SANDIA REPORT
SAND95–0974 • UC–405
Unlimited Release
Printed December 1997

Network Congestion Can Be Controlled: Routing Algorithms in Optical Networks and Ethernets

Leslie A. Goldberg, Philip D. MacKenzie, David S. Greenberg

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A03
Microfiche copy: A01

SAND95–0974
Unlimited Release
Printed December 1997

Distribution
Category UC–405

Network congestion can be controlled: Routing algorithms in optical networks and ethernets

Leslie A. Goldberg, Philip D. MacKenzie, David S. Greenberg
Algorithms and Discrete Mathematics Department
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-1110
(Leslie is currently in the Department of Computer Science at University of Warwick,
Phil is currently in the Department of Mathematics and Computer Science at Boise State University)

Abstract

Congestion and contention can greatly reduce the effective performance of an interconnection network. This report gathers together research done under a Laboratory Research and Development Program (LDRD) project at Sandia National Laboratories. The goal of the project was to explore the contention properties of novel optical interconnects. In the process of exploring optical interconnects the project also gained new insights into the use of backoff protocols in the current dominant interconnect technology, Ethernet.

1 Introduction

Over the last ten years Sandia has had great success by replacing the old monolithic supercomputer model with the massively parallel processing (MPP) model. In conjunction with both nCUBE and Intel Sandia has developed machines which contain thousands of processors. Groups at IBM and SGI/Cray have followed suit. Currently many groups are working on clustering commodity building blocks to attempt to emulate the success of MPPs without incurring the cost creating custom networks.

The research described in this report was targetted to look at networks of the future which might become useful in creating MPPs or distributed supercomputers. Three key properties determine the usefulness of a network to MPPs: latency, bandwidth, and congestion.

Latency is the time required for the first bit of a message to arrive from one node to another. Currently much of the latency time derives from software overhead and from memory latency on the nodes. A small amount of latency is due to the travel time through the network. At current clock speeds even short distances can yield measurable speed-of-light transit times. However, there is little which can be done within the network design to avoid these latencies. Thus when latency enters into the research it is usually in the form of ensuring that latency has not been increased rather than in looking for ways to decrease latency.

Bandwidth is the rate at which data can flow through the network between two nodes. Networks with higher clock rates and/or wider links have higher bandwidths. Optical technologies can potentially greatly increase the bandwidth available. By making the size of links smaller it can also allow there to be more links and thus increase the system wide bandwidth. Understanding the implications of using optical interconnects was thus chosen as a focus of the research described in this report.

Perhaps the greatest effect of network design is in the amount of congestion observed. Congestion can occur when several messages attempt to use the same link at the same time. In a fully connected network (with a direct link between every pair of nodes) there is no congestion on the links. Unfortunately with electric wires (or circuit board lines) it becomes prohibitively expensive to produce fully-connected networks for thousands of nodes. The ability of optical networks to broadcast a message in space and use wave properties to avoid interference provides a way around the cost of creating direct links between millions of pairs. However, the practical use of such a system must also address congestion due to many messages arriving at a single node at a time. This project set out to build on earlier work which suggested that better algorithms could limit arrival congestion.

One early surprise in the research was that it became apparent that the techniques being developed for optical interconnects of the future applied well to ethernet networks of the past and present. Ethernet is also based on broadcast over a shared medium, in this case an electrical bus rather than an optical media. Backoff protocols to recover from contention for the bus have been in use since 1976. However, there has remained room for theoretical improvement of the standard ethernet protocols. The majority of this report is a reprinting of a paper by the first two authors showing that superlinear backoff protocols are stable for a wide range of ethernet uses. This paper also includes a discussion of the use of the new protocol within optical networks.

2 Optical Networks

Over the last several years a standard model of optical networks has been developed. In this model each node is assumed to have a transmitter which can in each time step send a message to *any* other node. Each node also has a receiver which can receive a message from any other node. However, the receiver is not capable of processing multiple incoming messages at the same time. Thus, if only one message is sent to this node it is received properly but if more than one message is sent than nothing is received. It is relatively straightforward to show ways of providing confirmation of reception so it is assumed that the sender knows whether the message arrived or not.

Within this model a standard algorithmic test is the routing of an h-relation. In an h-relation each node has a message to send to at most h other nodes and each node is the target of at most h messages. The send bound is easy to check locally by having each node count its number of outgoing messages. The receive bound is less obvious to check and most algorithms just assume it is a given. The challenge is to avoid sending several messages to the same node concurrently while using only the ability to send messages as a means of coordination.

Several papers describing this model and work done in the model prior to this project are listed below.

- Leslie Ann Goldberg, Yossi Matias and Satish Rao, An Optical Simulation of Shared Memory. SPAA 6 (1994) 257-267. Submitted for journal publication.
- Leslie Ann Goldberg, Mark Jerrum and Phil MacKenzie, A Lower Bound for Routing on a Completely Connected Optical Communication Parallel Computer. To appear in SIAM Journal on Computing.
- Leslie Ann Goldberg, Mark Jerrum, Tom Leighton, and Satish Rao, Doubly Logarithmic Communication Algorithms for Optical Communication Parallel Computers. To appear in SIAM Journal on Computing.
- Leslie Ann Goldberg, Routing in Optical Networks: The Problem of Contention, in Interconnection Networks and Mapping and Scheduling Parallel Computations, DIMACS Series in Discrete Mathematics Vol 21. (Frank Hsu, Arnold Rosenberg and Dominique Sotteau, eds.) American Mathematical Society 1995.

Subsequent to leaving the Sandia project the first two authors also published the following paper.

Leslie Ann Goldberg and Philip D. MacKenzie, Contention Resolution with Guaranteed Constant Expected Delay, to appear in the Proceedings of the Symposium on Foundations of Computer Science 38 (1997).

3 Ethernet backoff protocols

The majority of the results of this project were published as the following paper which is reprinted in its entirety as Appendix A.

Leslie Ann Goldberg and Philip D.MacKenzie, Analysis of Practical Backoff Protocols for Contention Resolution with Multiple Servers. Proceedings of SODA 7 (1996) 554-563. Submitted for journal publication.

4 Summary

The LDRD project described in this report set out to better understand interconnection networks for potential use in future MPPs. New algorithms and theoretical bounds were discovered for both optical networks and ethernet busses

Appendix A: Reprinted paper

Analysis of Practical Backoff Protocols for Contention Resolution with Multiple Servers*

Leslie Ann Goldberg† Philip D. MacKenzie‡

Abstract

Backoff protocols are probably the most widely used protocols for contention resolution in multiple access channels. In this paper, we analyze the stochastic behavior of backoff protocols for contention resolution among a set of clients and servers, each server being a multiple access channel that deals with contention like an Ethernet channel. We use the standard model in which each client generates requests for a given server according to a Bernoulli distribution with a specified mean. The *client-server request rate* of a system is the maximum over all client-server pairs (i, j) of the sum of all request rates associated with either client i or server j. (Having a sub-unit client-server request rate is a necessary condition for stability for single-server systems.) Our main result is that any superlinear polynomial backoff protocol is stable for any multiple-server system with a sub-unit client-server request rate. Our result is the first proof of stability for any backoff protocol for contention resolution with multiple servers. (The multiple-server problem does not reduce to the single-server problem, because each client can only send a single message at any step.) Our result is also the first proof that *any* weakly acknowledgment based protocol is stable for contention resolution with multiple servers and such high request rates. Two special cases of our result are of interest. Hastad, Leighton and Rogoff have shown that for a single-server system with a sub-unit client-server request rate any *modified* superlinear polynomial backoff protocol is stable. These modified backoff protocols are similar to standard backoff protocols but require more random bits to implement. The special case of our result in which there is only one server extends the result of Hastad, Leighton and Rogoff to standard (practical) backoff protocols. Finally, our result applies to dynamic routing in optical networks. Specifically, a special case of our result demonstrates that superlinear polynomial backoff protocols are stable for dynamic routing in optical networks.

1 Introduction

We study the problem of contention resolution with multiple clients and multiple servers. We assume that each server handles contention as follows: when multiple clients attempt to access the server at the same time, none succeed. This is the contention-resolution mechanism that is used in an Ethernet channel. Specifically, a client attempts to access an ethernet channel by sending a message to the channel. If no other messages are sent to the channel at the same time then the client's message is received and the client receives an acknowledgment. Otherwise, the message is

*This work was performed at Sandia National Laboratories and was supported by the U.S. Department of Energy under contract DE-AC04-76DP00789. Address: Sandia National Laboratories, MS1110, PO Box 5800, Albuquerque, NM 87185-1110.

†E-mail: lagoldb@cs.sandia.gov

‡E-mail: philmac@cs.sandia.gov

not received and the client must retransmit the message. The clients in the system use a contention-resolution protocol to decide when to retransmit. During the time that a client is trying to send one message, it may generate more messages that it needs to send. These messages are stored in a buffer. An important feature of a good contention-resolution protocol is that, even when messages are generated fairly frequently, the size of the buffers that are used remain bounded.

We use the standard model in which each client generates requests for a given server according to a Bernoulli distribution with a specified mean. Following Håstad, Leighton and Rogoff [HLR93], we say that a contention-resolution protocol is *stable* for the specified request rates if the expectation of the average waiting time incurred by a message before it is successfully delivered is finite and the expectation of the time that elapses before the system returns to the initial state (in which there are no messages waiting in the buffers) is also finite. It is easy to see that if a protocol is not stable then the buffers that it requires to store waiting messages grow larger and larger over time and the amount of time that it takes to send each message increases without bound.

1.1 Related Previous Work

The most popular protocol that is used for contention-resolution on an Ethernet is the *Binary Exponential Backoff Protocol* of Metcalfe and Boggs [MB76]. In this protocol each client maintains a counter, b, which keeps track of the number of times that the client has tried to send its message and failed. After it unsuccessfully tries to send a message, it chooses t uniformly at random from the set $\{1, \ldots, 2^b\}$ and it retransmits after t steps. (In practice, a truncated Binary Exponential Backoff Protocol is usually used, in which t is chosen uniformly at random from $\{1, \ldots, 2^{\min\{10,b\}}\}$. Many works refer to this truncated version as "Binary Exponential Backoff.")

Most of the previous results on contention-resolution protocols concern systems in which the number of clients is infinite. As [HLR93] explains, these results have limited relevance to the finite case. It has not been shown that the Binary Exponential Backoff Protocol is stable for Ethernets with a finite number of clients. However, there are some related results. In [GGMM88], Goodman, Greenberg, Madras and March modify the protocol as follows. If a client has unsuccessfully tried to send a message then on each successive step (until the message is successfully delivered), it retransmits the message with probability 2^{-b}. (The decision as to whether to retransmit the message is independent of all previous decisions.) This Modified Binary Exponential Backoff Protocol is similar to the original protocol, but it is not implemented in practice because it requires too many random bits (a random number is required at every time-step). [GGMM88] shows that the modified protocol is stable as long as the sum of the request rates, which we refer to as λ, is sufficiently small. (The definition of stability that is used in [GGMM88] is actually slightly weaker than the one that we use above.) [HLR93] shows that if $\lambda > 1/2$, the modified protocol is unstable. However, it shows that any Modified Superlinear Polynomial Backoff Protocol (in which a client re-transmits with probability $(b+1)^{-\alpha}$) is stable as long as $\alpha > 1$ and $\lambda < 1$.

In [RU95], Raghavan and Upfal consider the problem of contention resolution with multiple servers, each of which handles contention in the same way as an Ethernet channel. Note that this problem does not reduce to multiple instances of the single-server problem because each client can send only one message on each time step. Raghavan and Upfal describe a contention-resolution protocol that is stable as long as the sum of the request rates associated with any client or server is bounded from above by a constant $\lambda' < 1$. The expected waiting time of a message in their protocol is $O(\log N)$. This is much smaller than the expected waiting time of messages in any backoff protocol, which they show to be $\Omega(N)$. However, their protocol is more complicated than a backoff protocol and λ' may be small compared to 1, so their protocol may not be stable for high request rates. Furthermore, like the modified backoff protocols, their protocol requires random

number generation on each time step. For these reasons, it seems likely that backoff protocols will continue to be used in practice for contention resolution with multiple servers.

1.2 Our Results

The *client-server request rate* of a system is the maximum over all client-server pairs (i, j) of the sum of all request rates associated with either client i or server j. Having a sub-unit client-server request rate is a necessary condition for stability for single-server systems. Our main result is that any superlinear polynomial backoff protocol is stable for any multiple-server system with a sub-unit client-server request rate.

Our result extends the previous results in the following ways. First, our result is the first stability proof that applies to standard (un-modified) backoff protocols. This is important because the standard protocols are used in practice[1]. The special case of our result in which there is just one server extends the result of [HLR93] to standard (practical) backoff protocols. Second, our result is the first stability proof for any backoff (or modified backoff) protocol for contention resolution with multiple servers. Thus, our result generalizes the result of [HLR93] to the multiple-server case.

We say that a contention-resolution protocol is *weakly acknowledgment based* if each client decides whether to transmit on a given step without knowing anything about the other clients other than the number of clients in the system and the results of its own previous transmissions. Our result is the first proof that any weakly acknowledgment based protocol is stable for contention resolution with multiple servers and such high request rates.

One application of our result is the following: When N processors are connected via a complete optical network (as in the OCPC model [AM88, RT94]), the resulting communication system consists of N clients and N servers. Thus, the special case of our result in which the number of clients is equal to the number of servers shows that if the sum of the request rates associated with a given processor is less than 1 then any superlinear polynomial backoff protocol can be used to route the messages.

2 The Protocol

There are many ways to generalize the ethernet backoff protocol to a multiple server protocol. We consider the following generalization, which is natural (and perhaps easiest to analyze).

We have N clients and K servers. For each client i and each server j we have a queue $Q_{i,j}$ which contains the messages that the client i has to send to server j. We use the notation $q_{i,j,t}$ to denote the length of $Q_{i,j}$ before step t. ($q_{i,j,1} = 0$.) We define a backoff counter whose value before step t is $b_{i,j,t}$ ($b_{i,j,1} = 0$). The protocol at step t is as follows. With probability $\lambda_{i,j}$, a message arrives at $Q_{i,j}$ at step t. If a message arrives and $q_{i,j,t} = 0$ then $Q_{i,j}$ decides to send on step t. If $q_{i,j,t} > 0$ then $Q_{i,j}$ decides to send on step t only if it previously decided to retransmit on step t. If client i has exactly one queue that decides to send, it sends a message from that queue (otherwise, it does not send any messages). After step t, the variables $q_{i,j,t+1}$ are set to be the new queue lengths. If $Q_{i,j}$ decided to send on step t but it was not successful (i.e., either client i did not actually send the message, or more than one message was sent to server j (we refer to either of these events as a *collision* at queue $Q_{i,j}$)), then it sets $b_{i,j,t+1}$ to $b_{i,j,t} + 1$ and it chooses an integer ℓ uniformly at random from $\{1, \ldots, \lfloor (b_{i,j,t+1} + 1)^{\alpha} \rfloor\}$ and it decides to retransmit on step $t + \ell$. If $Q_{i,j}$ successfully sent on step t then it sets $b_{i,j,t+1}$ to be 0.

[1] Although standard protocols are used in practice, the system that arises in practice is more complicated than the one that we study because of issues such as message length, synchronization and so on. See [HLR93] for the details.

In order to simplify the analysis of the above protocol, we use the following equivalent formulation: For each queue $Q_{i,j}$, we also define a step counter whose value before step t is $s_{i,j,t}$ ($s_{i,j,1} = 1$). Then in the new formulation of the protocol, if $q_{i,j,t} > 0$ then $Q_{i,j}$ decides to send on step t with probability $s_{i,j,t}^{-1}$. (This decision is made independently of other decisions.) After step t, the step counters are updated as follows. If $q_{i,j,t} > 0$ but $Q_{i,j}$ did not decide to send on step t then $s_{i,j,t+1}$ is set to $s_{i,j,t} - 1$. If $Q_{i,j}$ decided to send on step t but it was not successful then it sets $s_{i,j,t+1}$ to $\lfloor (b_{i,j,t+1} + 1)^{\alpha} \rfloor$. If $Q_{i,j}$ successfully sent on step t then it sets $s_{i,j,t+1}$ to be 1. (To see that this formulation is equivalent, note that the probability that $Q_{i,j}$ retransmits on a step t' in the range $t+1, \ldots, t + \lfloor (b_{i,j,t+1} + 1)^{\alpha} \rfloor$ after a collision at step t is $1/\lfloor (b_{i,j,t+1} + 1)^{\alpha} \rfloor$. Thus, each step in the range is equally likely to be chosen.)

3 The Proof of Stability

Following [HLR93], assume that the system starts in the initial state in which there are no messages waiting in the buffers and let T_{ret} be the number of steps until the system returns to this state. Let L_i be the number of messages in the system after step i, and let $L_{\text{avg}} = \lim_{n \to \infty} (1/n) \sum_{i=1}^{n} L_i$. Let W_{avg} denote the average waiting time incurred by a message before it is successfully delivered. Recall that a contention-resolution protocol is stable for a given set of request rates if $\text{Ex}[W_{\text{avg}}]$ and $\text{Ex}[T_{\text{ret}}]$ are finite when the system is run with those request rates. By a result of Stidham [Sti74], the fact that $\text{Ex}[W_{\text{avg}}]$ is finite follows from the fact that $\text{Ex}[L_{\text{avg}}]$ is finite.

The main result of our paper is that the protocol described in Section 2 is stable as long as $\alpha > 1$ and the system has a sub-unit client-server request rate. The condition that the system have a sub-unit client-server request rate is necessary in a single-server system. For the worst case multiple-server system (a system with the same number of clients and servers), the condition may reduce the usable bandwidth by up to a factor of 2.

The starting point for our proof is the proof of [HLR93], so we begin by briefly describing their proof. We use the notation of [HLR93] in our proof whenever it is possible to do so.

3.1 The Stability Proof of Håstad, Leighton and Rogoff

The proof of [HLR93] analyzes the behavior of a Markov chain which models the single-server system. The current state of the chain contains the current queue lengths and backoff counters for all of the clients. The probabilities of transitions in the chain are defined by the protocol. The authors define a *potential function* which assigns a potential to each state in the chain. If the chain is in state s just before step t, the potential of state s is defined to be

$$\text{POT}(s) = \sum_{i=1}^{N} q_{i,t} + \sum_{i=1}^{N} (b_{i,t} + 1)^{\alpha+1/2} - N.$$

The potential function is used to prove that $\text{Ex}[T_{\text{ret}}]$ and $\text{Ex}[L_{\text{avg}}]$ are finite.

The proof in [HLR93] has two parts. The bulk of the proof establishes the fact that there are constants δ, d and V such that for any state s with potential at least V, there is a tree of depth at most d of descendant states over which the decrease in the square of the potential is at least $\delta \text{POT}(s)$. The proof of this fact has three cases.

1. If state s contains a queue Q_i that will send and succeed with overwhelming probability, then the authors consider the complete tree of depth 1, and show that the expected decrease in the square of the potential is sufficiently large.

2. Otherwise, if state s contains a queue Q_i with a big backoff counter then the tree that they consider is the complete tree of depth 1 or 2. Since the backoff counter of Q_i is big, the potential decreases significantly if Q_i succeeds in sending a message. They show that this happens with sufficiently high probability that the expected decrease in the square of the potential is sufficiently large.
3. In the remaining case, they show that with reasonably high probability, a long queue (which we call the *control* queue) takes over and dominates the server for a long time, sending many messages. Specifically, the tree that they consider consists of long paths in which the control queue dominates the server (the potential decreases significantly on these paths) and of short branches off of the long paths in which something goes wrong and the control queue loses control. The potential may increase on these short branching paths. However, it turns out that it does not increase too much, so over the tree, the expected decrease in the square of the potential is sufficiently large.

The second (easier) part of their proof shows that, given the fact that each state with sufficiently large potential has a tree as described above, $\mathrm{Ex}[L_{\mathrm{avg}}]$ and $\mathrm{Ex}[T_{\mathrm{ret}}]$ are finite.

