Message Queueing Analysis in Wireless Networks with Mobile Station Failures and Handoffs

Xinyu Chen and Michael R. Lyu
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong
+852-2609-8427
{xychen, lyu}@cse.cuhk.edu.hk

Abstract— Access Points play an essential role in fault tolerant architectures for mobile computing environments which engage wireless networks. They are the performance bottleneck in the presence of failures and handoffs of mobile stations. Different message dispatch strategies impose different effects on the message sojourn time in Access Points. In this paper, we study five dispatch models which are the basic queueing model, the static and the dynamic processor-sharing models, the round-robin model, and the feedback model. We derive the expected message sojourn time in Access Points under steady state. We observe that the basic model and the static processor-sharing model demonstrate the worst performance. The other three models cut down the sojourn time by dynamically reducing the probability of message blocking which is introduced by failures and handoffs of mobile stations; however, which one is the best dispatch strategy depends on the specific environments. These analysis results can help designers of wireless networks explore better fault tolerant features of mobile systems for their reliability and performance.

TABLE OF CONTENTS

1. INTRODUCTION

As the technology matures, wireless networks are being used in more applications providing significant benefits to mobile users. For example, in the battlefield, a general can gather real-time information from his soldiers and send commands to them. Wireless networks can be designed for supporting crew-computing tools aboard the International Space Station [1]. The planetary exploration may also employ mobile wireless networks as its communication system architecture [2]. Figure 1 shows a typical wireless network architecture which engages three main components: Mobile Station (MS), Access Point (AP), and Static Host (SH). An

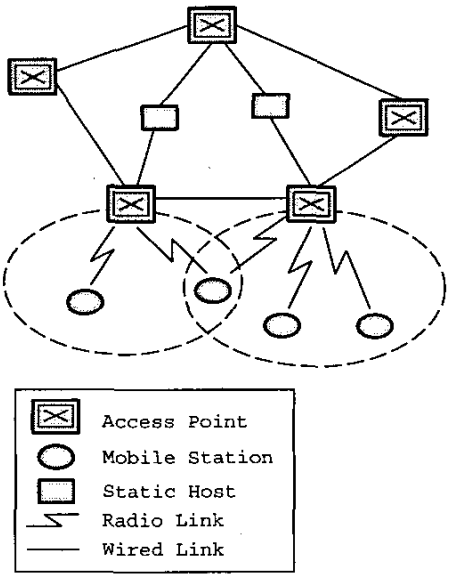

Figure 1. Wireless network architecture

MS moves in mobile wireless environments while maintaining network connections by a wireless interface. The AP is a transceiver device which transmits messages between the wireless and the wired networks and works as a message relay. The communications between MSs and APs are via radio links, while all APs and SHs are connected with wires. Each AP covers a geographical area (cell) within which it can communicate with its covered MSs directly. The cell areas are plotted as dashed ellipses in Figure 1. When an MS moves across the borders of the geographical areas, a *handoff* occurs between the old and the new APs. All hosts communicate with each other by messages only. All messages to and from an MS are buffered and relayed by its currently associated AP, and no messages can be exchanged among MSs directly. During a handoff, no computational messages can be transmitted between the APs and the MS.

0-7803-8155-6/04/$17.00/©2004 IEEE
IEEEAC paper #1146

Wireless computing enables users or explorers to access and exchange information while allowing them to roam around in mobile environments. This flexibility, however, causes more probable physical damage to MSs [3]. In addition, MSs contain low battery power and wireless links suffer limited bandwidth and long transfer delay, which make transient failures more likely. Fault tolerance is an essential feature which should be engaged in wireless networks, especially in battlefield wireless networks as a system failure may cause loss of human lives. Many researchers have delved into providing fault tolerance in wireless computing environments [3], [4], [5], [6], [7]. MSs are not suitable to be employed as the stable storage for saving checkpoints and message logs, therefore all these proposed recovery mechanisms select the AP as the stable storage. During failure-free execution, MSs take checkpoints and send the checkpoints to the currently connected APs. APs also log messages relayed by them. After an MS's failure, APs have to collect those scattered checkpoints and message logs due to movements of the MS, and resend these information to recover the MS's state. The techniques to tolerate AP failures are also explored in [5], [8]. Additionally it is pointed out that handoff can also be utilized as a mechanism to recover the AP failure [5].

From the above description, we learn that the AP plays an essential role in wireless networks. An AP is a message-relay station which disseminates messages between MSs and SHs. If an MS is in handoff, no computational messages can be relayed to it and messages received during handoff should be queued and buffered in the AP. The more probable failures also cause messages to be queued in the AP during the MS's recovery period. The AP, thus, becomes a bottleneck in improving the performance of fault tolerance in mobile environments. Therefore, it is essential to study the performance of the AP, such as the expected message sojourn time, in the presence of failures and handoffs of MSs. As different message processing strategies should demonstrate different effects on the expected message sojourn time, in this paper we consider five message dispatch strategies in the AP: the basic queuing model, the static and the dynamic processor-sharing models, the round-robin model, and the feedback model. We derive the expected message sojourn times under steady state for all the five models. We perform comparisons among their expected values, discuss the similarities and differences among these models, and determine whether they are suitable to be engaged as a message distribution strategy in an AP for wireless networks.

