimization in Multi-radio Wireless Mesh Networks", *Proceeding of ACM MobiHoc*, pp. 58-72, 2005.
- [3] M. Arlitt and T. Jin, "A Workload Characterization Study of the 1998 World Cup Web Site", *IEEE Network*, Vol. 14, No. 3, pp. 30-37, May/June 2000.
- [4] P. Bahl, A. Adya, J. Padhye, and A. Wolman, "Reconsidering Wireless Systems with Multiple Radios", *ACM Computer Communication Review*, Vol. 34, No. 5, pp. 39-46, October 2004.
- [5] P. Bahl, R. Chandra, and J. Dunagan, "SSCH: Slotted Seeded Channel Hopping for Capacity Improvement in IEEE 802.11 Ad-hoc Wireless Networks", *Proceedings of ACM/IEEE MOBICOM*, pp. 216-230, 2004.
- [6] G. Ciardo, A. Riska, and E. Smirni, "Equiloat: a Load Balancing Policy for Clustered Web Servers" *Performance Evaluation*, Vol. 46, pp. 101-124, 2001.
- [7] M. Crovella, M. Harchol-Balter and C. Murta, "Task Assignment in a Distributed System: Improving Performance by Unbalancing Load", *Proceedings of ACM SIGMETRICS*, Madison, WI, pp. 268-269, June 1998.
- [8] R. Draves, J. Padhye, and B. Zill, "Routing in Multi-radio, Multi-hop Wireless Mesh Networks", *Proceedings of ACM MobiCom*, pp. 114-128, 2004.
- [9] M. Gong and C. Williamson, "Simulation Evaluation of Hybrid SRPT Scheduling Policies", *Proceedings of IEEE/ACM MASCOTS*, Volendam, The Netherlands, pp. 355-363, October 2004.
- [10] P. Goyal, H. Vin, and H. Cheng, "A Scheduling Algorithm for Integrated Services Packet Switching Networks", *Journal of IEEE/ACM Transactions on Networking*, Vol 46, pp. 101-124, 1997.
- [11] V. Gupta, M. Harchol-Balter, K. Sigman, and W. Whitt "Analysis of Join-the-Shortest-Queue Routing for Web Server Farms" *Performance Evaluation*, Vol. 64, pp. 1062-1081, 2007.
- [12] B. Hara and A. Petrick, "IEEE 802.11 Handbook, A Designer's Companion", IEEE Press, 1999.
- [13] M. Harchol-Balter, M. Crovella and C. Murta, "On Choosing a Task Assignment Policy for a Distributed Server System", *Journal of Parallel and Distributed Computing*, pp. 204-228, 1999.
- [14] P. Hsiao, A. Hwang, H. Kung, and D. Vlah "Load-balancing Routing for Wireless Access Networks", *Proceeding of IEEE INFOCOM*, 2001.
- [15] B. Ko, V. Misra, J. Padhye, and D. Rubenstein, "Distributed Channel Assignment in Multi-Radio 802.11 Mesh Networks", *Proceedings of WCNC*, pp. 3978-3983, 2007.
- [16] J. Mo, H. So, and J. Walrand, "Comparison of Multi-channel MAC Protocols", *IEEE Transactions on Mobile Computing*, Vol. 7, pp. 50-65, 2008.
- [17] T. Nandagopal, S. Lu, and V. Bharghavan, "A Unified Architecture for the Design and Evaluation of Wireless Fair Queueing Algorithms", *Proceedings of ACM MobiCom*, pp. 355-363, 1999.
- [18] K. Ramachandran, E. Belding, K. Almeroth, and M. Buddhikot, "Interference-aware Channel Assignment in Multi-radio Wireless Mesh Networks", *Proceedings of IEEE INFOCOM*, pp. 1-12, 2006.
- [19] B. Schroeder and M. Harchol-Balter, "Evaluation of Task Assignment Policies for Supercomputing Servers: The Case for Load Unbalancing and Fairness", *Journal of Cluster Computing*, Vol 7, pp. 151-161, 2004.
- [20] A. Silberschatz and P. Galvin, *Operating System Concepts*, Addison-Wesley, 1998.
- [21] J. So and N. Vaidya, "Multi-Channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals Using A Single Transceiver", *Proceeding of ACM MobiHoc*, pp. 222-233, 2004.
- [22] W. Yoon, J. So, and N. Vaidya, "Routing Exploiting Multiple Heterogeneous Wireless Interfaces: A TCP Performance Study", *Proceedings of MILCOM*, pp. 1-7, 2006.