3.2 Overview of our Stability Proof and Comparison to The Proof of Håstad et al.

Following [HLR93], we view our protocol as being a Markov chain in which states are $3KN$-tuples containing the queue lengths, backoff counters and step counters associated with each queue. The transition probabilities between the states are defined by the protocol. This Markov chain is easily seen to be time invariant, irreducible, and aperiodic. We use a potential function argument to show that $\mathrm{Ex}[T_{\mathrm{ret}}]$ and $\mathrm{Ex}[L_{\mathrm{avg}}]$ are finite. In order to show that $\mathrm{Ex}[L_{\mathrm{avg}}]$ is finite we show that the expected average potential is bounded. According to our potential function, each state just before step t has the following potential associated with it.

$$\mathrm{POT}_t = \sum_{i=1}^{N}\sum_{j=1}^{K}[q_{i,j,t} + (b_{i,j,t}+1)^{\alpha+\frac{1}{2}} - s_{i,j,t}^{1-\frac{1}{4\alpha}}].$$

The use of step counters in our potential function is motivated by the following problem (which we describe in the single server case). Suppose that s is a state with two queues Q_1 and Q_2 that have step counters equal to 1, but huge backoff counters. In this case, with probability 1, Q_1 and Q_2 collide on this step, and increase their backoff counters. If the potential function of [HLR93] were used, this would cause a massive increase in potential. This is not the case with our potential function.

Our proof is structurally similar to that of [HLR93] in that we first show that for every state s with $\mathrm{POT}(s) \geq V$ there is a tree of depth at most $V-1$ rooted at s such that the expected decrease in the square of the potential over the tree is at least $\mathrm{POT}(s)$ and from this we prove that $\mathrm{Ex}[T_{\mathrm{ret}}]$ and $\mathrm{Ex}[L_{\mathrm{avg}}]$ are finite. Our proof of the first part is broken up into cases. However, we do not use the same cases as [HLR93]. For instance, our potential function prevents us from considering the first case of [HLR93] in which a single queue sends and succeeds with overwhelming probability. The problem is that this single queue only reduces the potential by 1, whereas the step counters of the other queues cause a larger increase in potential.

The first case that we consider is the case in which every backoff counter in s is small. Suppose that $Q_{1,1}$ is the longest queue in s. (We call $Q_{1,1}$ the *control* queue.) In the single-server case, [HLR93] finds a tree of depth U rooted at s such that with reasonably high probability, $Q_{1,1}$ sends successfully on most of the U steps. When this occurs, the potential goes down because almost U

messages are sent whereas at most λU messages are received. (The tree is defined in such a way that the backoff counters, which start small, do not increase the potential by too much)

In the K-server case this approach does not suffice. First of all, it could be the case that almost all of the messages start at queue $Q_{1,1}$, so it is the only queue that can dominate a server during the U steps. However, even though $Q_{1,1}$ sends a message on most of the U steps, about $K\lambda U$ messages are received on the U steps, so the potential increases (assuming $K > \lambda^{-1}$). One possible solution to this problem involves modifying the potential function to give different "weights" to messages depending upon the distribution of queue sizes or backoff counters. However, this solution seems to cause other difficult problems, and thus does not seem to help.

Our solution to the problem is approximately as follows. We define a tree of descendant states of depth U such that with reasonably high probability $Q_{1,1}$ successfully sends on most of the U steps, and the part of the potential that is attributed to client 1 and server 1 goes down. Next, we wish to prove that the part of the potential that is attributed to the queues that do not have client 1 or server 1 (we refer to these queues as *free queues*) does not go up too much over the tree. This problem is complicated by the fact that the free queues interact with the other queues as the Markov chain runs, so there are dependence issues. In order to deal with the dependence of the control queue and the other queues with client or server 1 on the free queues, we let $\mathcal{M}$ denote the Markov chain that describes our protocol and we define several Markov chains that are similar to $\mathcal{M}$ but do not depend upon the behavior of the free queues. Next, we define the states in our tree in terms of the chains that are similar to $\mathcal{M}$ rather than in terms of $\mathcal{M}$ itself. We prove that $\mathcal{M}$ is related to the other chains, and we use this fact to prove that we still expect the potential that is attributed to client 1 and server 1 to decrease over the tree. We now wish to prove that we do not expect the potential of the free queues to go up much over the tree. The definition of the tree has nothing to do with behavior of the free queues, so the problem is equivalent to finding an upper bound on the expected potential of the free queues at a given step, t. In order to find such a bound, we have to deal with dependences because the queues that are not free can affect the behavior of the free queues. If we (temporarily) ignore the dependences by pretending that the free queues are not disturbed by the other queues, our problem reduces to bounding the expected potential of a smaller client-server system at step t. To deal with the dependence, we define a stochastic process which is a Markov chain extended by certain "interrupt steps". We show that even with the interrupt steps, the expected potential of the free queues does not increase too much by step t. The details are given in Case 1 of our proof.

Cases 2 and 3 of our proof are similar to cases in the proof of [HLR93]. In both cases, s contains a backoff counter that is sufficiently large such that, with sufficiently high probability, the queue with the large backoff counter sends and succeeds and decreases the square of the potential.

Our fourth (and final) case is motivated by a problem that can occur when s has a queue $Q_{i,j}$ with a big backoff counter. In the single-server case, [HLR93] either finds a queue $Q_{i',j'}$ that will successfully send with overwhelming probability, or shows that with sufficiently high probability $Q_{i,j}$ sends successfully within 1 or 2 steps (as in our Cases 2 and 3). As discussed above, even if $Q_{i',j'}$ sends successfully, the potential may not decrease. However, $Q_{i',j'}$ might prevent $Q_{i,j}$ from sending successfully. Thus, the approach of [HLR93] does not suffice in the multiple-server case. We solve this problem by showing that unless it is sufficiently likely that $Q_{i,j}$ (or some other queue with a big backoff counter) sends successfully within some reasonable number of steps (in which case we are in Case 2 or 3), we can identify a control queue that dominates its server as in Case 1. This does not suffice, however, because there may be free queues with big backoff counters. Although we can guarantee that at any given step t the expected potential of the free queues does not increase too much (even if they have large backoff counters), we do not know of a way to guarantee that at any given step t the expected square of the potential does not increase too much in this case. We

solve this problem by identifying several control queues rather than just one, so that the free queues never have big backoff counters. Unfortunately, we cannot ensure that all of our control queues decide to send at the beginning of our tree. In order to make sure that the potential goes down, we must make sure that with reasonably high probability these *delayed* control queues succeed whenever they finally do send. (Otherwise, they may never send again and the potential would go up.) To ensure this, we identify temporary control queues which dominate their servers for a while, blocking any queues that may send messages which collide with the messages of the delayed control queues. After a temporary control queue stops being a control queue it becomes a free queue. Thus, we also have *delayed* free queues and we have to argue about the increase in the square of the potential of the delayed free queues as well as that of the ordinary free queues. This situation is described in Case 4 of our proof.

3.3 Preliminaries

Fact 3.1 *Given $r \geq 1$, and given $x, y \geq 0$ where $|x/y| < 1$, $(x+y)^r \leq y^r + \lceil r \rceil^{\lceil r+1 \rceil} y^{r-1} x$*

Proof: The quantity $\binom{r}{k}$ is defined as follows:

$$\binom{r}{k} = \begin{cases} \frac{r(r-1)\cdots(r-k+1)}{k(k-1)\cdots 1}, & \text{integer } k \geq 0; \\ 0, & \text{integer } k < 0. \end{cases}$$

The Binomial theorem says that if $|x/y| < 1$ then $(x+y)^r = \sum_k \binom{r}{k} x^k y^{r-k}$. We use the following observations to bound the sum.

1. If $k > r$ then $|\binom{r}{k} x^k y^{r-k}| \geq |\binom{r}{k+1} x^{k+1} y^{r-(k+1)}|$.
2. If r is not an integer then for any odd positive integer i, $\binom{r}{\lceil r \rceil + i} < 0$ and $\binom{r}{\lceil r \rceil + i + 1} > 0$.

Thus, $(x+y)^r \leq \sum_{k=0}^{\lceil r \rceil} \binom{r}{k} x^k y^{r-k}$. This quantity is at most

$$y^r + x y^{r-1} \sum_{k=1}^{\lceil r \rceil} \binom{r}{k}$$

which is at most $y^r + x y^{r-1} \lceil r \rceil^{\lceil r+1 \rceil}$. □

3.4 Lemmas about Markov Chains

In the following lemma, $\alpha > 1$ is a constant, and we assume U is large enough so that the analysis holds.

Lemma 3.1 *Let c be a sufficiently large constant. Consider a Markov chain with states corresponding to pairs of positive integers and transitions from (i,j) to $(i, j-1)$ with probability $1 - \frac{1}{j}$ and from (i,j) to $(i+1, \lfloor (i+1)^\alpha \rfloor)$ with probability $\frac{1}{j}$. If the initial state is (b_1, s_1) with $s_1 \leq b_1^\alpha$ and $t \leq U$ steps are taken, then with probability greater than $1 - O((\log U)^{-1})$ the state (b_2, s_2) reached at step t satisfies $b_2^{\alpha+\frac{1}{2}} - s_2^{1-\frac{1}{4\alpha}} - (b_1^{\alpha+\frac{1}{2}} - s_1^{1-\frac{1}{4\alpha}}) \leq c\, U^{1-\frac{1}{4(\alpha+1)}}$.*

Proof: Note that

$$b_2^{\alpha+\frac{1}{2}} - s_2^{1-\frac{1}{4\alpha}} - (b_1^{\alpha+\frac{1}{2}} - s_1^{1-\frac{1}{4\alpha}}) = (b_2^{\alpha+\frac{1}{2}} - b_1^{\alpha+\frac{1}{2}}) + (s_1^{1-\frac{1}{4\alpha}} - s_2^{1-\frac{1}{4\alpha}}),$$

and that when at most U steps are taken, either $s_1 \leq s_2$ in which case $s_1^{1-\frac{1}{4\alpha}} - s_2^{1-\frac{1}{4\alpha}} \leq 0$ or $s_1 > s_2$ in which case

$$(s_1^{1-\frac{1}{4\alpha}} - s_2^{1-\frac{1}{4\alpha}}) \leq (s_1 - s_2)^{1-\frac{1}{4\alpha}} \leq U^{1-\frac{1}{4\alpha}}.$$

So in general, we simply need to show that $b_2^{\alpha+\frac{1}{2}} - b_1^{\alpha+\frac{1}{2}} \leq O(U^{1-\frac{1}{4(\alpha+1)}})$.

For a given state (i,j), we say i is the *level* of the state. We proceed in three cases.

Case 1: $b_1 < U^{1/(\alpha+1)}$

For the first $\frac{1}{3}U^{\frac{\alpha}{\alpha+1}}$ steps after one reaches a level of at least $U^{\frac{1}{\alpha+1}}$, the probability of another increase in level is at most $2U^{-\frac{\alpha}{\alpha+1}}$. Then in U of these steps, the expected number of increases in level is at most $2U^{1-\frac{\alpha}{\alpha+1}} = 2U^{\frac{1}{\alpha+1}}$. Using a Chernoff bound, the probability of over twice that many is at most $2^{-\Omega(U^{1/(\alpha+1)})} \leq O((\log U)^{-1})$. Also, the number of increases at other steps after one reaches a level of at least $U^{\frac{1}{\alpha+1}}$ can be at most $3U^{1-\frac{\alpha}{\alpha+1}} = 3U^{\frac{1}{\alpha+1}}$, since there are at least $\frac{1}{3}U^{\frac{\alpha}{\alpha+1}}$ steps between any of those steps. Thus with probability $1 - O((\log U)^{-1})$, $b_2 < 8U^{\frac{1}{\alpha+1}}$, and thus

$$b_2^{\alpha+\frac{1}{2}} - b_1^{\alpha+\frac{1}{2}} \leq O(U^{1-\frac{1}{2(\alpha+1)}}).$$

Case 2: $b_1 \geq U^{1/\alpha}\log U$

If $s_1 \geq \frac{1}{2}U(\log U)^\alpha$, then the probability of any increase in level at any of U steps is at most $U(4U^{-1}(\log U)^{-\alpha}) \leq O((\log U)^{-1})$. If there is no increase in level, then $b_2 = b_1$.

If $s_1 < \frac{1}{2}U(\log U)^\alpha$. then there might be a large possibility of an increase in level. If this increase occurs, we are essentially in the situation above, so with probability at least $1 - O((\log U)^{-1})$, there will be no further increases in level. Then $b_2^{\alpha+1/2} - b_1^{\alpha+1/2}$ is bounded by $O(b_1^{\alpha-1/2})$, but

$$\begin{aligned} s_2^{1-\frac{1}{4\alpha}} - s_1^{1-\frac{1}{4\alpha}} &\geq (\lfloor (b_1+2)^\alpha \rfloor - U)^{1-\frac{1}{4\alpha}} - (\frac{1}{2}U(\log U)^\alpha)^{1-\frac{1}{4\alpha}} \\ &\geq (b_1+1)^{\alpha-\frac{1}{4}} - U^{1-\frac{1}{4\alpha}} - (\frac{1}{2}U(\log U)^\alpha)^{1-\frac{1}{4\alpha}} \\ &\geq \frac{1}{4}b_1^{\alpha-\frac{1}{4}} \end{aligned}$$

Thus

$$b_2^{\alpha+\frac{1}{2}} - s_2^{1-\frac{1}{4\alpha}} - (b_1^{\alpha+\frac{1}{2}} - s_1^{1-\frac{1}{4\alpha}}) \leq O(1).$$

Finally, if there is no increase in level, then $b_2 = b_1$.

Case 3: $U^{1/(\alpha+1)} < b_1 < U^{1/(\alpha)}\log U$

Using a Chernoff bound (similar to Case 1), we can show that with probability at least $1 - O((\log U)^{-1})$, there will be at most $O(\max\{Ub_1^{-\alpha}, \log\log U\})$ increases in levels in at most U steps. Using Fact 3.1, we see that if $Ub_1^{-\alpha} \leq \log\log U$ then $b_2^{\alpha+1/2} - b_1^{\alpha+1/2}$ is bounded by

$$O(b_1^{\alpha-1/2}\log\log U) \leq O(U^{1-1/(2\alpha)}(\log U)^{\alpha+1/2}\log\log U) \leq O(U^{1-1/(2(\alpha+1))}).$$

Similarly, if $\log\log U \leq Ub_1^{-\alpha}$ then $b_2^{\alpha+1/2} - b_1^{\alpha+1/2}$ is bounded by

$$O(b_1^{\alpha-1/2}Ub_1^{-\alpha}) \leq O(Ub_1^{-1/2}) \leq O(U^{1-1/(2(\alpha+1))}).$$

□

Let f be the function defined by $f(x) = \lceil x + \frac{1}{2} \rceil^{8\cdot\lceil x+\frac{3}{2} \rceil}$.

Lemma 3.2 *Consider a Markov chain with states corresponding to pairs of positive integers and transitions from (i,j) to $(i,j-1)$ with probability $1-\frac{1}{j}$ and from (i,j) to $(i+1,\lfloor(i+1)^\alpha\rfloor)$ with probability $\frac{1}{j}$. If the initial state is (b_1,s_1) with $s_1 \le b_1^\alpha$, and $t < ((b_1+1)/f(\alpha))^{1/8}$ steps are taken, then any state (b_2,s_2) reached at step t satisfies $b_2^{\alpha+\frac{1}{2}} - s_2^{1-\frac{1}{4\alpha}} - (b_1^{\alpha+\frac{1}{2}} - s_1^{1-\frac{1}{4\alpha}}) \le s_1$.*

Proof: Let x be the number of transitions that cause an increase in level. If $x=0$ then $b_2=b_1$ so the quantity has an upper bound of s_1. Otherwise, the quantity is at most

$$(b_1+t)^{\alpha+\frac{1}{2}} - (\lfloor(b_1+1)^\alpha\rfloor - t)^{1-\frac{1}{4\alpha}} - b_1^{\alpha+\frac{1}{2}} + s_1^{1-\frac{1}{4\alpha}}.$$

Now, $(\lfloor(b_1+1)^\alpha\rfloor - t)^{1-\frac{1}{4\alpha}}$ is at least $((b_1+1)^\alpha - (b_1+1)^{1/8})^{1-\frac{1}{4\alpha}}$ which is at least $(b_1+1)^{(\alpha-1/8)(1-\frac{1}{4\alpha})}$ which is at least $(b_1+1)^{(\alpha-1/2)}(b_1+1)^{(1/8)}$. We can use the bound on t in the statement of the lemma to show that this is at least $(b_1+1)^{(\alpha-1/2)}t^{\lceil\alpha+\frac{1}{2}\rceil^{\lceil\alpha+\frac{3}{2}\rceil}}$.

By Fact 3.1, $(b_1+t)^{\alpha+\frac{1}{2}} - b_1^{\alpha+\frac{1}{2}}$ is at most $b_1{}^{(\alpha-1/2)}t^{\lceil\alpha+\frac{1}{2}\rceil^{\lceil\alpha+\frac{3}{2}\rceil}}$. The bound follows. □

Corollary 3.1 *Consider a Markov chain with states corresponding to pairs of positive integers and transitions from (i,j) to $(i,j-1)$ with probability $1-\frac{1}{j}$ and from (i,j) to $(i+1,\lfloor(i+1)^\alpha\rfloor)$ with probability $\frac{1}{j}$. If the initial state is (b_1,s_1) with $s_1 \le b_1^\alpha$, and 1 step is taken, then any state (b_2,s_2) reached at step t satisfies $b_2^{\alpha+\frac{1}{2}} - s_2^{1-\frac{1}{4\alpha}} - (b_1^{\alpha+\frac{1}{2}} - s_1^{1-\frac{1}{4\alpha}}) \le s_1 + f(\alpha)^{\alpha+1/2}$.*

Lemma 3.3 *Consider a Markov chain with states corresponding to pairs of positive integers and transitions from (i,j) to $(i,j-1)$ with probability $1-\frac{1}{j}$ and from (i,j) to $(i+1,\lfloor(i+1)^\alpha\rfloor)$ with probability $\frac{1}{j}$. If the initial state is (b_1,s_1) with $s_1 \le b_1^\alpha$, and (b_2,s_2) denotes the state after one step is taken, and B^+ denotes $b_2^{\alpha+\frac{1}{2}} - s_2^{1-\frac{1}{4\alpha}} - (b_1^{\alpha+\frac{1}{2}} - s_1^{1-\frac{1}{4\alpha}})$, then $\mathrm{Ex}(B^+) \le 2f(\alpha)^{\alpha+1/2}$.*

Proof: $B^+ \le (b_1+1)^{\alpha+\frac{1}{2}} + b_1^{\alpha(1-\frac{1}{4\alpha})}$. If $(b_1+1) \le f(\alpha)$, this is at most $f(\alpha)^{\alpha+\frac{1}{2}} + f(\alpha)^\alpha$. Otherwise, we use Lemma 3.2 to show that, when the level increases, B^+ is at most s_1. The probability that the level increases is $1/s_1$. If the level does not increase, then B^+ is at most $-(s_1-1)^{1-\frac{1}{4\alpha}} + s_1^{1-\frac{1}{4\alpha}}$ which is at most 1. □

A natural concept about Markov Chains we use is that of a tree of descendent states from a given state s. Let the root node be $((s),t_0)$. Now for each node $((s,r_1,r_2,\ldots,r),i)$ at level i, and for each transition $r \to r'$ in the Markov chain, let $((s,r_1,r_2,\ldots,r,r'),i+1)$ be a child of that node. When there is no confusion, we often refer to a node simply by the last state in its list of states. Assuming there is a potential function defined on the states of the Markov chain, we define the potential of a node to be the potential of the last state in its list.

We say a Markov Chain with non-negative potentials assigned to each state is V-good if it satisfies the following properties.

1. If a state s has potential $\mathrm{POT}(s) \ge V$ then there is a tree of depth at most $V-1$ rooted at s such that the expected decrease in the square of the potential over the tree is at least $\mathrm{POT}(s)$.
2. For any state s with $\mathrm{POT}(s) \le V$, every transition from s is to a state with potential at most $2V$.
3. The number of states with potential less than $2V$ is at most 2^V.