2. Related Work

Performance analyses for computer systems under failures have been conducted by many researchers. Some papers derive the program completion time under system or component failures. Tantawi and Ruschitzka [9] consider general failure distributions to compute the program completion time. Duda [10] derives the distribution and the expectation of program execution time under failures with and without checkpointing. Chen and Lyu [11] analyze the program completion time in mobile environments with failures and handoffs of MSs. Those derived distributions of the program completion time are essential to carry out the queueing analysis. On the other hand, the following papers conduct the queueing analysis. Nicola *et al.* [12] present a queueing analysis of a fault tolerant server in which jobs suffer delays due to failures and queueing. They utilize an irreducible continuous-time Markov chain to model the server's behavior. Gaver [13] analyzes a single server which is subject to random interruptions and gives the distributions of a job's completion time under different interruptions. Gelenbe and Derochette [14] conduct performance analysis of rollback recovery systems under intermittent failures. They employ a three-state Markov chain, which includes the normal state, the recovery state, and the checkpointing state, to model a database system. Kulkarni *et al.* [15] carry out the analysis of the program completion time in which checkpointing is allowed during reprocessing of the program, and they utilize the derived results to build a queueing model. All the above models assume that the input to a queue follows a Poisson fashion. In this paper, we also construct a three-state Markov chain but with different states to study the behavior of an MS. We conduct the queueing analysis and derive the expected message sojourn time. However, we assume that the message dispatch facility (AP) itself is not subject to failures and only the message target, i.e., the MS, undergoes failures and handoffs, which cause messages to be blocked and delayed in the message relay.

All the aforementioned models consider how checkpointing strategies affect the program completion time and do not take the server's program processing policy into consideration. Different program or message processing strategies, however, can also influence the program completion time. Therefore, many papers have been published to address this problem. Rasch [16] derives the queue size distribution and the average waiting time for a time-shared system using round-robin scheduling, with and without swap overhead. Coffman *et al.* [17] derive the Laplace-Stieltjes Transform (LST) of the waiting time distribution based on the service requirement and the system state. Kleinrock *et al.* [18] solve the average response time for jobs conditioned on their service requirement, in which different scheduling disciplines are incorporated, such as first come first served, feedback, and round-robin. Nunez-Queija [19] gives the LST of the sojourn-time distribution in an M/M/1 queue with the processor-sharing service discipline, in which the server is subject to breakdowns. He also points out that the expected sojourn time is not proportional to the service requirement. These papers discuss the system performance in the absence of server failures. The points of departure of this paper from others are that we solve the message sojourn time in APs in the presence of failures and handoffs of MSs for wireless networks.

3. State Assumptions

An MS conducts recoveries after failures and undergoes handoffs during its execution; therefore, it experiences three states: the normal (operational) state, the recovery state, and

the handoff state. Let $Z(t)$ be the state of the MS at time t. $\{Z(t), t \geq 0\}$ is a three-state Markov process with a state-transition diagram shown in Figure 2. We assume that the in-

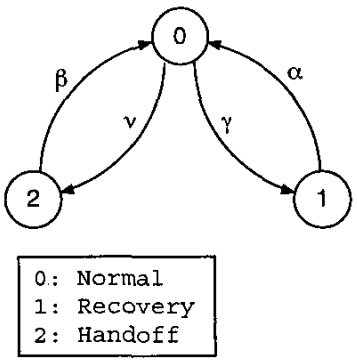

Figure 2. State transition diagram of mobile station

stants of the occurrences of failures to an MS form a homogeneous Poisson process with parameter γ and the time between two successive handoff events is modelled as an exponentially distributed random variable with parameter ν. State 0 is the normal state, during which messages can be dispatched to the MS. The MS may fail and then enters State 1. We assume that a failure is detected as soon as it occurs. A recovery process will be conducted in State 1. The recovery time R is regarded as an exponentially distributed random variable with parameter α. If a handoff occurs, the MS transits from State 0 to State 2. The handoff completion time H is also an exponentially distributed random variable but with parameter β. No failures take place during the recovery or the handoff period. According to the assumptions made before, we get the rate matrix of the three-state Markov process $\{Z(t), t \geq 0\}$ for the MS, which is

$$Q = \begin{bmatrix} -\gamma-\nu & \gamma & \nu \\ \alpha & -\alpha & 0 \\ \beta & 0 & -\beta \end{bmatrix}, \quad 0 < \alpha, \beta, \gamma, \nu < 1.$$

The stationary distribution of the Markov process is given by

$$\begin{bmatrix} p_0 & p_1 & p_2 \end{bmatrix} = \begin{bmatrix} \frac{\alpha\beta}{\alpha\beta+\alpha\nu+\beta\gamma} & \frac{\beta\gamma}{\alpha\beta+\alpha\nu+\beta\gamma} & \frac{\alpha\nu}{\alpha\beta+\alpha\nu+\beta\gamma} \end{bmatrix},$$

in which p_i denotes the probability of an MS in state i in the stationary situation. Note that we will use the notation $\phi_C(s)$ to denote the Laplace-Stieltjes Transform (LST) of $G_C(t)$ which is the cumulative distribution function (c.d.f.) of a random variable C. So $\phi_C(s) = \int_{t=0}^{\infty} e^{-st} dG_C(t) = E(e^{-sC})$.

4. Message Sojourn Time

The message arrival process for each MS used throughout this paper is assumed as a Poisson process with parameter λ which is the arrival intensity. If the number of MSs covered by an AP is n, $n > 0$, the total message arrival rate will be $n\lambda$ for an AP since the combination of Poisson inputs is still a Poisson input. The message dispatch requirement is assumed to be an exponentially distributed random variable D with parameter μ if the dispatch facility of an AP is occupied by one message exclusively, which means that the service rate of the dispatch facility is the constant μ. The message sojourn time in an AP, denoted as T, is the duration of a period between the instant when the message enters the AP and the instant when the message is totally disseminated. If a message starts its dissemination, the target MS stays in its normal state until the current message's dispatch is completed. This assumption is reasonable as the message dispatch requirement is relatively small under common situations. No messages can be distributed to an MS if the MS is in the recovery or the handoff state; therefore, an MS in these two states is called as an unavailable MS. The traffic intensity ρ_a for an AP is defined as $\rho_a = n\lambda E(D) = n\lambda/\mu$. With different dispatch strategies, the sojourn times are different in the presence of failures and handoffs of MSs. In the following sections we will consider five dispatch strategies: the basic model, the static and the dynamic processor-sharing models, the round-robin model, and the feedback model.

Basic Dispatch Model

The basic dispatch model is a normal M/M/1 queue in which all MSs share the same queue, shown in Figure 3. A message is deliverable when it is at the head of the queue and its corresponding MS is in the normal state. The message dispatch

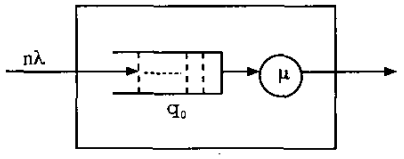

Figure 3. Basic dispatch model of an Access Point

time conditioned on dispatch requirement D, $X_{ba}(D)$, may be expressed by:

$$X_{ba}(D) = \begin{cases} D & : \ if \ Z = 0 \\ R + D & : \ if \ Z = 1 \\ H + D & : \ if \ Z = 2 \end{cases}.$$

If the message is deliverable, the dispatch time is D; otherwise, if the message is at the head of the queue and its targeted MS is in the recovery or the handoff state, it should be blocked until the MS returns back to the normal state. With the memoryless property of the exponential distribution, the residual recovery and handoff times are still R and H, respectively. Unconditioning on Z and D and applying $E(T) = E(X)/(1-\rho_a)$, the expected message sojourn time in the stationary state, $E_{ba}(T)$, is given by

$$E_{ba}(T) = \frac{E(D) + p_1 E(R) + p_2 E(H)}{1 - n\lambda E(D)}. \qquad (1)$$

We know that the sojourn time without failures and handoffs of MSs is $E(D)/(1-n\lambda E(D))$ [20]. The difference between these two sojourn times, then, is the recovery and the handoff time introduced by failures and handoffs of MSs.

Static Processor-Sharing Dispatch Model

In the basic dispatch model, when the message at the head of the queue is blocked, all subsequent messages will be blocked even though they could be dispatched if they get a chance to enter the dispatch facility. To solve this problem, we construct different queues for different MSs and come to the static processor-sharing model. In this model, messages arriving at an AP will be queued in n queues according to their target MSs. The dispatch facility is equally shared by these n queues, which implies that each queue is virtually associated with a dispatch facility whose service rate is μ/n. Figure 4

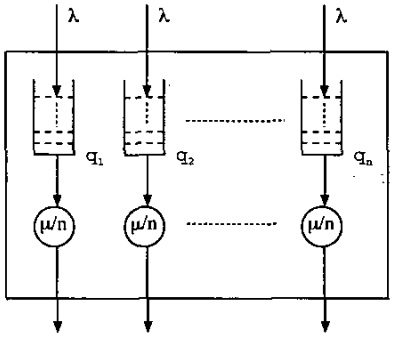

Figure 4. Static processor-sharing dispatch model of an Access Point

shows this dispatch model. The message dispatch time is expressed by

$$X_{sps}(D) = \begin{cases} nD & : \ if \ Z = 0 \\ R + nD & : \ if \ Z = 1 \\ H + nD & : \ if \ Z = 2 \end{cases},$$

and the expected sojourn time is

$$E_{sps}(T) = \frac{nE(D) + p_1 E(R) + p_2 E(H)}{1 - n\lambda E(D)}. \qquad (2)$$

Compared with the basic model, $E_{sps}(T) \geq E_{ba}(T)$. An intuitive explanation is that in the case of the static processor-sharing model, some capacities of the dispatch facility might be wasted when certain queues are empty, which indicates that the static processor-sharing dispatch policy is not a solution to the blocking problem induced by the basic model. One way to improve the performance of this model is to allocate the dispatch facility only among queues which contain deliverable messages. This introduces the dynamic processor-sharing dispatch model.