4. From each state s with $\mathrm{POT}(s) \leq V$ we can define a canonical path of length at most $2V$ to the unique state with potential 0 such that when the chain starts at s the probability that the path is taken is at least 2^{-V^3}.

Lemma 3.4 *Given a V-good Markov chain, let $T_{\mathrm{retV}}(s)$ denote the first step during which the potential is at most V at the start of the step, given that the chain starts at s. (If $\mathrm{POT}(s) \leq V$ then $T_{\mathrm{retV}}(s) = 1$.) Then starting at any state s,*

$$\mathrm{Ex}\left[\sum_{t=1}^{T_{\mathrm{retV}}(s)} \mathrm{POT}_t\right] \leq 2^V(\mathrm{POT}(s))^2,$$

Proof: (We model this proof after that in [HLR93].) As in [HLR93], since $T_{\mathrm{retV}}(s)$ might be infinite, a priori, we define a modified system that is terminated after T steps, meaning the system goes to the unique state of potential 0 at step T and stays there. We then prove

$$E(s,T) = \mathrm{Ex}\left[\sum_{t=1}^{\min(T,T_{\mathrm{retV}}(s))} \mathrm{POT}_t\right]$$

is bounded from above by $2^V\mathrm{POT}(s)^2$ by induction on T.

This is true for $T = 1$, since $\mathrm{POT}_1 = \mathrm{POT}(s)$. This is also true for any s with $\mathrm{POT}(s) \leq V$, since then $T_{\mathrm{retV}}(s) = 1$.

For the induction step, assume that $E(s,T') \leq 2^V\mathrm{POT}(s)^2$ for all $T' < T$ and any s with $\mathrm{POT}(s) \geq V$. We then bound $E(s,T)$ as follows.

Let the leaf s' of the tree of descendent states appear with probability $p_{s'}$, have potential $\mathrm{POT}(s')$ and be at depth $d_{s'}$. Let $\mathrm{POT}'(s')$ denote the sum of POT_t over the $d_{s'}$ steps taken to reach leaf s'. Since the potential can at most double at each step,

$$\mathrm{POT}'(s') \leq \mathrm{POT}(s)\sum_{j=0}^{d_{s'}} 2^j \leq 2^{d_{s'}+1}\mathrm{POT}(s).$$

Then following [HLR93], we can see that

$$\begin{aligned} E(s,T) &\leq \sum_{s'} p_{s'}(2^{d_{s'}+1}\mathrm{POT}(s) + E(s', T - d_{s'})) \\ &\leq 2^V\mathrm{POT}(s) + 2^V\sum_{s'} p_{s'}(\mathrm{POT}(s'))^2 \\ &\leq 2^V\mathrm{POT}(s) + 2^V\mathrm{POT}(s)^2 - 2^V\mathrm{POT}(s) \\ &= 2^V\mathrm{POT}(s)^2. \end{aligned}$$

As in [HLR93], this implies the lemma. □

Lemma 3.5 *Given a V-good Markov chain, if we start at state s with $\mathrm{POT}(s) \leq V$ then the expected potential at step t is at most $(2V)^2 2^{2V}$.*

Proof: For any state s', consider the partial tree of descendent states from s' in which, for every node, all proper ancestors of that node have potential greater than V. Let $S_t(s')$ be the set of nodes at level t of this tree. Let $E'(s',t) = \sum_{v \in S_t(s')} p_v \mathrm{POT}(v)$, where p_v is the probability of reaching

node v from s'. Let $E(s') = Ex[\sum_{i=1}^{T_{retV(s')}} \mathrm{POT}_i]$. Then $E(s') = \sum_t E'(s,t)$. By Lemma 3.4, $E(s') \leq 2^V(\mathrm{POT}(s))^2$. and thus $\sum_t E'(s',t) \leq 2^V(\mathrm{POT}(s))^2$.

Let $E(s,t)$ be the expected potential after t steps when starting in state s. We would like to prove for all t that when $\mathrm{POT}(s) \leq V$, $E(s,t) \leq (2V)^2 2^{2V}$. Let T be the full depth t tree of descendent states of s. Note $E(s,t)$ is the sum over leaves of this tree of the probability of reaching the leaf times the potential of that leaf. For any node $v \in T$, let d_v be the depth of v, and let p_v be the probability of reaching node v. Note that if Q is a set of nodes in T such that each leaf v in T has an ancestor v' in Q and every node on the path from v to v' has potential greater than V,

$$E(s,t) \leq \sum_{v' \in Q} E'(v', t - d_{v'}) p_{v'}.$$

Since the root of T has potential at most V, for every leaf v in T there is exactly one node $a(v)$ which is the closest ancestor to v whose parent has potential at most V. We let $Q = \{a(v) : v \text{ is a leaf of } T\}$, and note that it satisfies the conditions above.

Now we let $p_{s',i}$ be the probability of being in state s' at level i of T. (Note that this is the sum of probabilities of being in any node at level i of T with state s'.) Let S be the set of all states with potential at most $2V$. Then

$$\begin{aligned}
E(s,t) &\leq \sum_{v' \in Q} E'(v', t - d_{v'}) p_{v'} \\
&= \sum_{i=0}^{t} \sum_{s' \in S} E'(s', t-i) p_{s',i} \\
&\leq \sum_{s' \in S} \sum_{i=0}^{t} E'(s', t-i) \\
&= \sum_{s' \in S} \sum_{i=0}^{t} E'(s', i) \\
&\leq \sum_{s' \in S} 2^V (2V)^2 \\
&\leq (2V)^2 2^{2V}.
\end{aligned}$$

□

For the next lemma we extend a Markov chain with *interrupt steps*, which are steps in which we externally modify the transition probabilities of the chain. (Each step could modify the chain in a different way.) The timing and modification of these interrupt steps will be defined independently of the chain itself.

Lemma 3.6 *Consider a V-good Markov chain extended with a set of interrupt steps M, such that this extended Markov chain has the property that for any state s, the expected increase in potential in one step is at most z, whether or not the step is an interrupt step. If we start at state s then the expected potential at step t of this extended Markov chain is at most* $\mathrm{POT}(s) + (|M| + z)((2V)^2 2^{2V} + z)$.

Proof: We prove this result for every set M which has the property stated in the lemma, by induction on $|M|$. Let $E(s,t,M)$ be the expected potential after t steps when starting in state s with a set of interrupt steps M. For the base case, let $M = \emptyset$. We prove by induction on t that $E(s,t,\emptyset) \leq \mathrm{POT}(s) + (2V)^2 2^{2V}$. For $t \leq V$, $E(s,t,\emptyset) \leq \mathrm{POT}(s) + V$, and when $\mathrm{POT}(s) < V$, $E(s,t,\emptyset) \leq (2V)^2 2^{2V}$, by Lemma 3.5.

Now we must prove that the result holds for any s and t with $\mathrm{POT}(s) \geq V$ and $t > V$. We can assume that the result holds for all t' with $t' < t$. Since $\mathrm{POT}(s) \geq V$, we have a tree T of depth at most $V-1$ such that the expected change in potential is at most zero. Let S contain each leaf in T. For a leaf s' of T, let $p_{s'}$ be the probability of reaching that leaf, and let $d_{s'}$ be the distance of that leaf from the root. Now we have

$$\begin{aligned} E(s,t,\emptyset) &\leq \sum_{s' \in S} p_{s'} E(s', t - d_{s'}, \emptyset) \\ &\leq \sum_{s' \in S} p_{s'} (\mathrm{POT}(s') + (2V)^2 2^{2V}) \\ &\leq \mathrm{POT}(s) + (2V)^2 2^{2V}, \end{aligned}$$

Now that the base case for $|M| = 0$ is established, we need to prove the result for $|M| > 0$ assuming the result to be true for any t given that there are less than $|M|$ interrupt steps. (Note that we prove a slightly stronger result for the case $|M| = 0$ than for $|M| > 0$.)

Let t_1 be the time of the first interrupt step in M. Then $E(s, t_1 - 1, \emptyset) \leq \mathrm{POT}(s) + (2V)^2 2^{2V}$, and thus $E(s, t_1, \emptyset) \leq \mathrm{POT}(s) + (2V)^2 2^{2V} + z$, Now we examine the complete tree of states of depth t_1. Call this tree T, and let S be the set of leaves of T.

$$\begin{aligned} E(s,t,M) &\leq \sum_{s' \in S} p_{s'} E(s', t - t_1, M - \{t_1\}) \\ &\leq \sum_{s' \in S} p_{s'} (\mathrm{POT}(s') + (|M| - 1 + z)((2V)^2 2^{2V} + z)) \\ &\leq E(s, t_1, \emptyset) + (|M| - 1 + z)((2V)^2 2^{2V} + z) \\ &\leq \mathrm{POT}(s) + (|M| + z)((2V)^2 2^{2V} + z) \end{aligned}$$

□

Def: $\mathrm{POT}_{\mathrm{avg}} = \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} \mathrm{POT}_i$.

The following lemma is similar to one in [HLR93].

Lemma 3.7 *Given a V-good Markov chain,* $\mathrm{Ex}(\mathrm{POT}_{\mathrm{avg}}) \leq 2^{2V}(2V)^2$ *and* $\mathrm{Ex}(T_{\mathrm{ret}}) \leq (2^V(2V)^2 + 2V)2^{V^3}$.

Proof: Let s_0 be the state with potential zero. Let $T_{\mathrm{ret}}(s)$ be the number of steps taken to reach s_0 when starting from s. Let $p_i(s)$ be the probability of not reaching a state with potential at most V within i steps when starting in state s. (For convenience, let $p_{-1}(s) = 1$.) Then by Lemma 3.4, for any s with $\mathrm{POT}(s) \leq 2V$,

$$\mathrm{Ex}[T_{\mathrm{retV}}(s)] = \sum_{i \geq 0} p_i(s) \leq 2^V (2V)^2.$$

Let $Z = \max_{s:\mathrm{POT}(s) \leq V} \{\mathrm{Ex}[T_{\mathrm{ret}}(s)]\}$. We will determine an upper bound for Z.

From each state s with $\mathrm{POT}(s) \leq V$ we can define a canonical path of length at most $2V$ to s_0 such that when the chain starts at s the probability that the path is taken is at least 2^{-V^3}. If the path is not taken then the chain will make a transition from the path, ending in some state s' where $\mathrm{POT}(s') \leq 2V$. Let $p_{s \to s'}$ be the probability that s' is the first step off the canonical path from s to s_0. Let p be the probability that s goes to s_0 in the canonical way, and notice that this

will take at most V steps. Then for any s with $\mathrm{POT}(s) \leq V$,

$$\begin{aligned}
\mathrm{Ex}[T_{\mathrm{ret}}(s)] &\leq 2pV + \sum_{s'}[2V + \sum_{i\geq 0}(p_i(s') + Z(p_{i-1}(s') - p_i(s'))]p_{s\to s'} \\
&\leq 2pV + 2(1-p)V + \sum_{s'}\sum_{i\geq 0}(p_i(s') + Z(p_{i-1}(s') - p_i(s'))]p_{s\to s'} \\
&\leq 2pV + 2(1-p)V + \sum_{s'}[2^V(2V)^2 + Z]p_{s\to s'} \\
&\leq 2V + (1-p)[2^V(2V)^2 + Z].
\end{aligned}$$

Then we get $Z \leq 2V + (1-p)[2^V(2V)^2 + Z]$, from which we derive the bound $Z \leq p^{-1}[2V + 2^V(2V)^2]$. The result for $\mathrm{Ex}[T_{\mathrm{ret}}]$ follows by noting that $p \geq 2^{-V^3}$.

The bound on $\mathrm{Ex}[T_{\mathrm{ret}}]$ implies that the a V-good Markov chain is stationary. From Lemma 3.5, when starting from s_0, the expected potential at any step t is at most $2^{2V}(2V)^2$. Then we get

$$\begin{aligned}
\mathrm{Ex}[\mathrm{POT}_{\mathrm{avg}}] &= \mathrm{Ex}[\lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n}\mathrm{POT}_i] \\
&= \mathrm{Ex}[\liminf_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n}\mathrm{POT}_i] \\
&\leq \liminf_{n\to\infty}\mathrm{Ex}[\frac{1}{n}\sum_{i=1}^{n}\mathrm{POT}_i] \\
&= \liminf_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n}\mathrm{Ex}[\mathrm{POT}_i] \\
&\leq 2^{2V}(2V)^2.
\end{aligned}$$

The second equality relies on the fact that the limit exists with probability one (and an event with probability zero doesn't affect the expectation), which can be shown using the strong ergodic theorem for stationary processes . The first inequality comes from Fatou's Lemma since the random variables are always non-negative. □

3.5 The Proof

Now we are ready to prove stability of the N client, K server system as defined in the introduction.

Theorem 3.1 *Suppose that we have an N client, K server system and message bound $\lambda < 1$. Then there is a constant V such that the system corresponds to a V-good Markov chain.*

From Lemma 3.7 we get the following corollary.

Corollary 3.2 *Suppose that we have an N client, K server system and message bound λ. Then there is a constant V such that $\mathrm{Ex}(\mathrm{POT}_{\mathrm{avg}}) \leq 2^{2V}(2V)^2$ and $\mathrm{Ex}(T_{\mathrm{ret}}) \leq (2^V(2V)^2 + V)2^{V^3}$.*

Proof: [of Theorem 3.1] We proceed by induction on K. The case $K = 0$ is trivial with $V = 1$ so assume that the theorem holds for any K' server N' client system with $K' < K$ (more specifically, with constant $V_{K',N',\lambda}$). We will show that it holds for a K server N client system. That is, we must define a constant V such that the Markov chain is V-good. (Note that we only need to prove the Theorem holds for large N, since this will imply the Theorem for smaller N, using the same V.)

Given large enough V, Conditions 2 and 3 follow directly from the definition of the ethernet system. Condition 4 also follows directly from the definition. Suppose that s is a state with $\mathrm{POT}(s) \leq V$. The canonical path of length at most $2V$ from s to the unique state with potential 0 is defined as follows. First, no new messages arrive in the system during the walk on the path. Second, during the first V steps of the path, every non-empty queue decides to send. If there are still messages in the system after the first V steps then, during the remainder of the path, the queues take turns sending. (First, $Q_{1,1}$ sends until it is out of messages and then $Q_{1,2}$, and so on.) Since the system has at most V messages in state s, the path has at most $2V$ steps. The probability that no messages arrive is therefore at least $(1-\lambda)^{2KV}$. Since the backoff counters in state s are at most $2V-1$ the probability that every non-empty queue decides to send during the first V steps is at least $(4V)^{-\alpha VKN}$. The probability that the proper queue sends during the remaining steps is at least $(4V)^{-\alpha V}$. By the end of the first V steps, the step counter of every non-empty queue is at least V (actually, it is larger) Therefore, the probability that the other queues don't send during the remaining steps is at least 2^{-KNV}. Condition 4 follows.

The rest of this subsection proves that Condition 1 holds for a V which will depend on N, K, λ, and $V' = \max_{K'<K,N'<N} V_{K',N',\lambda}$. That is, we seek to prove that if a state s has potential $\mathrm{POT}(s) \geq V$ then there is a tree of depth at most $V-1$ rooted at s such that the expected decrease in the square of the potential over the tree is at least $\mathrm{POT}(s)$. In order to help the reader follow the proof, we note that the variables that we will use in the proof will satisfy the following inequality.

$$\frac{1}{1-\lambda}, \alpha, K, N, V' \leq Z \leq W \leq R \leq B \leq U \leq V.$$

We will assume in the proof that each variable is chosen to be sufficiently large with respect to the smaller variables. We will have $W = R/2$ and $Z = W^{1/(2\alpha)} - 2$.

Fix a state s with $\mathrm{POT}(s) \geq V$ and suppose that the Markov chain is in state s right before step t_0. We show that Condition 1 holds by splitting the analysis into cases, depending upon which (if any) of the following properties hold.

1. every backoff counter b_{i,j,t_0} is less than B,
2. there is a backoff counter $b_{i,j,t_0} \geq Z$ such that with probability at least

$$(1-\lambda)^{K5} 8^{-KN4} (b_{i,j,t_0}+5)^{-\alpha} 2^{-KN},$$

   queue $Q_{i,j}$ succeeds at least once during steps $t_0, \ldots, t_0+4$ and every other queue $Q_{i',j'}$ decides to send on step t (for $t \in \{t_0, \ldots, t_0+4\}$) only if $s_{i',j',t} \leq 8$.
3. there is a backoff counter $b_{i,j,t_0} \geq B$ such that with probability at least

$$(1-\lambda)^{K(4+R)} 8^{-KN4} (b_{i,j,t_0}+R+4)^{-\alpha} R^{-2\alpha KNR},$$

   queue $Q_{i,j}$ succeeds at least once during steps $t_0, \ldots, t_0+R+3$ and every other queue $Q_{i',j'}$ decides to send on step t (for $t \in \{t_0, \ldots, t_0+R+3\}$) only if $s_{i',j',t} \leq R^{2\alpha}$.

In our analysis we use the following random variables: We let $Q^+_{i,j}$ ($Q^-_{i,j}$) denote the increase (decrease) in potential due to the queue length of $Q_{i,j}$ over a path in the tree of descendent states. We let $B^+_{i,j}$ ($B^-_{i,j}$) denote the increase (decrease) in potential due to the combination of backoff counter and step counter for $Q_{i,j}$ over a path in the tree of descendent states. Then we let $Q^+ = \sum_{i=1}^{N} \sum_{j=1}^{K} Q^+_{i,j}$, and we define Q^-, B^+, and B^- analogously. We let δ denote the change in potential over a path in the tree of descendant states and we let Δ denote the change in the square of the potential over a path in the tree of descendant states.

We will use the following notation. Let $\rho_{i,j,t}$ and $\rho^*_{i,j,t}$ be random variables which are uniformly distributed over the unit interval. We can now describe our protocol in terms of these variables.

We will say that a message arrives at $Q_{i,j}$ at step t if $\rho_{i,j,t} \leq \lambda_{i,j}$. If $q_{i,j,t} > 0$ then $Q_{i,j}$ decides to send on step t if $\rho^*_{i,j,t} \leq s^{-1}_{i,j,t}$. The progress of the Markov chain describing our protocol (which we call $\mathcal{M}$) depends only the values of the ρ and ρ^* variables. Thus, the branching at depth t in our tree depends on the values of the random variables $\rho_{i,j,t}$ and $\rho^*_{i,j,t}$. In three of our cases, the states that we use for our tree are combinations of the states of Markov chains that are similar to $\mathcal{M}$ rather than states of $\mathcal{M}$ itself. (In Cases 2 and 3, all of the chains start in state s at step t_0 and run with the ρ and ρ^* values that are associated with the path in the tree. In Case 1, we argue about step t_0 separately, and the chains then start in a fixed state s' (dependent on s) at step $t_0 + 1$.) In order to define the states that we consider in our tree, we define, for every queue $Q_{i,j}$, a new Markov chain $\mathcal{M}_{i,j}$. In the chain $\mathcal{M}_{i,j}$, queue $Q_{i,j}$ follows the protocol, but all of the messages that it sends collide with messages sent by some external source. None of the other queues participate. We use the notation $q^+_{i,j,t}$, $b^+_{i,j,t}$ and $s^+_{i,j,t}$ to denote the queue lengths and counters when $\mathcal{M}_{i,j}$ is run. The progress of $\mathcal{M}_{i,j}$ is a function of the random variables $\rho_{i,j,t}$ and $\rho^*_{i,j,t}$. We let $B^{++}_{i,j}$ denote the increase in potential over a path in the tree due to the combination of the backoff counter and the step counter for $Q_{i,j}$ when $\mathcal{M}_{i,j}$ is run. (Since $B^{++}_{i,j}$ denotes an increase in potential when $\mathcal{M}_{i,j}$ is run (rather than when $\mathcal{M}$ is run), we use $q^+_{i,j,t}$, $b^+_{i,j,t}$ and $s^+_{i,j,t}$ in place of $q_{i,j,t}$, $b_{i,j,t}$ and $s_{i,j,t}$ in the potential function when we calculate $B^{++}_{i,j}$. At all other times, when we speak of the potential function, we mean the original potential function, which depends upon $q_{i,j,t}$, $b_{i,j,t}$ and $s_{i,j,t}$.) For every queue $Q_{d,d}$, we define a new Markov chain $\mathcal{M}_d$. In the chain $\mathcal{M}_d$, the queues in $\{Q_{i,j} \mid i = d \text{ or } j = d\}$ follow the protocol, but the other queues do not participate. We use the notation $q^d_{i,j,t}$, $b^d_{i,j,t}$ and $s^d_{i,j,t}$ to denote the queue lengths and counters when $\mathcal{M}_d$ is run. Note that if all of the chains are started at step t_0 then $q^+_{i,j,t_0+1} = q_{i,j,t_0+1} = q^d_{i,j,t_0+1}$. Similarly, each queue has the same initial counters for all three chains. Recall that in Case 1, we start the individual chains in a fixed state s' at step $t_0 + 1$. Thus, in Case 1 $q^+_{i,j,t_0+1} = q_{i,j,t_0+1} = q^d_{i,j,t_0+1}$. Similarly, at step $t_0 + 1$, each queue has the same counters for all three chains.

3.5.1 Case 1

Property 1 holds: When the Markov chain is started in state s right before step t_0 with $\text{POT}(s) \geq V$, every backoff counter b_{i,j,t_0} is less than B.

Without loss of generality, we assume that $Q_{1,1}$ is the largest queue in state s. We call $Q_{1,1}$ the *control queue* and any other queue with client or server 1 a *slave queue*. We call the other queues *free queues*. Recall that our goal is to show that there is a tree of depth at most $V - 1$ rooted at s such that the expected decrease in the square of the potential (over the tree) is at least $\text{POT}(s)$. We will let U denote the depth of this tree. (We will choose U such that $q_{1,1,t_0} \geq U$.) As we stated above, the branching in the tree depends upon the values of the ρ and ρ^* variables, so by fixing the values of the variables $\rho_{i,j,t}$ and $\rho^*_{i,j,t}$ for all i and j and all $t \leq t_0 + U - 1$ we fix a path p of length U. We define $\sigma(p)$ as follows: For every slave queue $Q_{i,j}$, and every step $t > t_0 + 4$, if $Q_{i,j}$ has $b^+_{i,j,t} \leq R^{1/\alpha}$ and it decides to send on step t in $\mathcal{M}_{i,j}$, then t is in $\sigma'_{i,j}(p)$. Let $\sigma'(p) = \{t \mid Q_{i,j} \text{ is a slave} \wedge t \in \sigma'_{i,j}(p)\}$. Let $\sigma(p) = \sigma'(p) \cup \{t + 1 \leq t_0 + U - 1 \mid t \in \sigma'(p)\}$. Let $\sigma_k(p)$ denote the kth step in σ. We say that path p is *good* if it satisfies the following conditions.

1. At step t_0, no message is received at any queue, the control queue decides to send in $\mathcal{M}$, every other queue $Q_{i,j}$ with $q^+_{i,j,t_0} > 0$ and $s^+_{i,j,t_0} \leq 5$ decides to send in $\mathcal{M}$, and no other queue $Q_{i,j}$ decides to send in $\mathcal{M}$.
2. At step $t_0 + 1$, no message is received at the control and slave queues, the control queue decides to send in $\mathcal{M}_1$, every slave queue $Q_{i,1}$ with $q^+_{i,1,t_0+1} > 0$ and $s^+_{i,1,t_0+1} \leq 4$ decides to send in $\mathcal{M}_{i,1}$, and no other slave queue $Q_{i,j}$ decides to send in $\mathcal{M}_{i,j}$.

At step t_0+2, no message is received at the control and slave queues, the control queue decides to send in $\mathcal{M}_1$, every slave queue $Q_{1,j}$ with $q^+_{1,j,t_0+2} > 0$ and $s^+_{1,j,t_0+2} \leq 3$ decides to send in $\mathcal{M}_{1,j}$, and no other slave queue $Q_{i,j}$ decides to send in $\mathcal{M}_{i,j}$.

At step t_0+3, no message is received at the control and slave queues, the control queue decides to send in $\mathcal{M}_1$, every slave queue $Q_{i,1}$ with $q^+_{i,1,t_0+3} > 0$ and $s^+_{i,1,t_0+3} \leq 3$ decides to send in $\mathcal{M}_{i,1}$, and no other queue $Q_{i,j}$ decides to send in $\mathcal{M}_{i,j}$.

3. At step t_0+4, no messages are received at the control and slave queues, the control queue decides to send in $\mathcal{M}_1$, and every slave queue $Q_{i,j}$ does not decide to send in $\mathcal{M}_{i,j}$
4. For each slave $Q_{1,j}$, and each $t \in \sigma'_{1,j}(p)$, $t \neq t_0 \bmod 2$. Also, for each slave $Q_{i,1}$, and each $t \in \sigma'_{i,1}(p)$, $t = t_0 \bmod 2$.
5. For every step $\sigma_k(p) \in \sigma(p)$, $\rho^*_{1,1,\sigma_k(p)} \leq (k+1)^{-\alpha}$.
6. If t is in $\sigma(p)$, and $Q_{i,j}$ is a slave queue with $b^+_{i,j,t} > R^{1/\alpha}$, then $Q_{i,j}$ does not decide to send on step t in $\mathcal{M}_{i,j}$. If t is not in $\sigma(p)$ and $Q_{i,j}$ is a slave queue with $b^+_{i,j,t} > R^{1/\alpha}$ which decides to send on step t in $\mathcal{M}_{i,j}$ then, for any t' in the range $t-2^\alpha-1,\ldots,t$, there is no slave queue $Q_{i',j'}$ with $b^+_{i',j',t'} > R^{1/\alpha}$ that decides to send on step t' in $\mathcal{M}_{i',j'}$.
7. If $Q_{i,j}$ is a slave or control queue then for every t in the range $t_0 \leq t < t_0+U$, $\rho^*_{i,j,t} > 2(U^{1/\alpha}\log(U))^{-\alpha}$.
8. For every slave queue $Q_{i,j}$ and any t in the range $t_0 \leq t \leq t_0+U$, we have $(b^+_{i,j,t}+1)^{\alpha+\frac{1}{2}} - {s^+_{i,j,t}}^{1-\frac{1}{4\alpha}} - ((b^+_{i,j,t_0}+1)^{\alpha+\frac{1}{2}} - {s^+_{i,j,t_0}}^{1-\frac{1}{4\alpha}}) \leq 2cU^{1-\frac{1}{4(\alpha+1)}}$ where c is the constant defined in Lemma 3.1
9. For any t in the range $t_0+5 \leq t < t_0+U$, the number of messages received by the control and slave queues during steps $t_0+5,\ldots,t$ is at most $\lambda(t-t_0-4)+U^{1/2}\log U$.

The tree that we consider will be the tree consisting of every good path of length U plus every child of every internal node of such a path. We will show that for this tree $\mathrm{Ex}[\Delta] \leq -\mathrm{POT}_{t_0}$. The key to showing this will be to prove that with sufficient probability a good path is taken when the chain is run. The properties in the definition of "good" deal with the Markov Chains $\mathcal{M}_d$ and $\mathcal{M}_{i,j}$. However, we will prove that in the internal nodes of our tree, the state of $\mathcal{M}$ is related to the states of $\mathcal{M}_d$ and $\mathcal{M}_{i,j}$. Thus, we will be able to show that for this tree $\mathrm{Ex}[\Delta] \leq -\mathrm{POT}_{t_0}$.

We start by proving a lemma which establishes some of the relationships between $\mathcal{M}$, $\mathcal{M}_1$ and $\mathcal{M}_{i,j}$.

Claim 3.1 *If n is a node in the tree at level $t \geq t_0+1$ (i.e., step t is just about to take place) and the parent of n is in good path p, then*

1. *For any slave queue $Q_{i,j}$, $q_{i,j,t} = q^+_{i,j,t} = q^1_{i,j,t}$, $b_{i,j,t} = b^+_{i,j,t} = b^1_{i,j,t}$ and $s_{i,j,t} = s^+_{i,j,t} = s^1_{i,j,t}$.*
2. *$q_{1,1,t} \leq q^1_{1,1,t}$, $b_{1,1,t} \leq b^1_{1,1,t}$ and $s_{1,1,t} \leq s^1_{1,1,t}$.*
3. *If $t > t_0+4$ then $b^1_{1,1,t} \leq 1 + |\sigma'(p) \cap \{t_0,\ldots,t-1\}|$.*

Proof: The proof is by induction on t. First note that the actions of every queue is forced at step t_0, and thus there is only one node s' at step t_0+1 in any good path. Then the base case includes the five steps $t \in \{t_0+1,\ldots,t_0+5\}$ The case $t = t_0+1$ is clear because the queue lengths and counters have the same values before step t_0+1 in all of the chains. To see that Item 1 holds for steps t_0+2 to t_0+4, note that any slave queue that decides to send collides with the control queue. Specifically, in step t_0+2, only queues that client-conflict with $Q_{1,1}$ decide to send, and in steps

t_0+1 and t_0+3 only queues that server-conflict with $Q_{1,1}$ decide to send, and in steps t_0+1, t_0+2 and t_0+3, $Q_{1,1}$ decides to send. To see that Item 1 holds for $t = t_0+5$ note that the slave queues do not decide to send on step t_0+4.

To see that Item 2 holds for steps t_0+2 through t_0+5 we consider the possible cases. In defining the cases, we observe that for any t, if Item 2 has been established for step $t-1$, then if $Q_{1,1}$ decides to send on step t in $\mathcal{M}_1$, $Q_{1,1}$ also decides to send on step t in $\mathcal{M}$. Also, if Item 1 and Item 2 have been established for step $t-1$, then if $Q_{1,1}$ sends and succeeds at step t in $\mathcal{M}_1$ then $Q_{1,1}$ sends and succeeds at step t in $\mathcal{M}$. (Note that Item 2 was established for step t_0+1.) The cases are:

a. $Q_{1,1}$ sends and succeeds in $\mathcal{M}$.

b. $Q_{1,1}$ decides to send in $\mathcal{M}$ and fails and does not decide to send in $\mathcal{M}_1$.

c. $Q_{1,1}$ decides to send and fails in $\mathcal{M}$ and $\mathcal{M}_1$.

d. $Q_{1,1}$ does not decide to send in $\mathcal{M}$ or $\mathcal{M}_1$.

It is clear that Item 2 holds in Cases a, c, and d. Finally, we note that Case b cannot occur on steps t_0+1 through t_0+4 since $Q_{1,1}$ decides to send on those steps in $\mathcal{M}_1$, and inductively Item 2 can be established for steps t_0+2 through t_0+5. To see that Item 3 holds for $t = t_0+5$, we note that $b^1_{1,1,t_0+5}$ is 0.

We now do the induction step. In order to establish Item 1, we want to show that if a slave queue $Q_{i,j}$ sends on step t (for $t > t_0+4$) then it collides in $\mathcal{M}$ and in $\mathcal{M}_1$. We consider two cases. If $t \in \sigma(p)$ (suppose that $t = \sigma_k(p)$), then, by property 4 and property 6, if $j = 1$, then no slave queue $Q_{1,j'}$ decides to send on step t. Whether or not $j = 1$, by property 5, $\rho^*_{1,1,t} \leq (k+1)^{-\alpha}$. Also, (by Item 3, inductively), $b^1_{1,1,t} \leq k$, so $s^1_{1,1,t} \leq (k+1)^{\alpha}$. By Item 2, inductively, $s_{1,1,t} \leq s^1_{1,1,t}$, so the control queue decides to send on step t in $\mathcal{M}$ and $\mathcal{M}_1$. Thus, $Q_{i,j}$ has a collision. Now suppose $t \notin \sigma(p)$. We will show that $b^1_{1,1,t} = 0$. (To do this, we can assume inductively that for $t' < t$, if $t' \notin \sigma(p)$, then $b^1_{1,1,t'} = 0$.) Consider the maximum $t' < t$ where either $t' \in \sigma(p)$, some slave sent at step t', or $t' = t_0+5$. If $t' \in \sigma(p)$, then $t'+1 \notin \sigma(p)$, so no slave queue sends at step t', but by property 5 and the argument used above, $Q_{1,1}$ sends and succeeds at step t' in $\mathcal{M}_1$. Thus $b^1_{1,1,t} = 0$. If $t' = t_0+5$, then $Q_{1,1}$ sends and succeeds at step t' in $\mathcal{M}_1$, and therefore $b^1_{1,1,t} = 0$. Otherwise, by property 6, $t' < t - 2^{\alpha}$, and inductively $b^1_{1,1,t'} = 0$, so $b^1_{1,1,t'+1} = 1$. But then $Q_{1,1}$ sends and succeeds by step $t-1$, so $b^1_{1,1,t} = 0$. Thus, (since the queue size of the control queue is at least U), it decides to send on step t and $Q_{i,j}$ has a collision.

In order to establish Item 2, we want to rule out Case b, in which $Q_{1,1}$ decides to send at step t in $\mathcal{M}$ and fails and does not decide to send in $\mathcal{M}_1$. We have already shown, in the analysis in the preceding paragraph, that $Q_{1,1}$ sends in $\mathcal{M}_1$ on every step in $\sigma(p)$. So suppose that $t \notin \sigma(p)$. By the same argument as in the preceding paragraph, we can show that $b^1_{1,1,t} = 0$ unless there is some $t' < t$, where some slave sends on step $t' \notin \sigma(p)$ and $t' \geq t - 2^{\alpha}$. So either no slave sends at step t, in which case $Q_{1,1}$ will succeed if it decides to send, or else some slave sends at step t, and by property 6, $b^1_{1,1,t} = 0$ (and thus $Q_{1,1}$ decides to send in $\mathcal{M}_1$). Thus, Case b does not occur.

In order to establish Item 3, we note that we have already shown that $Q_{1,1}$ sends in $\mathcal{M}_1$ on every step in $\sigma(p)$, and that it succeeds on the last step of every consecutive block of steps in $\sigma(p)$. Furthermore, $Q_{1,1}$ succeeds on step t_0+4. By property 6 (and, inductively, by Item 1), $Q_{1,1}$ can have at most 1 collision in $\mathcal{M}_1$ just before the consecutive block of steps from $\sigma(p)$. Item 3 follows. □

As in [HLR93], we will use the equality

$$\mathrm{Ex}[\Delta] = 2\mathrm{POT}_{t_0} \cdot \mathrm{Ex}[\delta] + \mathrm{Ex}[\delta^2].$$

Thus it is sufficient to show that $\mathrm{Ex}[\delta] \leq -1$ and $\mathrm{Ex}[\delta^2] \leq \mathrm{POT}_{t_0}$. Let E be the event that a good path is taken when the chains are run. (That is, E is the event that all the conditions in the

definition of "good" hold for U steps.) Let E_i be the event that condition i holds for U steps. Let $U' = \max\{U^{1/2}\log U, U^{1/\alpha}\log U, U^{1-1/(4(\alpha+1))}, U^{1-1/(4\alpha)}\log^{\alpha-1/4} U\}$, and note $U' = o(U)$.

Call two paths in the tree *equivalent* if and only if every queue has the same ρ and ρ^* values at step t_0, and every control and slave queue has the same sequence of ρ and ρ^* values over the the remaining transitions in the paths. This notion of equivalence is clearly an equivalence relation. Furthermore, if one path in the tree ends at level t (i.e., if $t < t_0 + U$, there is no good path continuing on from the node at level t, but there is a good path continuing on from the node at level $t-1$), then every equivalent path also ends at level t.

Let $\mathcal{M}'$ denote the Markov chain in which the free queues run the protocol (after step t_0), and no other queues participate. By induction, there is a constant V' such that $\mathcal{M}'$ is V'-good. Now suppose that we fix a sequence of ρ and ρ^* values for the control and slave queues and we run $\mathcal{M}$. If we just look at the free queues during this run, we can think of this as being a run of $\mathcal{M}'$, in which $\mathcal{M}'$ is extended by the set of interrupt steps I which is determined by the sequence of ρ and ρ^* values for the control and slave queues (and the ρ and ρ^* values of the queues at step t_0). Lemma 3.3 shows that when $\mathcal{M}'$ is extended by I, the expected increase in potential in any one step (other than step t_0) is at most $3KNf(\alpha)^{\alpha+1/2}$. (The expected increase due to each backoff and step counter is at most $2f(\alpha)^{\alpha+1/2}$ and the expected increase due to each queue is at most 1.) If the fixed sequence of ρ and ρ^* values is such that the path taken is in the tree (i.e, all of the properties continue to hold (except possibly after the last step)), then the number of interrupt steps in I is at most $KNU^{1/\alpha}\log(U)$. (To see this, note that, by Claim 3.1, every slave queue collides every time it sends (except possibly the last time it sends). Furthermore, since Property 7 holds, a slave queue does not send once its backoff counter is $U^{1/\alpha}\log(U) - 1$ (except possibly on the last step). Therefore, the slave queues provide at most $KNU^{1/\alpha}\log(U)$ interruptions.)

Claim 3.2 *Suppose that we fix a particular equivalence class of paths of length at least t, and we condition on the event that when $\mathcal{M}$ is run for t steps, starting with step t_0, one of the paths from this equivalence class is taken. Then the expected potential of the free queues, after the t steps, is at most the potential of the free queues at step t_0+1 plus* $((KNU^{1/\alpha}\log U) + 3KNf(\alpha)^{\alpha+1/2})((2V')^2 2^{2V'} + 3KNf(\alpha)^{\alpha+1/2})$.

Proof: We view the free queues as forming a Markov chain $\mathcal{M}'$ which is extended by the set of interrupts I that is determined by the set of ρ and ρ^* values associated with the equivalence class. We now apply Lemma 3.6 using the facts that the expected increase in potential in any one step is at most $3KNf(\alpha)^{\alpha+1/2}$ and the number of interruptions is at most $(KNU^{1/\alpha}\log U)$. □

Claim 3.3 *There is a function f_1 such that*

$$\mathrm{Ex}[\delta \mid E] \leq -(1-\lambda)U + U' \cdot f_1(\alpha, K, N, V', R).$$

Proof: Given E and Claim 3.1, there are at most

$$[(K+N-1)U^{1/\alpha}\log U]2^{\alpha}$$

steps on which the control queue does not broadcast successfully. (Each of the $K+N-1$ slave queues provides at most $U^{1/\alpha}\log U$ interrupts. For any run of interrupts in $\sigma(p)$, the control queue sends successfully after the last step of that run (which is still in $\sigma(p)$). For any interrupts not in $\sigma(p)$, the control queue sends within 2^{α} steps. Therefore, at least

$$U - [(K+N-1)U^{1/\alpha}\log(U)]2^{\alpha}$$

messages are sent successfully. By Property 9, the number of messages that are received by the control and slave queues is at most $\lambda U + U^{1/2}\log U$. By Property 8 and Claim 3.1, the increase in potential due to the backoff counters and step counters of the slave queues is at most $2cU^{1-\frac{1}{4(\alpha+1)}}$. By Claim 3.1, the backoff counter of the control queue is at most $(K+N-1)(R^{1/\alpha}+1)+1$, so the increase in potential due to the backoff counter and step counter of the control queue is at most $((K+N-1)(R^{1/\alpha}+1)+2)^{\alpha+1/2}$. Claim 3.2 shows that for each equivalence class of paths, the expected potential of the free queues increases by at most $((KNU^{1/\alpha}\log U) + 3KNf(\alpha)^{\alpha+1/2})((2V')^2 2^{2V'} + 3KNf(\alpha)^{\alpha+1/2})$ during steps $t_0+1,\ldots,t_0+U-1$. By Corollary 3.1, it increases by at most $KN(5+f(\alpha)^{\alpha+1/2})$ on step t_0. □

Claim 3.4

$$\Pr(E) \geq 2((1-\lambda)/3)^{5+(K+N-1)(R^{1/\alpha}+1)}5^{-5(KN-1)}(B+5)^{-5\alpha}((2(K+N-1)(R^{1/\alpha}+1)+1)!)^{-\alpha}.$$

Proof: We can divide the calculation as follows.

$$\Pr(E) = \Pr(E_1)\Pr(E_2 \mid E_1)\Pr(E_3 \mid E_1 \wedge E_2)\Pr(E_4 \mid E_1 \wedge E_2 \wedge E_3)\Pr(E_5 \mid \bigwedge_{i=1}^{4} E_i)\Pr(\bigwedge_{j=6}^{9} E_j \mid \bigwedge_{i=1}^{5} E_i).$$

Now we analyze each probability in turn. Clearly, $\Pr(E_1) \geq (1-\lambda)^K(B+1)^{-\alpha}5^{-(KN-1)}$. Note that every slave $Q_{1,j}$ with $q_{1,j,t_0+1} > 0$ has $s_{1,j,t_0+1} > 1$ and every slave $Q_{i,1}$ with $q_{i,1,t_0+2} > 0$ has $s_{i,1,t_0+2} > 1$ and every slave $Q_{1,j}$ with $q_{1,j,t_0+3} > 0$ has $s_{1,j,t_0+3} > 1$. Thus, $\Pr(E_2 \mid E_1) \geq (1-\lambda)^3(B+4)^{-3\alpha}4^{-3(K+N-1)}$. Note that every slave $Q_{i,j}$ has $s_{i,j,t_0+4} > 1$. Thus, $\Pr(E_3 \mid E_1 \wedge E_2) \geq (1-\lambda)(B+5)^{-\alpha}2^{-(K+N-1)}$.