Dynamic Processor-Sharing Dispatch Model

The dynamic processor-sharing dispatch model puts messages into n queues the same way as the static processor-sharing dispatch model does; however, the dispatch facility is dynamically shared among the MSs who are in the normal state and whose corresponding queues are not empty. This condition can be restated that the dispatch facility is dynamically shared among queues which contain deliverable messages. Figure 5 shows this dispatch model. At a specific

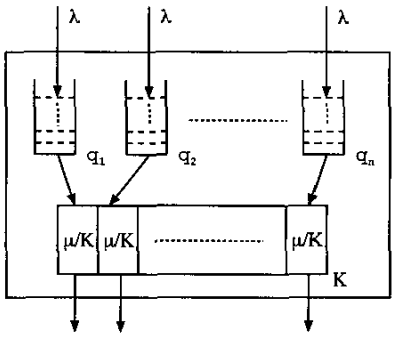

Figure 5. Dynamic processor-sharing dispatch model of an Access Point

time, the dispatch facility contains K deliverable messages, in which each message comes from different queues. As the behaviors of n queues are the same, without loss of generality, we only analyze the experience of the message at the head of q_1. When the message becomes deliverable and enters the dispatch facility, there are other $(K-1)$ deliverable messages in the dispatch facility. Therefore, this message receives dispatch service with rate μ/K and the traffic intensity varies with K. Let ρ_d be the expected traffic intensity for each queue. Since

$$P(K = k) = \binom{n-1}{k-1} (\rho_d p_0)^{k-1} (1 - \rho_d p_0)^{n-k}, \ 1 \leq \ k \leq n.$$

we get

$$\rho_d = \frac{\lambda E(D)}{1 - p_0 \lambda E(D)(n-1)}. \qquad (3)$$

The message dispatch time conditioned on K is given by

$$X_{dps}(D|\ K = \begin{cases} KD & : \ if \ Z = 0 \\ R + KD & : \ if \ Z = 1 \\ H + KD & : \ if \ Z = 2 \end{cases}.$$

Unconditioning on Z, K, and D, we get the message sojourn time as

$$E_{dps}(T) = \frac{[\rho_d p_0 (n-1) + 1] E(D) + p_1 E(R) + p_2 E(H)}{1 - \rho_d}. \qquad (4)$$

We know that the static and the dynamic processor-sharing models are the same when $n = 1$. As a consequence, Equation (4) equals Equation (2) when $n = 1$.

Round-Robin Dispatch Model

For the dynamic processing-sharing model, the dispatch facility is shared among queues which contain deliverable messages. If we dedicate the facility exclusively for one queue at a time and make the facility serve each queue in a sequential and cyclic turn, the round-robin dispatch model comes into view. When the dispatch facility visits a message queue, if the queue contains a deliverable message, it will pick up the

$$\phi_{X_r}(s) = \frac{\phi_D(s)[p_0 + (1-p_0\rho)^{n-1}(p_1\phi_R(s) + p_2\phi_H(s))]}{1-(1-p_0)[(1-p_0\rho+p_0\rho\phi_D(s))^{n-1}-(1-p_0\rho)^{n-1}]} \quad (6)$$

$$E(X_r) = \frac{p_0E(D)[1+\rho(1-p_0)(n-1)] + (1-p_0\rho)^{n-1}[(1-p_0)E(D)+p_1E(R)+p_2E(H)]}{p_0+(1-p_0)(1-p_0\rho)^{n-1}} \quad (7)$$

message from the head of the queue and dispatch it to its MS, and then turn to the next message queue. If there are no deliverable messages in the queue, it will visit the next message queue without overhead or swap time. Due to that no swap time is assumed, we note that when no messages have been delivered in the last turn for each queue, the dispatch facility should wait at the current queue until its corresponding MS returns back to work; otherwise, the dispatch process would not make any progress. Figure 6 shows this processing model.

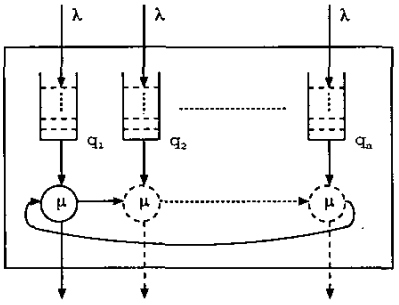

Figure 6. Round-robin dispatch model of an Access Point

The message dispatch time $X_{rr}(D)$ is given by

$$X_{rr}(D) = \begin{cases} D \\ \quad : if\ Z=0 \\ R+D \\ \quad : if\ Z=1\ n\ d\ K=0 \\ H+D \\ \quad : if\ Z=2\ and\ K=0 \\ \sum_{i=1}^{K} D_i + \tilde{X}_r(D) \\ \quad : if\ Z=1,2\ and\ K\geq 1 \end{cases}, \quad (5)$$

in which $\tilde{X}_{rr}(D)$ is a random variable with the same probability distribution as $X_{rr}(D)$ and K is the number of delivered messages in the last service turn before the dispatch facility visits the queue again. If $Z=0$, the message will be dispatched with time D. If $Z=1,2$ and $K=0$, the dispatch facility should stay in the current queue until the target MS returns back to operation and then disseminates the message. If $Z=1,2$ and $K\geq 1$, the message should be blocked in the queue until the dispatch facility visits the queue again to decide whether or not for dissemination with dispatch time $\tilde{X}_r(D)$. The time to the next visit is $\sum_{i=1}^{K} D_i$ if K messages will be delivered in this turn among the other $(n-1)$ queues.