The next probability essentially requires two separate arguments, one to lower bound the probability of each slave queue receiving its first message at either an odd or even step, and one to lower bound the probability of it attempting each send (until its backoff counter exceeds $R^{1/\alpha}$) at either an odd or even step.

In the first argument, note that we must show that $Q_{i,j}$ receives its first message at either an odd or even step. (If $\lambda_{i,j} = 0$, then $Q_{i,j}$ never receives any messages and we can disregard it.) If $\lambda_{i,j} \geq \frac{1}{2}$, then it receives a message at step t_0+5 (an odd step) with probability at least $\frac{1}{2}$, and it receives its first message at step t_0+6 with probability at least $\frac{1}{2}(1-\lambda_{i,j}) \geq \frac{1}{2}(1-\lambda)$. If $\lambda_{i,j} < \frac{1}{2}$, let π be the probability that a message is first received on an odd step (noting that the first step possible is t_0+5, an odd step). Then it can be easily shown that a message is first received on an even step with probability $(1-\lambda_{i,j})\pi$. Thus $\pi + (1-\lambda_{i,j})\pi = 1$, implying $\pi = (2-\lambda_{i,j})^{-1}$. Since $0 < \lambda_{i,j} < \frac{1}{2}$ and $\frac{1}{2} < \pi < \frac{2}{3}$, $(1-\lambda_{i,j})\pi > \frac{1}{4}$.

In the second argument, we must show that slave queue $Q_{i,j}$ attempts each send (until its backoff counter exceeds $R^{1/\alpha}$) at either an odd or even step, assuming that if it is empty, it attempts its first send at the correct step. First, we deal with the first step of the slave queues $Q_{i,j}$ with $q_{i,j,t_0+5} > 0$. If $Q_{1,j}$ is a slave queue with $q_{1,j,t_0+5} > 0$ and $s_{1,j,t_0+5} = 1$ then it sends on step t_0+5, which is fine. Every other slave queue $Q_{i,j}$ with $q_{i,j,t_0+5} > 0$ has $s_{i,j,t_0+5} \geq 2$. Since $s_{i,j,t_0+5} \geq 2$, there is a step of the correct parity in the range $t_0+5,\ldots,t_0+s_{i,j,t_0+5}+4$. Let t' be the last such step. The probability that $Q_{i,j}$ does not send before step t' is at least $1/s_{i,j,t_0+5}$ and the probability that it sends on step t', given that it did not send earlier, is at least $1/3$. We now consider steps after t_0+5. If $Q_{i,j}$ collides at step $t-1$, we have $s_{i,j,t} = \lfloor (b_{i,j,t}+1)^\alpha \rfloor$. By Claim 3.1, a slave queue never succeeds, so $s_{i,j,t} \geq 2$ and we can use the same argument that we used for the first step.

Thus, since the relevant step counters are at most $(R^{1/\alpha}+1)^\alpha$, we have

$$\Pr(E_4 \mid E_1 \wedge E_2 \wedge E_3) \geq \left(\frac{1}{4}(1-\lambda)\frac{1}{(R^{1/\alpha}+1)^\alpha}\right)^{(K+N-1)(R^{1/\alpha}+1)}.$$

Using Claim 3.1, we see that for any good path p, $|\sigma'(p)| \leq (K+N-1)(R^{1/\alpha}+1)$. Thus, $|\sigma(p)| \leq 2(K+N-1)(R^{1/\alpha}+1)$. We conclude that $\Pr(E_5 \mid \bigwedge_{i=1}^{4} E_i) \geq ((2(K+N-1)(R^{1/\alpha}+1)+1)!)^{-\alpha}$.

Next, we note that $\Pr(\bigwedge_{j=6}^{9} E_j \mid \bigwedge_{i=1}^{5} E_i)$ is at least

$$1 - \Pr(\overline{E_6} \mid \bigwedge_{i=1}^{5} E_i) - \Pr(\overline{E_7} \mid \bigwedge_{i=1}^{5} E_i) - \Pr(\overline{E_8} \mid \bigwedge_{i=1}^{5} E_i) - \Pr(\overline{E_9} \mid \bigwedge_{i=1}^{5} E_i)$$

We calculate $\Pr(\overline{E_6} \mid \bigwedge_{i=1}^{5} E_i)$, by considering the following game. Suppose that we have $U-2$ boxes, which are labeled $t_0+5, \ldots, t_0+U-1$. Each box will represent one time-step. Note that $\sigma(p)$ is completely determined by the values of the variables $\rho^*_{i,j,t}$ for slave queues $Q_{i,j}$ with $b^+_{i,j,t} \leq R^{1/\alpha}$. We look at these variables, and place a blank pebble in each box that represents a time-step in $\sigma(p)$. If slave queue $Q_{i,j}$ had $b^+_{i,j,t_0+5} > R^{1/\alpha}$ then we choose a random number ℓ between 1 and s_{i,j,t_0+5} and we put pebble $P_{i,j}$ in box $t_0+4+\ell$. (This choice of the random number is dependent upon the values $\rho^*_{i,j,t_0+5}, \rho^*_{i,j,t_0+6}, \ldots$) Otherwise, we use the values of the same ρ^* variables that we used to identify $\sigma(p)$ and we identify the integer t such that $b^+_{i,j,t} > R^{1/\alpha}$ and $b^+_{i,j,t-1} \leq R^{1/\alpha}$ and we put pebble $P_{i,j}$ in box $t-1$. To play the game we now consider the boxes in order. When we consider box t we check whether it contains a pebble, $P_{i,j}$. If so, we choose a random number ℓ between 1 and $\lfloor (b_{i,j,t+1}+1)^\alpha \rfloor$ and we put $P_{i,j}$ in box $t+\ell$. (This choice of the random number is dependent upon the values $\rho^*_{i,j,t+1}, \rho^*_{i,j,t+2}, \ldots$) We lose the game if any box other than box t_0+5 ever contains more than one pebble, or if any two boxes $t \neq t_0+5$ and $t' \neq t_0+5$ ever contain pebbles and have $|t-t'| \leq 2^\alpha$. Otherwise, we win. One can see that winning this game corresponds exactly to having condition E_4 hold. Note that the ρ^* values that we use to play the game are independent of the ρ^* values that we used to show that the probabilities that E_1-E_5 hold. The probability that $P_{i,j}$ causes a loss is at most the sum of

$$\frac{2(K+N-1)(R^{1/\alpha}+1)+(2^{\alpha+1}+1)(K+N-1)}{R}$$

(this is an upper bound on the probability that $P_{i,j}$ hits another pebble on its initial placement), and

$$\sum_{b \geq R^{1/\alpha}+1} \frac{2(K+N-1)(R^{1/\alpha}+1)+(2^{\alpha+1}+1)(K+N-1)}{\lfloor (b+1)^\alpha \rfloor}.$$

The sum is $O((K+N-1)R^{(1/\alpha)-1})$. Since there are at most $K+N-1$ pebbles, the probability of losing is $O((K+N-1)^2 R^{(1/\alpha)-1})$.

Let $H = \min(1/6, (B+4)^{-\alpha}, (2(K+N-1)(R^{1/\alpha}+1)+1)^{-\alpha})$. During the proof that $E1$-$E5$ hold with sufficiently high probability we sometimes forced $\rho_{i,j,t}$ values to be large. The only times that we forced $\rho_{i,j,t}$ values to be small, we only forced them to be as small as H. Thus,

$$\begin{aligned} \Pr(\overline{E_7} \mid \bigwedge_{i=1}^{5} E_i) &\leq (K+N-1)U2\,(U^{1/\alpha}\log U)^{-\alpha}/H \\ &\leq 2(K+N-1)(\log U)^{-\alpha}/H. \end{aligned}$$

The portion of

$$(b^+_{i,j,t}+1)^{\alpha+\frac{1}{2}} - {s^+_{i,j,t}}^{1-\frac{1}{4\alpha}} - ((b^+_{i,j,t_0}+1)^{\alpha+\frac{1}{2}} - {s^+_{i,j,t_0}}^{1-\frac{1}{4\alpha}})$$

that is caused by backoff counters before they exceed $R^{1/\alpha}$ is at most $KN(R^{1/\alpha}+1)^{\alpha+1/2}$. For the remaining portion, we use Lemma 3.1, to conclude that the the probability that the portion due to any one queue exceeds $c\,U^{1-\frac{1}{4(\alpha+1)}}$ is $O((\log U)^{-1})$. Thus,

$$\Pr(\overline{E_8} \mid \bigwedge_{i=1}^{5} E_i) \leq O((K+N-1)(\log U)^{-1}).$$

For the last calculation, note that the conditioning in the calculation of $E4$ only affects the arrival of the first $K+N-1$ messages. For the remaining messages, let M_t be the number of other messages received at control and slave queues during steps $t_0+5,\ldots,t$. The expected value of M_t is at most $\lambda t - t_0 - 4$. By a Chernoff bound

$$\Pr(M_t \geq \lambda(t-t_0+4)+U^{1/2}\log U \mid E_1 \wedge E_2 \wedge E_3) \leq 2\exp(-2(U^{1/2}\log U)^2/U) \leq 2\exp(-2\log^2 U).$$

Thus

$$\Pr(\overline{E_9} \mid \bigwedge_{i=1}^{5} E_i) \leq 2U\exp(-2\log^2 U).$$

Assuming that U and R are sufficiently large compared to N and K we have shown

$$\Pr(\bigwedge_{j=6}^{9} E_j \mid \bigwedge_{i=1}^{5} E_i) \geq \frac{1}{2}.$$

The claim follows. □

Claim 3.5 *There is a positive function f_2 such that* $\mathrm{Ex}[\delta \mid \overline{E}] \leq U' \cdot f_2(\alpha, K, N, V', R)$.

Proof: Let δ' denote the change in potential over all but the last step of a path in the tree of descendant states and let δ'' denote the change in potential during the last step of a path in the tree of descendant states. Clearly, $\delta = \delta' + \delta''$.

The proof of Claim 3.3 shows that $\mathrm{Ex}[\delta' \mid \overline{E}] \leq U' \cdot f_1(\alpha, K, N, V', R)$.

Suppose that p is a path of length t that doesn't satisfy E. We will calculate an upper bound on the amount that the potential could increase on step t_0+t-1. (Thus, we are upper bounding δ'' for this path.) The increase due to messages arriving at slave and control queues is at most $K+N-1$. The increase due to the backoff counter and step counter of queue $Q_{i,j}$ is at most

$$(b_{i,j,t_0+t-1}+2)^{\alpha+1/2} - (b_{i,j,t_0+t-1}+1)^{\alpha+1/2} + (b_{i,j,t_0+t-1}+1)^{\alpha-1/4}.$$

Using Fact 3.1, this is at most

$$\lceil \alpha+1/2 \rceil^{\lceil \alpha+3/2 \rceil} 2(b_{i,j,t_0+t-1}+1)^{\alpha-1/4}.$$

Since the parent of the last node in p is part of a good path, $b_{1,1,t_0+t-1} \leq (K+N-1)(R^{1/\alpha}+1)+1$ and for every slave queue $Q_{i,j}$, $b_{i,j,t_0+t-1} \leq U^{1/\alpha}\log(U)-1$. Thus, as long as U is big enough compared to K, N and R, the increase in potential due to the backoff counters and step counters of control and slave queues is at most $\lceil \alpha+1/2 \rceil^{\lceil \alpha+3/2 \rceil} 2KNU'$.

Finally, we use the fact (from Lemma 3.3) that the expected increase in potential of the free queues in any one step is at most $3KNf(\alpha)^{\alpha+1/2}$. □

Claim 3.6 $\mathrm{Ex}[\delta] \leq -1$.

Proof: $\mathrm{Ex}[\delta] \leq \mathrm{Ex}[\delta \mid E]\Pr[E] + \mathrm{Ex}[\delta \mid \overline{E}]$. The claim follows from Claims 3.3, 3.4, and 3.5 provided that U is sufficiently large compared to α, K, N, V', R, B, and $1/(1-\lambda)$. □

Claim 3.7 $\mathrm{Ex}[\delta^2] \leq \mathrm{POT}_{t_0}$.

Proof: Since each queue $Q_{i,j}$ can gain at most U messages and has $b_{i,j,t_0} \leq B$, $\delta \leq KN(U + (B+U+1)^{\alpha+1/2})$. Thus, as long as V is sufficiently large compared to α, K, N, B, and U, $\mathrm{Ex}[\delta^2] \leq V \leq \mathrm{POT}_{t_0}$. □

3.5.2 Case 2

Property 2 holds: When the Markov chain is started in state s right before step t_0 with $\mathrm{POT}(s) \geq V$, there is a backoff counter $b_{i,j,t_0} \geq Z$ such that with probability at least

$$(1-\lambda)^{K5}8^{-KN4}(b_{i,j,t_0}+4)^{-\alpha}2^{-KN},$$

queue $Q_{i,j}$ succeeds at least once during steps $t_0, \ldots, t_0+4$ and every other queue $Q_{i',j'}$ decides to send on step t (for $t \in \{t_0, \ldots, t_0+4\}$) only if $s_{i',j',t} \leq 8$.

Without loss of generality, let $Q_{1,1}$ be the queue $Q_{i,j}$ described in Property 2 and let E be the event that queue $Q_{1,1}$ succeeds at least once during steps $t_0, \ldots, t_0+4$ and every other queue $Q_{i,j}$ decides to send on step t (for $t \in \{t_0, \ldots, t_0+4\}$) only if $s_{i,j,t} \leq 8$. Recall that our goal is to show that there is a tree of depth at most $V-1$ rooted at s such that the expected decrease in the square of potential (over the tree) is at least $\mathrm{POT}(s)$. The tree that we will consider is the complete tree of depth 5. We consider steps t_0 through t_0+4 and analyze $\mathrm{POT}^2_{t_0+5} - \mathrm{POT}^2_{t_0}$. Clearly

$$\mathrm{Ex}[\mathrm{POT}^2_{t_0+5} - \mathrm{POT}^2_{t_0}] = \mathrm{Ex}[\mathrm{POT}^2_{t_0+5} - \mathrm{POT}^2_{t_0} | E]\Pr[E] + \mathrm{Ex}[\mathrm{POT}^2_{t_0+5} - \mathrm{POT}^2_{t_0} | \overline{E}]\Pr[\overline{E}]$$

We start by computing a lower bound for the decrease in potential in the event that E occurs. Let $g(\alpha)$ denote $(5\lceil \alpha + 1/2 \rceil^{\lceil \alpha+3/2 \rceil})^8$ First, we show that for every queue $Q_{i,j}$ except $Q_{1,1}$, when E occurs, $B^+_{i,j} \leq (g(\alpha)+6)^{\alpha+\frac{1}{2}}$. This is easy to see in the case that $b_{i,j,t_0} < g(\alpha)$. If $b_{i,j,t_0} \geq g(\alpha)$ then either $Q_{i,j}$ doesn't send (in which case $B^+_{i,j} = 0$) or $Q_{i,j}$ sends and succeeds (in which case $B^+_{i,j} \leq 5^{\alpha+1/2}$ or $Q_{i,j}$ decides to send and collides, in which case it never decides to send again and $B^+_{i,j}$ is at most

$$(b_{i,j,t_0}+2)^{\alpha+1/2} - (b_{i,j,t_0}+1)^{\alpha+1/2} - (\lfloor (b_{i,j,t_0}+2)^{\alpha} \rfloor - 4)^{1-1/(4\alpha)} + s^{1-1/(4\alpha)}_{i,j,t_0}$$

Using Fact 3.1, this is at most

$$\lceil \alpha + 1/2 \rceil^{\lceil \alpha+3/2 \rceil}(b_{i,j,t_0}+1)^{\alpha-1/2} - (\lfloor (b_{i,j,t_0}+2)^{\alpha} \rfloor - 4)^{1-1/(4\alpha)} + s^{1-1/(4\alpha)}_{i,j,t_0}$$

which is at most s_{i,j,t_0} since $b_{i,j,t_0} \geq g(\alpha)$.

For $Q_{1,1}$, when E occurs, $B^-_{1,1} \geq (b_{1,1,t_0}+1)^{\alpha+\frac{1}{2}} - (b_{1,1,t_0}+1)^{\alpha-\frac{1}{4}} - 5^{\alpha+1/2}$. $Q^+ \leq KN5$. Thus, when E occurs, since $b_{1,1,t_0} \geq Z$ and Z is sufficiently large compared to α, K and N, the potential decreases by at least $\frac{1}{2}(b_{1,1,t_0}+1)^{\alpha+\frac{1}{2}}$.

Thus, $\mathrm{POT}^2_{t_0+5} - \mathrm{POT}^2_{t_0}$ is at most

$$-(b_{1,1,t_0}+1)^{\alpha+\frac{1}{2}}\mathrm{POT}(s) + \frac{1}{4}(b_{1,1,t_0}+1)^{2\alpha+1} \leq -\frac{1}{2}(b_{1,1,t_0}+1)^{\alpha+\frac{1}{2}}\mathrm{POT}(s)$$

since $\mathrm{POT}(s) \geq (3/4)(b_{1,1,t_0}+1)^{\alpha+\frac{1}{2}}$. Using the lower bound on the probability of E from the statement of Property 2 and the fact that $b_{1,1,t_0} \geq Z$ and that Z is sufficiently large compared to α, K, N and $1/(1-\lambda)$, we find that

$$\mathrm{Ex}[\mathrm{POT}^2_{t_0+5} - \mathrm{POT}^2_{t_0} | E] \Pr[E] \leq -\frac{1}{2}(b_{1,1,t_0}+1)^{\frac{1}{4}}\mathrm{POT}(s).$$

Using the facts $Q^+ \leq 5KN$, $Q^- \geq 0$, and $B^- \geq 0$, we see that

$$\mathrm{Ex}[\mathrm{POT}^2_{t_0+5} - \mathrm{POT}^2_{t_0} \mid \overline{E}] \Pr[\overline{E}] \leq \mathrm{Ex}[(\mathrm{POT}_{t_0} + 5KN + B^+)^2 - \mathrm{POT}^2_{t_0} \mid \overline{E}] \Pr[\overline{E}].$$

Clearly, this is at most

$$[(5KN)^2 + 10KN \cdot \mathrm{POT}_{t_0} + 2(\mathrm{POT}_{t_0} + 5KN)\mathrm{Ex}[B^+ \mid \overline{E}] + \mathrm{Ex}[(B^+)^2 \mid \overline{E}]] \cdot \Pr[\overline{E}].$$

We can bound the last two expectations by noting that $\mathrm{Ex}[Y|\overline{E}]\Pr[\overline{E}] \leq \mathrm{Ex}[Y]$.

Recall that we defined B^+ to be $\sum_{i=1}^{N}\sum_{j=1}^{K} B^+_{i,j}$. Thus, $\mathrm{Ex}[B^+] = \sum_{i,j}\mathrm{Ex}[B^+_{i,j}]$. Similarly, $\mathrm{Ex}[(B^+)^2] = \sum_{i,j}\mathrm{Ex}[(B^+_{i,j})^2] + 2\sum_{\{i,j\}\neq\{i',j'\}}\mathrm{Ex}[B^+_{i,j}B^+_{i',j'}]$. We will now proceed to bound $\mathrm{Ex}[B^+_{i,j}]$, $\mathrm{Ex}[(B^+_{i,j})^2]$ and $\mathrm{Ex}[B^+_{i,j}B^+_{i',j'}]$ when b_{i,j,t_0} (or b_{i',j',t_0}) is large.

Claim 3.8 *Fix any sequence of values for the ρ and ρ^* variables. Then, for every queue $Q_{i,j}$ such that $b_{i,j,t_0} \geq 100$, when $\mathcal{M}$ and $\mathcal{M}_{i,j}$ are run with these ρ and ρ^* values, $B^+_{i,j} \leq B^{++}_{i,j}$.*

Proof: Until the first successful transmission by $Q_{i,j}$ in $\mathcal{M}$, $b_{i,j,t} = b^+_{i,j,t}$ and $s_{i,j,t} = s^+_{i,j,t}$. (Thus if $Q_{i,j}$ does not have any successful transmissions in $\mathcal{M}$, then the claim holds.) Assuming the first successful transmission in $\mathcal{M}$ is at step t', $b_{i,j,t'+1} = 0$ and $s_{i,j,t'+1} = 1$, but $b^+_{i,j,t'+1} \geq 100$ and $s^+_{i,j,t'+1} \geq 100^\alpha$. In the next $5-t'$ steps, $b_{i,j,t} < 5$ but $b^+_{i,j,t} \geq 100$. Then

$$(b^+_{i,j,t_0+5}+1)^{\alpha+\frac{1}{2}} - (s^+_{i,j,t_0+5})^{1-\frac{1}{4\alpha}} \geq 100^\alpha \geq 5^{\alpha+\frac{1}{2}} \geq (b_{i,j,t_0+5}+1)^{\alpha+\frac{1}{2}} - (s_{i,j,t_0+5})^{1-\frac{1}{4\alpha}}.$$

□

Claim 3.9 $\mathrm{Ex}[B^+_{i,j}] \leq Z^{\frac{1}{8}}/(KN)$.

Proof: If $b_{i,j,t_0} \leq (10\alpha)^{16\alpha}$, then $B^+_{i,j} \leq ((10\alpha)^{16\alpha}+6)^{\alpha+1/2} \leq (20\alpha)^{32\alpha^2} \leq Z^{\frac{1}{8}}/(KN)$. Otherwise, we use Claim 3.8 to show that $\mathrm{Ex}[B^+_{i,j}] \leq \mathrm{Ex}[B^{++}_{i,j}]$ and we bound $\mathrm{Ex}[B^{++}_{i,j}]$ as follows. If $Q_{i,j}$ doesn't send during the 5 steps then $B^{++}_{i,j} \leq 5$. Otherwise, we know by Lemma 3.2 that $B^{++}_{i,j} \leq s_{i,j,t_0}$. The probability that $Q_{i,j}$ sends during the 5 steps is $\min(1, 5/s_{i,j,t_0})$. Therefore, $\mathrm{Ex}[B^{++}_{i,j}] \leq 5 + (5/s_{i,j,t_0})s_{i,j,t_0} \leq 10 \leq Z^{1/8}/(KN)$. □

Claim 3.10 $\mathrm{Ex}[(B^+_{i,j})^2] \leq (\mathrm{POT}(s) + Z^{\frac{1}{4}})/(KN)$

Proof: If $b_{i,j,t_0} \leq (10\alpha)^{16\alpha}$, then we follow the proof of claim 3.9 to show that $(B^+_{i,j})^2 \leq Z^{\frac{1}{4}}/(KN)$. Otherwise, we use Claim 3.8 to show that $\mathrm{Ex}[(B^+_{i,j})^2] \leq \mathrm{Ex}[(B^{++}_{i,j})^2]$ and we bound $\mathrm{Ex}[(B^{++}_{i,j})^2]$ as in claim 3.9 to get $25 + 5s_{i,j,t_0}$. Since $s_{i,j,t_0} \leq (b_{i,j,t_0}+1)^\alpha$ and $10\alpha < b^{1/(16\alpha)}_{i,j,t_0}$, $\mathrm{Ex}[(B^{++}_{i,j})^2]$ is at most $(b_{i,j,t_0}+1)^{\alpha+1/16}$ which is at most $\mathrm{POT}(s)/KN$. □

Claim 3.11 $\mathrm{Ex}[B^+_{i,j}B^+_{i',j'}] \leq Z^{\frac{1}{4}}/(KN)^2$

Proof: If $b_{i,j,t_0} \leq (10\alpha)^{16\alpha}$ then we follow the proof of Claim 3.9 to show that $B^+_{i,j} \leq Z^{\frac{1}{8}}/(KN)$. We conclude that $\mathrm{Ex}[B^+_{i,j}B^+_{i',j'}] \leq [Z^{\frac{1}{8}}/(KN)]\mathrm{Ex}[B^+_{i',j'}]$ so the result follows from Claim 3.9. On the other hand, if $b_{i,j,t_0} > (10\alpha)^{16\alpha}$ and $b_{i',j',t_0} > (10\alpha)^{16\alpha}$, then we use Claim 3.8 to show that $B^+_{i,j} \leq B^{++}_{i,j}$ and $B^+_{i',j'} \leq B^{++}_{i',j'}$. Thus, $\mathrm{Ex}[B^+_{i,j}B^+_{i',j'}] \leq \mathrm{Ex}[B^{++}_{i,j}B^{++}_{i',j'}]$. But $B^{++}_{i,j}$ and $B^{++}_{i',j'}$ are independent, so the result follows from Claim 3.9 □

Recall that our goal was to bound $\mathrm{Ex}[\mathrm{POT}^2_{t_0+5} - \mathrm{POT}^2_{t_0}]$ and that we have shown that this is at most

$$-\frac{1}{2}(b_{1,1,t_0}+1)^{\frac{1}{4}}\mathrm{POT}(s) + (5KN)^2 + 10KN \cdot \mathrm{POT}(s) + 2(\mathrm{POT}(s)+5KN)\mathrm{Ex}[B^+] + \mathrm{Ex}[(B^+)^2].$$

Claim 3.9 shows that $\mathrm{Ex}[B^-] \leq Z^{\frac{1}{8}}$ and Claims 3.10 and 3.11 show that $\mathrm{Ex}[(B^+)^2] \leq \mathrm{POT}(s) + Z^{\frac{1}{4}} + 2Z^{\frac{1}{4}}$. Using the facts that $\mathrm{POT}(s) \geq Z^{3/8}$, that $b_{1,1,t_0} \geq Z$ and that Z is large compared to N and K we find that $\mathrm{Ex}[\mathrm{POT}^2_{t_0+5} - \mathrm{POT}^2_{t_0}]$ is at most $-\mathrm{POT}(s)$.

3.5.3 Case 3

Property 3 holds: When the Markov chain is started in state s right before step t_0 with $\mathrm{POT}(s) \geq V$, there is a backoff counter $b_{i,j,t_0} \geq B$ such that with probability at least

$$(1-\lambda)^{K(4+R)}8^{-KN4}(b_{i,j,t_0}+R+4)^{-\alpha}R^{-2\alpha KNR},$$

queue $Q_{i,j}$ succeeds at least once during steps $t_0, \ldots, t_0+R+3$ and every other queue $Q_{i',j'}$ decides to send on step t (for $t \in \{t_0, \ldots, t_0+R+3\}$) only if $s_{i',j',t} \leq R^{2\alpha}$.

Without loss of generality, let $Q_{1,1}$ be the queue $Q_{i,j}$ described in Property 3 and let E be the event that queue $Q_{1,1}$ succeeds at least once during steps $t_0, \ldots, t_0+R+3$ and every other queue $Q_{i,j}$ decides to send on step t (for $t \in \{t_0, \ldots, t_0+R+3\}$) only if $s_{i,j,t} \leq R^{2\alpha}$. Recall that our goal is to show that there is a tree of depth at most $V-1$ rooted at s such that the expected decrease in the square of potential (over the tree) is at least $\mathrm{POT}(s)$. The tree that we will consider is the complete tree of depth $R+4$. We consider steps t_0 through t_0+R-3 and analyze $\mathrm{POT}^2_{t_0+R+4} - \mathrm{POT}^2_{t_0}$. Clearly

$$\mathrm{Ex}[\mathrm{POT}^2_{t_0+R+4} - \mathrm{POT}^2_{t_0}] = \mathrm{Ex}[\mathrm{POT}^2_{t_0+R+4} - \mathrm{POT}^2_{t_0}|E]\Pr[E] + \mathrm{Ex}[\mathrm{POT}^2_{t_0+R+4} - \mathrm{POT}^2_{t_0}|\overline{E}]\Pr[\overline{E}]$$

We start by computing a lower bound for the decrease in potential in the event that E occurs. First, we show that for every queue $Q_{i,j}$ except $Q_{1,1}$, when E occurs, $B^+_{i,j} \leq (R^2+R+4)^{\alpha+\frac{1}{2}}$. This is easy to see in the case that $b_{i,j,t_0} < R^2$. If $b_{i,j,t_0} \geq R^2$ then either $Q_{i,j}$ doesn't send (in which case $B^+_{i,j}=0$) or $Q_{i,j}$ sends and succeeds (in which case $B^+_{i,j} \leq R^{\alpha+1/2}$ or $Q_{i,j}$ decides to send and collides, in which case it never decides to send again and $B^+_{i,j}$ is at most

$$(b_{i,j,t_0}+2)^{\alpha+1/2} - (b_{i,j,t_0}+1)^{\alpha+1/2} - (\lfloor (b_{i,j,t_0}+2)^{\alpha} \rfloor - (R+3))^{1-1/(4\alpha)} + s^{1-1/(4\alpha)}_{i,j,t_0}$$

Using Fact 3.1, this is at most

$$\lceil \alpha+1/2 \rceil^{\lceil \alpha+3/2 \rceil}(b_{i,j,t_0}+1)^{\alpha-1/2} - (\lfloor (b_{i,j,t_0}+2)^{\alpha} \rfloor - (R+3))^{1-1/(4\alpha)} + s^{1-1/(4\alpha)}_{i,j,t_0}$$

which is at most s_{i,j,t_0} since $b_{i,j,t_0} \geq R^2$ and R is sufficiently large.

For $Q_{1,1}$, when E occurs, $B^-_{1,1} \geq (b_{1,1,t_0}+1)^{\alpha+\frac{1}{2}} - (b_{1,1,t_0}+1)^{\alpha-\frac{1}{4}} - (R+4)^{\alpha+1/2}$. $Q^+ \leq KN(R+4)$. Thus, when E occurs, since $B \leq b_{1,1,t_0}$ and B is sufficiently large compared to R, K, and N, the potential decreases by at least $\frac{1}{2}(b_{1,1,t_0}+1)^{\alpha+\frac{1}{2}}$.

Thus, $\mathrm{POT}^2_{t_0+R+4} - \mathrm{POT}^2_{t_0}$ is at most

$$-(b_{1,1,t_0}+1)^{\alpha+\frac{1}{2}}\mathrm{POT}(s) + \frac{1}{4}(b_{1,1,t_0}+1)^{2\alpha+1} \leq -\frac{1}{2}(b_{1,1,t_0}+1)^{\alpha+\frac{1}{2}}\mathrm{POT}(s)$$

since $\mathrm{POT}(s) \geq (3/4)(b_{1,1,t_0}+1)^{\alpha+\frac{1}{2}}$. Using the lower bound on the probability of E from the statement of Property 2 and the fact that $B \leq b_{1,1,t_0}$ and B is sufficiently large compared to R, K, and N, we find that

$$\mathrm{Ex}[\mathrm{POT}^2_{t_0+R+4} - \mathrm{POT}^2_{t_0} | E]\Pr[E] \leq -\frac{1}{2}(b_{1,1,t_0}+1)^{\frac{1}{4}}\mathrm{POT}(s).$$

Using the facts $Q^+ \leq (R+4)KN$, $Q^- \geq 0$, and $B^- \geq 0$, we see that

$$\mathrm{Ex}[\mathrm{POT}^2_{t_0+R+4} - \mathrm{POT}^2_{t_0} \mid \overline{E}]\Pr[\overline{E}] \leq \mathrm{Ex}[(\mathrm{POT}_{t_0} + (R+4)KN + B^+)^2 - \mathrm{POT}^2_{t_0} \mid \overline{E}]\Pr[\overline{E}].$$

Clearly, this is at most

$$[((R+4)KN)^2 + 2(R+4)KN \cdot \mathrm{POT}_{t_0} + 2(\mathrm{POT}_{t_0} + (R+4)KN)\mathrm{Ex}[B^+ \mid \overline{E}] + \mathrm{Ex}[(B^+)^2 \mid \overline{E}]] \cdot \Pr[\overline{E}].$$

We can bound the last two expectations by noting that $\mathrm{Ex}[Y|\overline{E}]\Pr[\overline{E}] \leq \mathrm{Ex}[Y]$.

Recall that we defined B^+ to be $\sum_{i=1}^{N}\sum_{j=1}^{K} B^+_{i,j}$. Thus, $\mathrm{Ex}[B^+] = \sum_{i,j}\mathrm{Ex}[B^+_{i,j}]$. Similarly, $\mathrm{Ex}[(B^+)^2] = \sum_{i,j}\mathrm{Ex}[(B^+_{i,j})^2] + 2\sum_{\{i,j\}\neq\{i',j'\}}\mathrm{Ex}[B^+_{i,j}B^+_{i',j'}]$. We will now proceed to bound $\mathrm{Ex}[B^+_{i,j}]$, $\mathrm{Ex}[(B^+_{i,j})^2]$ and $\mathrm{Ex}[B^+_{i,j}B^+_{i',j'}]$ when b_{i,j,t_0} (or b_{i',j',t_0}) is large.

Claim 3.12 *Fix any sequence of values for the ρ and ρ^* variables. Then, for every queue $Q_{i,j}$ such that $b_{i,j,t_0} \geq (2R)^2$, when $\mathcal{M}$ and $\mathcal{M}_{i,j}$ are run with these ρ and ρ^* values, $B^+_{i,j} \leq B^{++}_{i,j}$.*

Proof: Until the first successful transmission by $Q_{i,j}$ in $\mathcal{M}$, $b_{i,j,t} = b^+_{i,j,t}$ and $s_{i,j,t} = s^+_{i,j,t}$. (Thus if $Q_{i,j}$ does not have any successful transmissions in $\mathcal{M}$, then the claim holds.) Assuming the first successful transmission in $\mathcal{M}$ is at step t', $b_{i,j,t'+1} = 0$ and $s_{i,j,t'+1} = 1$, but $b^+_{i,j,t'+1} \geq (2R)^2$ and $s^+_{i,j,t'+1} \geq (2R)^{2\alpha}$. In the next $R+4-t'$ steps, $b_{i,j,t} < R$ but $b^+_{i,j,t} \geq (2R)^2$. Then

$$(b^+_{i,j,t_0+R+4}+1)^{\alpha+\frac{1}{2}} - (s^+_{i,j,t_0+R+4})^{1-\frac{1}{4\alpha}} \geq (2R)^{2\alpha} \geq R^{\alpha+\frac{1}{2}} \geq (b_{i,j,t_0+R+4}+1)^{\alpha+\frac{1}{2}} - (s_{i,j,t_0+R+4})^{1-\frac{1}{4\alpha}}.$$

□

Claim 3.13 $\mathrm{Ex}[B^+_{i,j}] \leq B^{\frac{1}{8}}/(KN)$.

Proof: If $b_{i,j,t_0} \leq (2\alpha R)^{16\alpha}$, then $B^+_{i,j} \leq ((2\alpha R)^{16\alpha} + R + 1)^{\alpha+1/2} \leq (4\alpha R)^{32\alpha^2} \leq B^{\frac{1}{8}}/(KN)$. Otherwise, we use Claim 3.12 to show that $\mathrm{Ex}[B^+_{i,j}] \leq \mathrm{Ex}[B^{++}_{i,j}]$ and we bound $\mathrm{Ex}[B^{++}_{i,j}]$ as follows. If $Q_{i,j}$ doesn't send during the $R+4$ steps then $B^{++}_{i,j} \leq R+4$. Otherwise, we know by Lemma 3.2 that $B^{++}_{i,j} \leq s_{i,j,t_0}$. The probability that $Q_{i,j}$ sends during the $R+4$ steps is $\min(1,(R+4)/s_{i,j,t_0})$. Therefore, $\mathrm{Ex}[B^{++}_{i,j}] \leq R+4+((R+4)/s_{i,j,t_0})s_{i,j,t_0} \leq 2(R+4) \leq B^{1/8}/(KN)$. □

Claim 3.14 $\mathrm{Ex}[(B^+_{i,j})^2] \leq (\mathrm{POT}(s) + B^{\frac{1}{4}})/(KN)$

Proof: If $b_{i,j,t_0} \leq (2\alpha R)^{16\alpha}$, then we follow the proof of claim 3.13 to show that $(B^+_{i,j})^2 \leq B^{\frac{1}{4}}/(KN)$. Otherwise, we use Claim 3.12 to show that $\mathrm{Ex}[(B^+_{i,j})^2] \leq \mathrm{Ex}[(B^{++}_{i,j})^2]$ and we bound $\mathrm{Ex}[(B^{++}_{i,j})^2]$ as in claim 3.13 to get $(R+4)^2 + (R+4)s_{i,j,t_0}$. Since $s_{i,j,t_0} \leq (b_{i,j,t_0}+1)^{\alpha}$ and $2\alpha R < b^{1/(16\alpha)}_{i,j,t_0}$, $\mathrm{Ex}[(B^{++}_{i,j})^2]$ is at most $(b_{i,j,t_0}+1)^{\alpha+1/16}$ which is at most $\mathrm{POT}(s)/KN$. □

Claim 3.15 $\mathrm{Ex}[B^+_{i,j}B^+_{i',j'}] \leq B^{\frac{1}{4}}/(KN)^2$

Proof:

If $b_{i,j,t_0} \leq (2\alpha R)^{16\alpha}$ then we follow the proof of Claim 3.13 to show that $B^+_{i,j} \leq B^{\frac{1}{8}}/(KN)$. We conclude that $\mathrm{Ex}[B^+_{i,j}B^+_{i',j'}] \leq [B^{\frac{1}{8}}/(KN)]\mathrm{Ex}[B^+_{i',j'}]$ so the result follows from Claim 3.13. On the other hand, if $b_{i,j,t_0} > (2\alpha R)^{16\alpha}$ and $b_{i',j',t_0} > (2\alpha R)^{16\alpha}$, then we use Claim 3.12 to show that $B^+_{i,j} \leq B^{++}_{i,j}$ and $B^+_{i',j'} \leq B^{++}_{i',j'}$. Thus, $\mathrm{Ex}[B^+_{i,j}B^+_{i',j'}] \leq \mathrm{Ex}[B^{++}_{i,j}B^{++}_{i',j'}]$. But $B^{++}_{i,j}$ and $B^{++}_{i',j'}$ are independent, so the result follows from Claim 3.13 □

Recall that our goal was to bound $\mathrm{Ex}[\mathrm{POT}^2_{t_0+R+4} - \mathrm{POT}^2_{t_0}]$ and that we have shown that this is at most

$$-\frac{1}{2}(b_{1,1,t_0}+1)^{\frac{1}{4}}\mathrm{POT}(s)+((R+4)KN)^2+2(R+4)KN\cdot\mathrm{POT}(s)+2(\mathrm{POT}(s)+(R+4)KN)\mathrm{Ex}[B^+]+\mathrm{Ex}[(B^+)^2].$$

Claim 3.13 shows that $\mathrm{Ex}[B^+] \leq B^{\frac{1}{8}}$ and Claims 3.14 and 3.15 show that $\mathrm{Ex}[(B^+)^2] \leq \mathrm{POT}(s)+B^{\frac{1}{4}}+2B^{\frac{1}{4}}$. Using the facts that $\mathrm{POT}(s) \geq B^{3/8}$, that $b_{1,1,t_0} \geq B$ and that B is large compared to N, K and R we find that $\mathrm{Ex}[\mathrm{POT}^2_{t_0+(R+4)} - \mathrm{POT}^2_{t_0}]$ is at most $-\mathrm{POT}(s)$.