Let the traffic intensity for each queue be $\rho = \lambda/\mu$. Assuming that the probability of a queue being busy at equilibrium is ρ, we obtain

$$P(K=k) = \binom{n-1}{k}(p_0\rho)^k(1-p_0\rho)^{n-1-k},\ 0\leq k\leq n-1.$$

Taking LST for Equation (5) and applying $\phi_{X_r}(s) = \phi_{\tilde{X}_{rr}}(s)$, we get Equation (6). After engaging the moment generating property of the Laplace transform [21], we obtain Equation (7) and

$$E(T_{rr}) = \frac{E(X_{rr})}{1-\rho}. \quad (8)$$

We can see when $n=1$ the round-robin model is reduced to the process-sharing model.

Feedback Dispatch Model

Another way to reduce the blocked overhead forced by the message at the head of the queue in the basic model is to move the blocked message away and to yield the delivery chance to the subsequent messages. The removed message should be fed back to the queue, which introduces the feedback dispatch model, shown in Figure 7. In this model, each

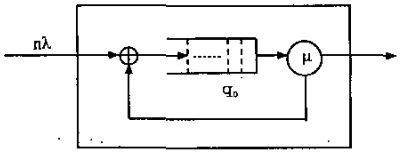

Figure 7. Feedback dispatch model of an Access Point

removed message is instantaneously added to the tail of the queue in its original turn. We also assume that no overhead or swap time is associated with the process of loading and unloading messages from the dispatch facility [22]. Let M be the total number of messages in the queue and in the dispatch facility when a message enters the queue. Again we note that when the message arrives at the dispatch facility and finds its MS is not available whereas all its previous M messages have been fed back, it should stay there until its MS returns back to work. It should not be fed back; otherwise, the dispatch process would make no progress.

Analysis of this model is more difficult due to the fact that the message input to the queue q_0 is no longer a Poisson input, since the message arrivals join the queue via the feedback path [16]. However, as the service rate of the dispatch

$$T_{fb}(D) = \begin{cases} D & : \ if \ Z = 0 \ and \ (M = 0 \ or \ M \neq 0 \ and \ K = 0)) \\ R + D & : \ if \ Z = 1 \ and \ (M = 0 \ or \ M \neq 0 \ and \ K = 0)) \\ H + D & : \ if \ Z = 2 \ and \ (M = 0 \ or \ M \neq 0 \ and \ K = 0)) \\ V + \sum_{i=1}^{K-1} D_i + D & : \ if \ Z = 0 \ and \ (M \neq 0 \ and \ 1 \leq K \leq M) \\ V + \sum_{i=1}^{K-1} D_i + W_{fb}(D) & : \ if \ Z = 1, 2 \ and \ (M \neq 0 \ and \ 1 \leq K \leq M) \end{cases} \quad (9)$$

$$\phi_{T_{fb}}(s) = \frac{\phi_D(s)(1-\rho_a)\left[\frac{p_1\phi_R(s) + p_2\phi_H(s)}{1-(1-p_0)\rho_a} + \frac{p_0}{1-(1-p_0+p_0\phi_D(s))\rho_a}\right]}{1+(1-p_0)(1-\rho_a)\left[\frac{1}{1-(1-p_0)\rho_a} - \frac{1}{1-(1-p_0+p_0\phi_D(s))\rho_a}\right]} \quad (10)$$

$$E(T_{fb}) = \frac{[(1-p_0)E(D) + p_1E(R) + p_2E(H)](1-\rho_a)^2 + p_0E(D)[1-(1-p_0)\rho_a]}{(1-\rho_a)[1-(1-p_0)^2\rho_a]} \quad (11)$$

facility is a constant μ, the distribution of stationary message number in the queue and in the dispatch facility for the feedback model is the same as the one for a normal M/M/1 queue, which is stated by PASTA property [20]. Therefore, $P(M = m) = (1-\rho_a)\rho_a^m$. However, the message sojourn time in the AP cannot be determined as the same as the sojourn time in the M/M/1 queue, so we derive it from another approach as follows. It can be expressed by Equation (9), in which K is the number of messages that have been dispatched since a message enters the AP until the message reaches the head of the queue at the first time. V is the residual service time of a message currently in service when the message enters the AP and $W_{fb}(D)$ denotes the time period between the instant that the message is fed back to the tail of the queue and the instant that the message finishes its dispatch. Note $V + \sum_{i=1}^{K-1} D_i$ is the waiting time in the queue after the message enters the queue and before it enters the dispatch facility during which K messages have been sent out. K follows a binormal distribution as

$$P(K = k|M) = \binom{M}{k} {p_0}^k (1-p_0)^{M-k}.$$

Since the service time D is exponentially distributed and the message, after being fed back, joins the queue with the prior knowledge that the message in dissemination has just started [16], V and $W_{fb}(D)$ employ the same distributions as those of D and $T_{fb}(D)$, respectively. Following the same steps as those in the round-robin model, we get Equation (10), the LST of the c.d.f. of the message sojourn time. The corresponding expected message sojourn time, $E(T_{fb})$, is given by Equation (11).