3.5.4 Case 4

None of Properties 1–3 hold. In order to define the terms that we need for this case, we consider a run of the chain for steps $t_0, \ldots, t_0+3$ in which no messages arrive and $Q_{i,j}$ decides to send on step t if $q_{i,j,t} > 0$ and $s_{i,j,t} \leq 8-t+t_0$. Note that if $q_{i,j,t_0+4} > 0$ and $s_{i,j,t_0+4} = 1$ then $Q_{i,j}$ succeeded in sending on step t_0+3 (so $l_{i,j,t_0+4} = 0$). If $q_{i,j,t_0+4} > 0$ and $s_{i,j,t_0+4} > 1$ then $s_{i,j,t_0+4} > 4$.

We use the following definitions. We say that queue $Q_{i,j}$ is *forced* on step t if $q_{i,j,t} > 0$ and $s_{i,j,t} = 1$. We say that it is *almost forced* if $q_{i,j,t} > 0$ and $s_{i,j,t} \leq 2$. We say that queue $Q_{i,j}$ is *short* if $q_{i,j,t_0+4} < R/2$. Otherwise, we say that it is *long*. If $j \neq j'$ we say that $Q_{i,j}$ *client-conflicts* with queue $Q_{i,j'}$. If $i \neq i'$ we say that $Q_{i,j}$ *server-conflicts* with queue $Q_{i',j}$ If $Q_{i,j}$ client-conflicts or server-conflicts with $Q_{i',j'}$ then we say that $Q_{i,j}$ *conflicts* with queue $Q_{i',j'}$. A queue $Q_{i,j}$ is a *potentially active queue* if $q_{i,j,t_0+4} = 0$ and $\lambda_{i,j} > 1/R^2$. A queue $Q_{i,j}$ is a *working queue* if $q_{i,j,t_0+4} > 0$ and $s_{i,j,t_0+4} < R^{2\alpha}$. A queue is called a *potentially working queue* if it is a potentially active queue or a working queue. A queue $Q_{i,j}$ is a *blocking queue* if it is potentially active or it has $q_{i,j,t_0+4} > 0$ and $b_{i,j,t_0+4} < R^{2/\alpha} - 2$.

In the appendix we will show that we can split the queues into categories so that the following conditions (which we call the *Case 4 conditions*) are satisfied.

1. There will be three categories of *control queues: solid control queues, delayed control queues,* and *temporary control queues.* No two control queues will conflict. Every queue that conflicts with a control queue is called a *slave* of that control queue. (A queue can be the slave of up to two control queues.) Every queue that is not a control queue or a slave is a *free queue.*
2. Slaves are not blocking queues. If a slave $Q_{i,j}$ has $b_{i,j,t_0+4} \geq B$ then $Q_{i,j}$ is a slave of a solid or delayed control queue.
3. Every solid control queue $Q_{i,j}$ is long and has $b_{i,j,t_0+4} = 0$ and $s_{i,j,t_0+4} = 1$.
4. Every delayed control queue $Q_{i,j}$ is long and has $b_{i,j,t_0+4} < Z$. If $Q_{i',j}$ is a working slave of $Q_{i,j}$, then either $q_{i',j,t_0+4} = 1$ and $\lambda_{i',j} \leq 1/R^2$, or there is a temporary or solid control queue $Q_{i',j'}$ with $q_{i',j',t_0+4} \geq \min(R/2, s_{i,j,t_0+4}-2, s_{i',j,t_0+4}-1)$. If $Q_{i,j'}$ is a working slave of $Q_{i,j}$, then either $q_{i,j',t_0+4} = 1$ and $\lambda_{i,j'} \leq 1/R^2$, or there is a temporary or solid control queue $Q_{i',j'}$ with $q_{i',j',t_0+4} \geq \min(R/2, s_{i,j,t_0+4}-2, s_{i,j',t_0+4}-1)$.
6. If $Q_{i,j}$ is a temporary control queue then $b_{i,j,t_0+4} = 0$, $s_{i,j,t_0+4} = 1$ and $q_{i,j,t_0+4} \geq 2$.
7. Every free queue $Q_{i,j}$ has $b_{i,j,t_0+4} < B$.

We now show that if the Case 4 conditions are satisfied, there is a tree of depth at most $V-1$ rooted at s such that the expected decrease in the square of the potential over the tree is at least $\mathrm{POT}(s)$. We will let W denote the depth of this tree.

In our proof, we use the following terminology. We refer to solid and delayed control queues as *permanent* control queues and we refer to slaves of these control queues as *permanent slaves.* All other slaves are called *temporary slaves.* We refer to temporary slaves and temporary control queue as *delayed free queues.* Without loss of generality, we assume that the permanent control queues are queues $Q_{1,1}$ through $Q_{r,r}$, and that the temporary control queues are queues $Q_{r+1,r+1}$ through $Q_{r',r'}$, ordered by q_{d,d,t_0+4} in decreasing order (i.e., $q_{r',r',t_0+4} \leq q_{r+1,r+1,t_0+4}$). If $Q_{i,j}$ is a slave queue and $m = \min\{i,j\}$ then we refer to $Q_{m,m}$ as the *primary* control queue of $Q_{i,j}$. We associate a *threshold value* $h_{i,j}$ with each queue $Q_{i,j}$ as follows. If $Q_{i,j}$ is a permanent control queue then $h_{i,j} = t_0 + W$. If it is a temporary control queue then $h_{i,j} = t_0 + 4 + \min(W^{1/2}, q_{i,j,t_0+4})$. If it is a free queue then $h_{i,j} = t_0 + 4$. The threshold value of each slave is equal to the threshold value of its primary control queue. If $h_{i,j} < t_0 + W$ then we will say that $Q_{i,j}$ is a free queue at the start of step $h_{i,j}$.

As in Case 1, the branching in our tree depends on the values of the ρ and ρ^* variables, so by fixing the values of the variables $\rho_{i,j,t}$ and $\rho^*_{i,j,t}$ for all i and j and all $t \leq t_0 + W - 1$, we fix a path p of length U. We make the following definitions for path p: For every slave queue $Q_{i,j}$, let $t_{i,j}$ denote the first step after $t_0 + 3$ on which $Q_{i,j}$ decides to send. If $t_{i,j} < h_{i,j}$ then let $\sigma'_{i,j}(p) = t_{i,j}$ and put $t_{i,j}$ in $\sigma'(p)$. Otherwise, let $\sigma'_{i,j}(p) = \infty$. Let $\sigma(p) = \sigma' \cup \{t+1 \leq t_0 + W - 1 \mid t \in \sigma'(p)\}$. Let $\sigma_k(p)$ denote the kth step in $\sigma(p)$. Let t_d denote the first step after step $t_0 + 3$ at which control queue $Q_{d,d}$ decides to send. If $t_d < t_0 + 4 + W^{1/2}$ then let $\tau_d(p) = t_d$. Otherwise, let $\tau_d(p) = \infty$. We say that path p is *good* if it satisfies the following properties.

1. On each step t, $(t_0 \leq t \leq t_0 + 3)$, no messages arrive and $Q_{i,j}$ decides to send iff $q_{i,j,t} > 0$ and $s_{i,j,t} \leq 8 - t + t_0$.
2. For each delayed control queue $Q_{d,d}$, if $s_{d,d,t_0+4} \leq (KN)^3$ then $\tau_d(p) = t_0 + 4$. Otherwise, $\tau_d(p) \leq t_0 + s_{d,d,t_0+4} + 2$.
3. For each slave queue $Q_{i,j}$, if $Q_{i,j}$ client-conflicts with a solid or temporary control queue and $s_{i,j,t_0+4} \leq (KN)^3$ then $\sigma'_{i,j}(p) = t_0 + 4$. If $s_{i,j,t_0+4} \leq (KN)^3$ but $Q_{i,j}$ doesn't client-conflicts with a solid or temporary control queue then $\sigma'_{i,j}(p) = t_0 + 5$. If $s_{i,j,t_0+4} > (KN)^3$ and $\sigma'_{i,j}(p) < h_{i,j}$ then $t_0 + 6 \leq \sigma'_{i,j}(p) \leq t_0 + s_{i,j,t_0+4} + 3$.
4. Consider two slave queues $Q_{i,j}$ and $Q_{i',j'}$ such that $q_{i,j,t_0+4} > 0$ and $q_{i',j',t_0+4} > 0$. If $\sigma'_{i,j}(p) = \sigma'_{i',j'}(p)$ then either $\sigma'_{i,j}(p) = t_0 + 4$, $\sigma'_{i,j}(p) = t_0 + 5$, or $\sigma'_{i,j}(p) = \infty$.
5. For each delayed control queue $Q_{d,d}$ and each slave $Q_{i,j}$, either $\tau_d(p) = t_0 + 4$ or $\tau_d(p) \neq \sigma'_{i,j}(p)$.
6. If for a control queue $Q_{d,d}$, $\tau_d(p) < \sigma_k(p) < h_{d,d}$, then $\rho^*_{d,d,\sigma_k(p)} \leq (k+1)^{-\alpha}$.
7. If $Q_{i,j}$ is a slave queue and $q_{i,j,t_0+4} > 0$ then for all t $(t_0 + 4 \leq t < h_{i,j})$, $\rho^*_{i,j,t} > 2(W \log W)^{-1}$.
8. If $Q_{i,j}$ is a slave queue and $q_{i,j,t_0+4} = 0$ then for all t $(t_0 + 4 \leq t < h_{i,j})$, $\rho_{i,j,t} > 2(W \log W)^{-1}$.
9. During the first t steps, the number of messages received by the permanent control and permanent slave queues is at most $r(\lambda t + W^{1/2} \log W)$.

The tree that we consider will be the tree consisting of every good path of length W plus every child of every internal node of such a path. We will show that for this tree, $\mathrm{Ex}[\Delta] \leq -\mathrm{POT}_{t_0}$. The key to showing this will be to prove that with sufficient probability a good path is taken when the chain is run.

First, we prove some claims about good paths.

Claim 3.16 *On any good path p, each delayed control queue $Q_{d,d}$ succeeds the first time that it decides to send after step $t_0 + 3$.*

Proof: This follows from Property 5 unless $\tau_d(p) = t_0 + 4$. By Property 3, the only slaves that send on step $t_0 + 4$ client-conflict with a solid or temporary control queue, so they cannot collide with $Q_{d,d}$. □

Claim 3.17 *On any good path p, every slave queue $Q_{i,j}$ decides to send at most once during steps $t_0 + 4, \ldots, h_{i,j} - 1$. Every control queue $Q_{d,d}$ decides to send on steps $\tau_d, \ldots, h_{d,d} - 1$.*

Proof: We start by observing that, if a slave $Q_{i,j}$ is not working, then, by Property 7 and Property 8, it will not decide to send at all during steps $t_0 + 4, \ldots, h_{i,j} - 1$. If a working slave $Q_{i,j}$ has $q_{i,j,t_0} = 1$ and $\lambda_{i,j} \leq (2W)^{-2}$, then, by Properties 7 and 8, after it decides to send once, it will not decide to send again. Every remaining slave conflicts with a temporary control queue or a solid control queue. If one of the remaining slaves decides to send and does not succeed, then by Property 7, it will not decide to send again.

We will prove by induction on t that if a remaining slave $Q_{i,j}$ first decides to send on step $\min\{t, h_{i,j} - 1\}$ it has a collision. Furthermore, every control queue $Q_{d,d}$ decides to send on steps $\tau_d, \ldots, \min\{t, h_{d,d} - 1\}$.

The base case is $t = t_0 + 4$, which holds by the definition of τ_d, the fact that each solid and temporary control queue $Q_{d,d}$ has $q_{d,d,t_0+4} > 0$ and $s_{d,d,t_0+4} = 1$, and Property 3.

For the inductive case, consider step $t + 1$. Suppose that for control queue $Q_{d,d}$, $t + 1 < h_{d,d}$. Then $q_{d,d,t+1} > 0$. If $t + 1 = \tau_d$ then $Q_{d,d}$ decides to on step $t + 1$ by definition. Suppose that $t + 1 > \tau_d$. By induction, $Q_{d,d}$ decides to send on step t. If $t \notin \sigma(p)$ or t is the last step in a consecutive block of steps in $\sigma(p)$, then $Q_{d,d}$ succeeded on step t so $s_{d,d,t+1} = 1$ and $Q_{d,d}$ decides to send on step $t+1$. Otherwise, we use Claim 3.16 to show that every delayed control queue succeeds the first time that it decides to send after step t_0+3. Therefore, for any control queue $Q_{d,d}$, $b_{d,d,t+1}$ is at most the number of collisions that it had during steps $t_0 + 4, \ldots, t$. Thus, by induction, $b_{d,d,t+1} \leq |\sigma'(p) \cap \{t_0 + 4, \ldots, t\}|$ and, therefore, $s_{d,d,t+1} \leq (|\sigma'(p) \cap \{t_0 + 4, \ldots, t\}| + 1)^{\alpha}$. By Property 6, $Q_{d,d}$ decides to send on step $t + 1$.

We now show that if $Q_{i,j}$ is a remaining slave and it first decides to send on step $t + 1 < h_{i,j}$, it has a collision.

The first case that we consider is the case $t + 1 = t_0 + 5$. In this case $Q_{i,j}$ collides with the solid or temporary control queue that it server-conflicts with. (If $Q_{i,j}$ client-conflicts with a solid or temporary or solid control queue it will instead send on step $t_0 + 4$. Note that the control queue decides to send on step $t_0 + 5$ since $h_{d,d} \geq t_0 + 6$. Furthermore, nothing that client-conflicts with it sends.)

The other case that we consider is the case in which $t + 1 > t_0 + 5$. By Property 4, $Q_{i,j}$ does not send at the same step as any other slave queue. If $Q_{i,j}$ conflicts with a solid control queue $Q_{d,d}$, then since no other slave queue sends at the same step as $Q_{i,j}$, it will be blocked by $Q_{d,d}$. If $Q_{i,j}$ conflicts with a temporary control queue $Q_{d,d}$ and sends before step $h_{d,d}$ then since no other slave queue sends at the same step as $Q_{i,j}$, it will be blocked by $Q_{d,d}$. The remaining case to consider is when the primary control queue of $Q_{i,j}$ is a delayed control queue $Q_{d,d}$, $Q_{i,j}$ also conflicts with temporary control queue $Q_{d',d'}$, but $Q_{i,j}$ sends after $h_{d',d'}$. Now, by the definition of delayed control queue, $q_{d',d',t_0} \geq s_{d,d,t_0+4} - 2$ so $h_{d',d',t_0} \geq t_0 + s_{d,d,t_0+4} + 2$. Thus, the step on which $Q_{i,j}$ sends is at least step $t_0 + s_{d,d,t_0+4} + 2$. By Property 2, $Q_{d,d}$ sends by this step, so $Q_{i,j}$ has a collision. □

Claim 3.18 *On any good path p, every control queue decides to send by step $t_0 + 3 + W^{1/2}$.*

Proof: Every solid or temporary control queue decides to send on step t_0+4. If $Q_{d,d}$ is a delayed control queue then $b_{d,d,t_0+4} < Z = W^{1/2\alpha} - 2$ so $s_{d,d,t_0+4} < W^{1/2}$. □

As in [HLR93], we will use the equality

$$\mathrm{Ex}[\Delta] = 2\mathrm{POT}(s) \cdot \mathrm{Ex}[\delta] + \mathrm{Ex}[\delta^2].$$

Thus it is sufficient to show that $\mathrm{Ex}[\delta] \leq -1$ and $\mathrm{Ex}[\delta^2] \leq \mathrm{POT}(s)$. Let E_1 be the event that a good path is taken when the chain is run. (That is, E_1 is the event that all conditions in the definition of "good" hold for W steps.) Let D_i be the event that condition i holds for U steps. For each queue $Q_{i,j}$, let $E_{2,i,j}$ be the event that $Q_{i,j}$ decides to send at step t ($t_0+4 \leq t < h_{i,j}$) with $b_{i,j,t} > (W \log W)^{1/\alpha} - 1$ and $s_{i,j,t} > (b_{i,j,t}+1)^{\alpha}/2$. Let $E_2 = \bigcup_{i,j} E_{2,i,j}$. Let E_3 be the event $\overline{E_1 \vee E_2}$. Let $W' = \max\{W^{1/2}\log W, W^{1/\alpha}\log^2 W, W^{1-1/(4\alpha(\alpha+1))}), (W\log W)^{1-1/(4\alpha)}\}$, and note that $W' = o(W)$.

Call two paths in the tree *equivalent* iff every queue $Q_{i,j}$ has the same ρ and ρ^* values from step t_0 through step $t_0 + h_{i,j} - 1$. This notion of equivalence is clearly an equivalence relation. Furthermore, if one path in the tree ends at step t (i.e., if $t < t_0 + W$, there is no good path continuing on from the node at level t, but there is a good path continuing on from the node at level $t-1$), then every equivalent path also ends at step t. (This is because the ρ and ρ^* values of a queue $Q_{i,j}$ on or after step $h_{i,j}$ are not considered in any of the properties.)

Let ν be the set of $h_{i,j}$ values for all queues $Q_{i,j}$. By the definition of the $h_{i,j}$ values, $|\nu| \leq K$. Assume ν is ordered, and let ν_k be the kth element of ν. During steps $\nu_k, \ldots, \nu_{k+1} - 1$, there will be a certain set of queues $Q_{i,j}$ ($d_k \leq i \leq N$, $d_k \leq j \leq K$) for some d_k which are the free queues. Let $\mathcal{M}'$ denote the Markov chain in which these free queues run the protocol and no other queues participate. By induction, there is a constant V' such that $\mathcal{M}'$ is V'-good. Now suppose that we fix the sequence of ρ and ρ^* values for the control and slave queues and we first run $\mathcal{M}$ for steps $t_0, \ldots, \nu_k - 1$ and we then run $\mathcal{M}$ for steps $\nu_k, \ldots, \nu_{k+1} - 1$. If we just look at the free queues during steps $\nu_k, \ldots, \nu_{k+1} - 1$, we can think of this as being a run of $\mathcal{M}'$, starting at step ν_k, in which $\mathcal{M}'$ is extended by the set of interrupt steps I which is determined by the sequence of ρ and ρ^* values for the control and slave queues. Lemma 3.3 shows that when $\mathcal{M}'$ is extended by I, the expected increase in potential in any one step is $O(KN)$. If the fixed sequence of ρ and ρ^* values is such that all of the properties continue to hold (except possibly after the last step) then the number of interrupt steps in I is at most $KN+1$. (To see this, note that each slave sends at most once in a good path.) Let $\mathcal{F}_t$ denote the set of free queues at the start of step t.

Claim 3.19 *Suppose that we fix a particular equivalence class of paths of length at least t, and we condition on the event that when $\mathcal{M}$ is run for t steps, starting with step t_0, one of the paths from this equivalence class is taken. Then the expected potential of the queues in $\mathcal{F}_t$ after the t steps is at most the original potential of the queues in $\mathcal{F}_t$ plus $O(KN) + O(K^2N)((2V')^2 2^{2V'} + O(KN)) + (O(KN)^{\alpha+3/2})W^{1/2}$.*

Proof: Note that each queue $Q_{i,j}$ in $\mathcal{F}_t$ satisfies all of the properties in the definition of good during steps $t_0, \ldots, h_{i,j} - 1$. (Otherwise, the paths would end before step $h_{i,j}$, so $Q_{i,j}$ would not become free.)