5. Comparisons and Discussions

In this section we compare different dispatch models in order to observe the relationships among them and to determine which one is the best under different conditions according to our defined performance metric: the expected message sojourn time. Some parameter values are not changed throughout this section, and we give them here: $\alpha = 10^{-1}$, $\beta = 1$, $\nu = 10^{-3}$. The handoff operation is a mobile characteristic provided by wireless networks and only contains some message exchanges, while the recovery procedure includes an additional state recovery. Therefore, the expected handoff completion time should be less than the expected recovery time.

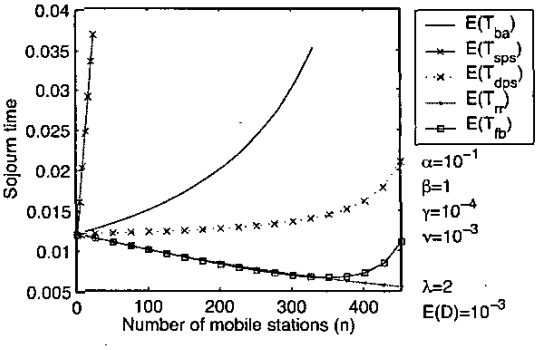

Figure 8. Average sojourn time vs. number of mobile stations

Figure 8 shows the results of average message sojourn time in an AP when the number of MSs increases. Different dispatch models demonstrate various relationships with the increase of the number of MSs, n. The basic model and the dynamic processor-sharing model exhibit moderate increases while the static processor-sharing model experiences sharp increase. The dispatch capacity dedicated to a certain MS decreases with n in the processor-sharing models, which induces the increase of the sojourn time; however, dynamic scheduling compensates part of loss of the dispatch capacity . $E(T_{fb})$ decrease first and then increase later. $E(T_{rr})$ shows no discernible increase in Figure 8. The round-robin model and the feedback model are different with other models in that they give dispatch opportunities to other dispatchable messages instead of blocking them by yielding the dispatch facility. They hide the unavailable periods by dispatching

messages for other MSs. Therefore, their sojourn times enjoy decrease with the number of MSs. However, if too many MSs are present, the round time to the next dispatch opportunity will exceed the blocking time, which cancels out the advantage and thus the expected sojourn times increase. From this figure, we can see that when $n = 1$, all models take the same effect in the expected message dispatch time. Actually under this condition, all the other four models are reduced to the basic model.

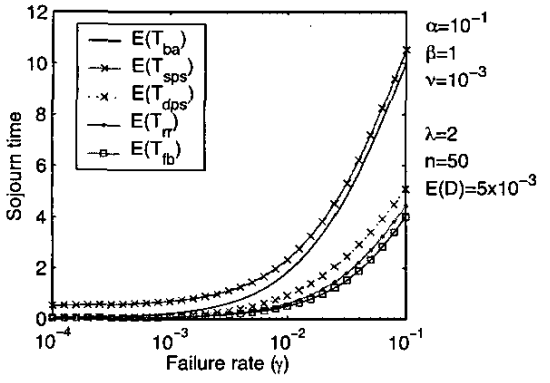

Figure 9. Average sojourn time vs. failure rate

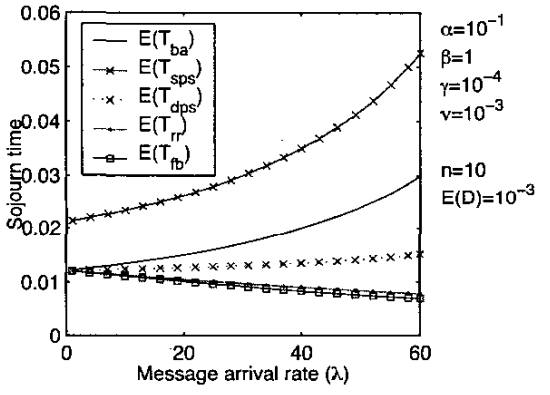

Figure 10. Average sojourn time vs. message arrival rate

We next show how the failure rate affects the expected sojourn time in Figure 9. The sojourn times of all the five models increase with the failure rate γ. This is obvious as the higher the failure rate is, the greater the undeliverable probability for a message is. When the failure rate is low, the four models except the static processor-sharing model are almost the same as they make the best of resource utilizations, while the static processor-sharing model wastes part of the dispatch capacity due to the low traffic intensity. As the failure rate increases, the performance of the basic model approaches to that of the static processor-sharing model. The performances of the other three models maintain a good level with little divergence. When the failure arrival rate is high enough, the dispatch facility is occupied by one message exclusively with a large probability in the dynamic processor-sharing model. Under this condition, these three models converge. These observations can also be made with the handoff rate ν. Prolonging the recovery period or the handoff period apparently increases the message sojourn time.

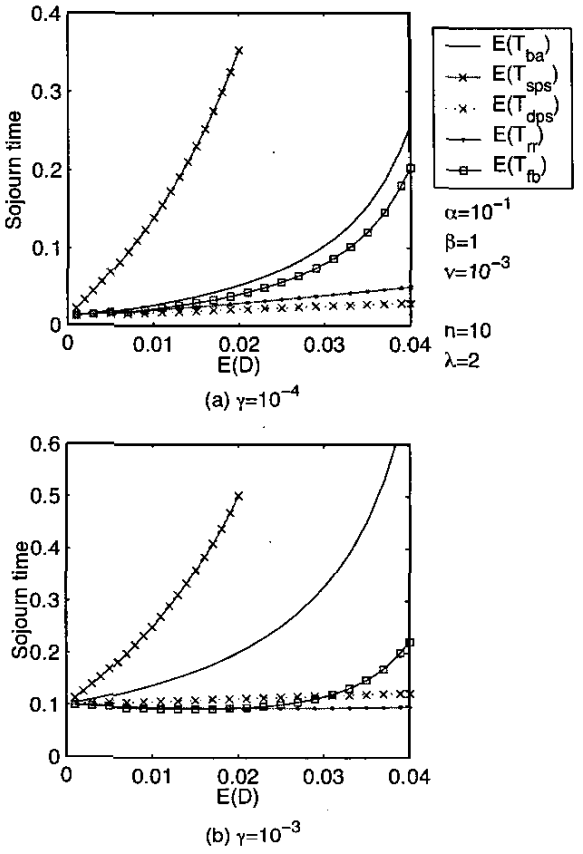

Figure 11. Average sojourn time vs. expected message dispatch requirement

The message arrival rate λ produces similar effects as the number of MSs in all the five models, shown in Figure 10. With different failure rates, some models exhibit different behaviors with the expected message dispatch requirement $E(D)$ which indicates the capacity of the dispatch facility, shown in Figure 11. Increasing the dispatch requirement adds the sojourn time in all the five models under a relatively low failure rate. An interesting observation is that with a relatively high failure rate, the round-robin and the feedback models may demonstrate decrease in the sojourn time, although the decrease is slight. This is also due to the dispatch facility or message cyclic characteristic provided by these two models.

From all of the above figures, the static processor-sharing model demonstrates the worst performance, and the basic model exhibits the second worst performance. The other three models outperform these two models; however, their relative performance varies under different conditions, including the number of MSs, the failure rate, the message arrival rate, and so on. We note, however, that there is only one queue in the basic and the feedback models. When the num-

ber of MSs is large, the message arrival rate should be small; otherwise the queue in these two models will not be stable.

6. CONCLUSIONS

In this paper, we perform analyses for message sojourn time within an Access Point in the presence of failures and handoffs of mobile stations in wireless networks. To see how different message dispatch strategies influence the sojourn time, we study five dispatch models. We derive the expected message sojourn time under steady state. Analytical and numerical results show that the basic model and the static processor-sharing dispatch model demonstrate the worst performance. The other three models may be suitable for applications as the dispatch strategy for the Access Point; however, the runtime environment determines which one should be implemented.

7. ACKNOWLEDGEMENTS

The work described in this paper was fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CUHK4182/03E).

REFERENCES

[1] R. Alena, E. Yaprak, and S. Lamouri, "Modeling a wireless network for international space station," in *Proc. 2000 IEEE Aerospace Conference*, vol. 1, Mar. 2000, pp. 223–228.

[2] R. Alena, D. Evenson, and V. Rundquist, "Analysis and testing of mobile wireless networks," in *Proc. 2002 IEEE Aerospace Conference*, vol. 3, Mar. 2002, pp. 1131–1144.

[3] N. Neves and W. K. Fuchs, "Adaptive recovery for mobile environments," *Communications of the ACM*, vol. 40, no. 1, pp. 68–74, Jan. 1997.

[4] A. Acharya and B. R. Badrinath, "Checkpointing distributed applications on mobile computers," in *Proc. the 3rd International Conference on Parallel and Distributed Information Systems*, Sept. 1994, pp. 73–80.

[5] X. Chen and M. R. Lyu, "Message logging and recovery in wireless CORBA using access bridge," in *Proc. the 6th International Symposium on Autonomous Decentralized Systems*, Pisa, Italy, Apr. 2003, pp. 107–114.

[6] T. Park and H. Y. Yeom, "An asynchronous recovery scheme based on optimistic message logging for the mobile computing systems," in *Proc. the 20th International Conference on Distributed Computing Systems*, Apr. 2000, pp. 436–443.

[7] D. K. Pradhan, P. Krishna, and N. H. Vaidya, "Recovery in mobile environments: Design and trade-off analysis," in *Proc. the 26th International Symposium on Fault-Tolerant Computing*, June 1996.

[8] S. Alagra, R. Rajagopalan, and S. Venkatesan, "Tolerating mobile support station failures," in *Proc. the 1st Conference on Fault Tolerant Systems*, Madras,India, Dec. 1995, pp. 225–231.

[9] A. N. Tantawi and M. Ruschitzka, "Performance analysis of checkpointing strategies," *ACM Transactions on Computer Systems*, vol. 2, no. 2, pp. 123–144, June 1984.

[10] A. Duda, "The effects of checkpointing on program execution time," *Information Processing Letters*, vol. 16, pp. 221–229, June 1983.

[11] X. Chen and M. R. Lyu, "Performance and effectiveness analysis of checkpointing in mobile environments," in *Proc. the 22nd Symposium on Reliable Distributed Systems*, Florence, Italy, Oct. 2003.

[12] V. F. Nicola, V. G. Kulkarni, and K. S. Trivedi, "Queueing analysis of fault-tolerant computer systems," *IEEE Trans. Software Eng.*, vol. 13, pp. 363–375, Mar. 1987.

[13] D. P. Gaver Jr, "A waiting line with interrupted service, including priorities," *Journal of the Royal Statistical Society, Series B*, vol. 24, pp. 73–90, 1962.

[14] E. Gelenbe and D. Derochette, "Performance of rollback recovery systems under intermittent failures," *Communications of the ACM*, vol. 21, no. 6, pp. 493–499, June 1978.

[15] V. G. Kulkarni, V. F. Nicola, and K. S. Trivedi, "Effects of checkpointing and queueing on program performance," *Communications in Statistics - Stochastic Models*, vol. 6, no. 4, pp. 615–648, 1990.

[16] P. J. Rasch, "A queueing theory study of round-robin scheduling of time-shared computer systems," *Journal of the ACM*, vol. 17, no. 1, pp. 131–145, Jan. 1970.

[17] E. G. Coffman, "Waiting time distributions for processor-sharing systems," *Journal of the ACM*, vol. 17, no. 1, pp. 123–130, Jan. 1970.

[18] L. Kleinrock and R. R. Muntz, "Processor sharing queueing models of mixed scheduling disciplines for time shared systems," *Journal of the ACM*, vol. 19, pp. 464–482, July 1972.

[19] R. Nunez-Queija, "Sojourn times in a processor sharing queue with service interruptions," *Queueing Systems*, vol. 34, pp. 351–386, 2000.

[20] B. R. Haverkort, *Performance of Computer Communication Systems: A Model-Based Approach.* Chichester: John Wiely & Sons, 1998.

[21] J. L. Schiff, *The Laplace Transform: Theory and Applications.* New York: Springer, 1999.

[22] E. G. Coffman, "Feedback queueing models for time-shared systems," *Journal of the ACM*, vol. 15, no. 4, pp. 549–576, Oct. 1968.

Xinyu Chen *received the B.E. degree in mechanical engineering from Beijing Institute of Technology, Beijing, China, in 1997 and the M.E. degree in signal and information processing from Peking University, Beijing, China, in 2000. Now he is a Ph.D. candidate in the Department of Computer Science and Engineering at the Chinese University of Hong Kong, Hong Kong, China. His research interests include fault-tolerant distributed systems and mathematical modelling.*

Dr. Michael R. Lyu *received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, China, in 1981, the M.S. degree in computer engineering from University of California, Santa Barbara, in 1985, and the Ph.D. degree in computer science from University of California, Los Angeles, in 1988.*

He is currently a Professor in the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong. He was with the Jet Propulsion Laboratory as a Technical Staff Member from 1988 to 1990. From 1990 to 1992, he was with the Department of Electrical and Computer Engineering, The University of Iowa, Iowa City, as an Assistant Professor. From 1992 to 1995, he was a Member of the Technical Staff in the applied research area of Bell Communications Research (Bellcore), Morristown, New Jersey. From 1995 to 1997, he was a Research Member of the Technical Staff at Bell Laboratories, Murray Hill, New Jersey. His research interests include software reliability engineering, distributed systems, fault-tolerant computing, wireless communication networks, Web technologies, digital libraries, and E-commerce systems. He has published over 150 refereed journal and conference papers in these areas. He received Best Paper Awards in ISSRE'98 and ISSRE'2001. He has participated in more than 30 industrial projects, and helped to develop many commercial systems and software tools. He was the editor of two book volumes: Software Fault Tolerance (New York: Wiley, 1995) and The Handbook of Software Reliability Engineering (Piscataway, NJ: IEEE and New York: McGraw-Hill, 1996).

Dr. Lyu initiated the First International Symposium on Software Reliability Engineering (ISSRE) in 1990. He was the program chair for ISSRE'96, and has served in program committees for many conferences, including ISSRE, SRDS, HASE, ICECCS, ISIT, FTCS, DSN, ICDSN, EUROMICRO, APSEC, PRDC, PSAM, ICCCN, ISESE, and WWW. He was the General Chair for ISSRE2001, and the WWW10 Program Co-Chair. He has been frequently invited as a keynote or tutorial speaker to conferences and workshops in U.S., Europe, and Asia. He has been an Associate Editor of IEEE Transactions on Reliability, IEEE Transactions on Knowledge and Data Engineering, and Journal of Information Science and Engineering. Dr. Lyu is a fellow of IEEE.