By Lemma 3.3, the expected increase in the potential of the queues in $\mathcal{F}_t$ during steps t_0 through t_0+3 is at most $O(KN)$.

Suppose that $t' > t_0 + 3$ and that $Q_{i,j}$ is a queue in $\mathcal{F}_t$ that is not free at the start of step t'. If $Q_{i,j}$ is a control queue then its potential goes up by at most $2 + (KN+1)^{\alpha+1/2}$ on step t'. (This follows from Claim 3.17, since each slave sends at most once prior to step t'.) If $Q_{i,j}$ is a slave

queue then $b_{i,j,t'} > R^{2/\alpha} - 2$. If it sends on step t', then since it does not violate property 7, $s_{i,j,t'} \leq (W \log W)/2$. By definition, the change in its potential is at most

$$1 + (b_{i,j,t'} + 2)^{\alpha+1/2} - \lfloor (b_{i,j,t'} + 2)^{\alpha} \rfloor^{1-1/(4\alpha)} - (b_{i,j,t'} + 1)^{\alpha+1/2} + s_{i,j,t'}^{1-1/(4\alpha)}.$$

By Fact 3.1, this is at most

$$1 + \lceil \alpha + 1/2 \rceil^{\lceil \alpha+3/2 \rceil} (b_{i,j,t'} + 1)^{\alpha-1/2} - \lfloor (b_{i,j,t'} + 2)^{\alpha} \rfloor^{1-1/(4\alpha)} + s_{i,j,t'}^{1-1/(4\alpha)}.$$

Since $b_{i,j,t'}$ is sufficiently large with respect to α, this is at most

$$1 - (1/2)(b_{i,j,t'} + 1)^{\alpha-1/4} - s_{i,j,t'}^{1-1/(4\alpha)}.$$

Clearly, this is negative, so the potential goes down.

There are at most KN queues in $\mathcal{F}_t$, and (by Claim 3.18) at most $W^{1/2}$ steps after step $t_0 + 3$ before a delayed free queue becomes a free queue.

To finish the proof of the claim, we will prove that during steps $\nu_k, \ldots, \nu_{k+1} - 1$, the potential of the current free queues (those queues that are free at the start of step ν_k) increases by $O(KN)((2V')^2 2^{2V'} + O(KN))$. Since $|\nu| \leq K$, this will prove the claim.

We view the current free queues as forming a Markov chain $\mathcal{M}'$ which is extended by the set of interrupts I' that is determined by the set of ρ and ρ^* values associated with the equivalence class. We now apply Lemma 3.6 using the facts that the expected increase in potential in any one step is $O(KN)$ and the number of interruptions is at most $KN + 1$. □

Claim 3.20 *There is a function f_1 such that*

$$\mathrm{Ex}[\delta | E_1] \leq -r[(1 - \lambda)W - W' \cdot f_1(N, K, V')].$$

Proof: Given E_1, we use Claims 3.18 and 3.17 to show that each permanent control queue successfully broadcasts for all but at most $KN + W^{1/2} + 4$ steps. Thus, we send at least $r(W - (KN + W^{1/2} + 4))$ messages. By Property 9, we receive at most $r(\lambda W + W^{1/2} \log W)$ messages in the permanent control and permanent slave queues. By Claim 3.17, the increase in potential due to the backoff and step counter of a permanent control queue is at most $(KN + 1)^{\alpha+1/2}$. We can follow the proof of Claim 3.19 to show that if a permanent slave decides to send, its potential goes down. Thus, the increase in potential due to the backoff and step counter of a permanent slave is at most $W^{1-1/(4\alpha)}$. Thus for each path the potential attributed to the permanent control and permanent slave queues decreases by at least

$$r((1 - \lambda)W - (W^{1/2} \log W + KN + W^{1/2} + 4) + O(NW^{1-1/(4\alpha)})).$$

Last, from Claim 3.19, for each possible equivalence class, the expected potential of the free and delayed free queues increases by at most $O(KN) + O(K^2N)((2V')^2 2^{2V'} + O(KN)) + (O(KN)^{\alpha+3/2})W^{1/2}$. □

Claim 3.21 *There is a positive function f_2 such that*

$$\Pr(E_1) \geq \frac{1}{f_2(N, K, \lambda)}.$$

Proof: We can divide the calculation as follows

$$\Pr(E_1) = \prod_{i=1}^{9} \Pr(D_i \mid \bigwedge_{j=1}^{i-1} D_j).$$

Now we analyze each probability in turn. We know $\Pr(D_1) \geq (1-\lambda)^{4K} 8^{-4KN}$,

Since for each delayed control queue $Q_{d,d}$, $4 \leq s_{d,d,t_0+4} \leq W^{1/2}$, $\Pr(D_2|D_1) \geq (KN)^{-3}$.

Note that the number of slaves is at most KN, and for each slave $Q_{i,j}$, $s_{i,j,t_0+4} \geq 4$. Then the probability that a slave $Q_{i,j}$ with $s_{i,j,t_0+4} \leq (KN)^3$ sends at the appropriate step (t_0+4 or t_0+5) is at least $\frac{1}{2}(KN)^{-3}$, and that a slave $Q_{i,j}$ with $s_{i,j,t_0+4} > (KN)^3$ sends by step $t_0 + s_{i,j,t_0+4} + 3$ is at least $\frac{1}{2}$. Thus $\Pr(D_3 \mid D_1 \wedge D_2) \geq (2KN)^{-3KN}$.

Given Property 3, for each slave $Q_{i,j}$ with $s_{i,j,t_0+4} > (KN)^3$, the probability of conflicting with any of the other slaves is at most $(KN)^{-2}$. Thus $\Pr(D_4 \mid D_1 \wedge D_2 \wedge D_3) \geq 1 - (KN)^{-1}$.

Given Properties 1 through 4, the probability that no delayed control queue sends at step t_0+5 is at least $1 - K(KN)^{-3}$. Then the probability that some slave queue first sends at a step in which one of the at most K delayed control queues first send (except for steps t_0+4 and t_0+5) is at most $(KN)(K/((KN)^3 - KN))$. Thus,

$$\Pr(D_5|D_1 \wedge D_2 \wedge D_3 \wedge D_4) \geq (1 - K(KN)^{-3})(1 - (KN)(K/((KN)^3 - KN)) \geq \frac{1}{2}.$$

Since $|\sigma(p)| \leq 2KN$, and the number of control queues is at most K, $\Pr(D_6| \bigwedge_{i=1}^{5} D_i) \geq ((2KN+1)!)^{-\alpha K}$.

It is easily seen that $\Pr(D_7| \bigwedge_{i=1}^{6} D_i) \geq 1 - 2KNW(W \log W)^{-1} \geq \frac{1}{2}$.

In the proofs that D_1 through D_7 hold with sufficiently high probability we forced some of the $\rho_{i,j,t}$ values to be large. The only times that we forced $\rho_{i,j,t}$ values to be small, we only forced them to be as small as $(KN)^{-3}$. Thus, the probability that a given queue fails to satisfy Property 8 on a given step is at most $2(KN)^3/(W \log W)$ and $\Pr(D_8| \bigwedge_{i=1}^{7} D_i) \geq 1 - KNW(2(KN)^3(W \log W)^{-1}) \geq \frac{1}{2}$.

For the last calculation, let M_t be the number of messages received by the permanent control and permanent slave queues by step t. The conditioning on D_7 only helps, so the expected value of M_T is at most $r\lambda t$. By a Chernoff bound

$$\Pr(M_t \geq r\lambda T + rW^{1/2} \log W | \bigwedge_{i=1}^{8} D_i) \leq 2\exp(-2(rW^{1/2} \log W)^2/(rW)) \leq 2\exp(-2r \log^2 W).$$

Thus

$$\Pr(D_9| \bigwedge_{i=1}^{8} D_i) \geq 1 - 2W \exp(-2r \log^2 W) \geq \frac{1}{2}.$$

The claim follows. □

Claim 3.22 *There is a positive function f_4 such that* $\mathrm{Ex}[\delta \mid E_3] \leq W' \cdot f_4(N, K, V')$.

Proof: Consider a group of equivalent paths G that satisfy E_3. Let δ' denote the change in potential over all but the last step, and let δ'' denote the change in potential of the last step. Clearly $\delta = \delta' + \delta''$. The proof of Claim 3.20 shows that $\mathrm{Ex}[\delta' \mid E_3] \leq W' \cdot f_1(N, K, V')$. To bound δ'', note that in the last step in the path, the expected increase in potential of the free queues is at most $O(KN)$. Also note that the increase in potential due to messages arriving is at most KN. Now we bound the increase in potential due to backoff counters and step counters of the non-free queues. Assuming that a queue does not fail in a send, the potential increase associated

with the backoff and step counters of that queue is bounded by 1 (i.e., the step counter decreases by 1). Since E_2 does not hold, a queue that sends and fails must have either $s_{i,j,t} \leq \frac{1}{2}(b_{i,j,t}+1)^\alpha$ or $b_{i,j,t} \leq (W \log W)^{1/\alpha} - 1$. If $b_{i,j,t} \leq (W \log W)^{1/\alpha} - 1$ then the potential increases by at most $(W \log W)^{1-1/(4\alpha)}$. Otherwise, since $s_{i,j,t} \leq \frac{1}{2}(b_{i,j,t}+1)^\alpha$ and $b_{i,j,t} > (W \log W)^{1/\alpha} - 1$, the potential actually decreases on a failed send. Thus the potential increase of the last step due to queues that send and fail is at most $O(KN(W \log W)^{1-1/(4\alpha)})$. □

Claim 3.23 *There is a positive function f_3 such that* $\mathrm{Ex}[\delta | E_2] \Pr[E_2] \leq W' \cdot f_3(K, N, V')$.

Proof: First, we observe that if $E_{2,i,j}$ is satisfied then $b_{i,j,t_0} > (W \log W)^{1/\alpha} - 2$. (To see this, suppose instead that $b_{i,j,t_0} \leq (W \log W)^{1/\alpha} - 2$. Then if $E_{2,i,j}$ holds, for some t we have $(W \log W)^{1/\alpha} - 2 \geq b_{i,j,t} \geq (W \log W)^{1/\alpha} - 3$ (either this is true for $t = t_0$ or there is a collision at step $t-1$). Then $s_{i,j,t} \geq \lfloor ((W \log W)^{1/\alpha} - 2)^\alpha \rfloor$. So if $Q_{i,j}$ sends after step t then Property 7 will be violated so the path will end.)

Let $\mathcal{B}$ be the set of queues $Q_{i,j}$ with $b_{i,j,t_0} > (W \log W)^{1/\alpha} - 2$.

Let δ' denote the change in potential over all but the last step and δ'' denote the change in potential of the last step. Clearly $\delta = \delta' + \delta''$. As in the proof of Claim 3.22, $\mathrm{Ex}[\delta' \mid E_3] \leq W' \cdot f_1(N, K, V')$, at most KN messages arrive on the last step, and the potential due to backoff counters and step counters of queues that are not in $\mathcal{B}$ go up by at most $O(KN(W \log W)^{1-1/(4\alpha)})$ on the last step. Let δ''' denote the increase on in potential on the last step due to the backoff counters and step counters of queues in $\mathcal{B}$.

We wish to bound $\mathrm{Ex}[\delta''' \mid E_2] \Pr[E_2]$. We do this as follows. For each queue $Q_{i,j}$ in $\mathcal{B}$, let

$$b^*(Q_{i,j}) = \begin{cases} b_{i,j,t_0}, & \text{if } b_{i,j,t_0} > (W \log W)^{1/\alpha} - 1 \text{ and } s_{i,j,t_0} > (b_{i,j,t_0}+1)^\alpha/2; \\ b_{i,j,t_0}+1, & \text{otherwise.} \end{cases}$$

(Note that $E_{2,i,j}$ will occur if $Q_{i,j}$ sends with backoff counter at least $b^*(Q_{i,j})$ but that it will not occur because of $Q_{i,j}$ sending with a smaller backoff counter. Also note that $Q_{i,j}$ will never sound with backoff counter bigger than $b^*(Q_{i,j})$ because it will violate Property 7 when it sends with backoff counter $b^*(Q_{i,j})$, so the path will end.) Let the queues in $\mathcal{B}$ be $Q_1, \ldots, Q_m$, ordered such that $b^*(Q_1) \geq \cdots \geq b^*(Q_m)$. Let S_i be the event that Q_i attempts to send once it has attained a backoff counter of $b^*(Q_i)$. Then

$$\begin{aligned} \mathrm{Ex}[\delta''' | E_2] \Pr[E_2] &\leq \sum_{i=1}^{m} \mathrm{Ex}[\delta''' | S_i \wedge \bigwedge_{j=1}^{i-1} \overline{S_j}] \Pr[S_i \wedge \bigwedge_{j=1}^{i-1} \overline{S_j}] \\ &\leq \sum_{i=1}^{m} \mathrm{Ex}[\delta''' | S_i \wedge \bigwedge_{j=1}^{i-1} \overline{S_j}] \Pr[S_i] \end{aligned}$$

If Q_i attempts to send once it has attained backoff counter $b^*(Q_i)$ then its potential increases by at most

$$(b^*(Q_i)+2)^{\alpha+1/2} - (b^*(Q_i)+1)^{\alpha+1/2} + (b^*(Q_i)+1)^{\alpha-1/4}.$$

Using Fact 3.1, we find that if $S_i \wedge \bigwedge_{j=1}^{i-1} \overline{S_j}$ then $\delta''' = \mathrm{O}(KN(b^*(Q_i)+1)^{\alpha-1/4})$. Once Q_i has reached backoff counter $b^*(Q_i)$, its step counter will be at least $\frac{1}{4}(b^*(Q_i)+1)^\alpha$ for the next W steps, and thus $\Pr[S_i] \leq 4W(b^*(Q_i)+1)^{-\alpha}$.

Plugging this into the equations above, we obtain

$$\begin{aligned}
\mathrm{Ex}[\delta'''|E_2] &\leq \sum_{i=1}^{m}(O(KN(b^*(Q_i)+1)^{\alpha-\frac{1}{4}}))W(b^*(Q_i)+1)^{-\alpha} \\
&\leq O((KN)^2(W(b^*(Q_i)+1)^{-\frac{1}{4}})) \\
&\leq O((KN)^2(W((W\log W)^{1/\alpha}-1)^{-\frac{1}{4}})) \\
&\leq O(W'(KN)^2).
\end{aligned}$$

□

Claim 3.24 $\mathrm{Ex}[\delta] \leq -1$.

Proof: Using the previous claims we have

$$\begin{aligned}
\mathrm{Ex}[\delta] &= \mathrm{Ex}[\delta|E_1]\Pr[E_1] + \mathrm{Ex}[\delta|E_2]\Pr[E_2] + \mathrm{Ex}[\delta|E_3]\Pr[E_3] \\
&\leq [-r(1-\lambda)W + rW'\cdot f_1]\frac{1}{f_2} + W'(f_3+f_4) \\
&\leq -(1-\lambda)W\frac{1}{f_2} + W'(f_1+f_3+f_4) \\
&\leq -1,
\end{aligned}$$

assuming W is large enough. □

Claim 3.25 $\mathrm{Ex}[(\delta)^2] \leq \mathrm{POT}$.

Proof: If $Q_{i,j}$ is a free queue or a delayed free queue then $b_{i,j,t_0} < B$. Therefore, the potential due to $Q_{i,j}$ increases by at most $O((B+W)^{\alpha+1/2})$.

Suppose that $Q_{i,j}$ is a permanent control queue or a permanent slave, but that $E_{2,i,j}$ does not hold. Using the proofs of Claims 3.20 and 3.22, we see that the potential due to queue $Q_{i,j}$ increases by at most $O(W'\cdot f_1(N,K,V'))$.

Thus, as long as V' is sufficiently large compared to N, K, V', B and W, $\mathrm{Ex}[\delta^2 \mid E_1 \vee E_3] \leq V$.

To bound $\mathrm{Ex}[(\delta)^2 \mid E_2]\Pr(E_2)$ we follow the proof of Claim 3.23.

$$\begin{aligned}
\mathrm{Ex}[(\delta)^2|E_2]\Pr(E_2) &\leq \sum_{i=1}^{m}\mathrm{Ex}[(\delta)^2|S_i \wedge \bigwedge_{j=1}^{i-1}\overline{S_j}]\Pr[S_i \wedge \bigwedge_{j=1}^{i-1}\overline{S_j}] \\
&\leq \sum_{i=1}^{m}\mathrm{Ex}[(\delta)^2|S_i \wedge \bigwedge_{j=1}^{i-1}\overline{S_j}]\Pr[S_i]
\end{aligned}$$

As before, $\Pr[S_i] \leq 4W'(b^*(Q_i)+1)^{-\alpha}$. Given that $S_i \wedge \bigwedge_{j=1}^{i-1}\overline{S_j}$,

$$\delta \leq O((B+W)^{\alpha+1/2}) + O(W'\cdot f_1(N,K,V')) + O(KN(b^*(Q_i)+1)^{\alpha-1/4}).$$

Thus,

$$\begin{aligned}
\mathrm{Ex}[(\delta)^2|E_2]\Pr(E_2) &\leq \sum_{i=1}^{m}[O((B+W)^{\alpha+1/2} + W'\cdot f_1(N,K,V') + (b^*(Q_i)+1)^{\alpha-1/4})]^2 W(b^*(Q_i)+1)^{-\alpha} \\
&\leq f(B,W,N,K)(b^*(Q_1)+1)^{\alpha}.
\end{aligned}$$

The claim follows since $\mathrm{POT}_{t_0} = \Omega((b^*(Q_1))^{\alpha+1/2})$ and V is sufficiently large with respect to B, W, N, and K. □

This Concludes the proof of Theorem 3.1. □

Acknowledgments: We thank Tom Leighton for proposing the problem of dynamic routing in optical networks and John DeLaurentis for useful discussions related to this paper.

References

[AM88] R. J. Anderson and G. L. Miller. Optical communication for pointer based algorithms. Technical Report CRI 88-14, University of Southern California, 1988.

[GGMM88] Jonathan Goodman, Albert G. Greenberg, Neal Madras, and Peter March. Stability of binary exponential backoff. *J. Assoc. Comput. Mach.*, 35(3):579–602, 1988. A preliminary version appeared in STOC 85.

[HLR93] Johan Håstad, Tom Leighton, and Brian Rogoff. Analysis of backoff protocols for multiple access channels. Pre-print - a preliminary version appeared in STOC 87, 1993.

[MB76] R. Metcalfe and D. Boggs. Distributed packet switching for local computer networks. *Comm. ACM*, 19:395–404, 1976.

[RT94] Satish Rao and Thanasis Tsantilas. Optical interprocessor communication protocols. In *Proc. 1st Symp. on Massively Para. Proc. Using Optical Interconnects*, 1994.

[RU95] Prabhakar Raghavan and Eli Upfal. Stochastic contention resolution with short delays. In *Proc. STOC 95*, 1995.

[Sti74] Shaler Stidham. The last word on $L = \lambda W$. *Operations Research*, 22:417–421, 1974.

A Establishing the Case 4 Conditions

In the following case analysis, we show that if we are in Case 4 then we can split the queues into categories so that the Case 4 conditions are satisfied.

First, we note that since we are in Case 4, none of properties 1–3 hold. That is, when the Markov chain is started in state s right before step t_0 with $\mathrm{POT}(s) \geq V$, there is not a backoff counter $b_{i,j,t_0} \geq Z$ such that with probability at least

$$(1-\lambda)^{K5}8^{-KN4}(b_{i,j,t_0}+5)^{-\alpha}2^{-KN},$$

queue $Q_{i,j}$ succeeds at least once during steps $t_0, \ldots, t_0+4$ and every other queue $Q_{i',j'}$ decides to send on step t (for $t \in \{t_0, \ldots, t_0+4\}$) only if $s_{i',j',t} \leq 8$. There is a backoff counter $b_{i,j,t_0} \geq B$, but for every such backoff counter, it is not the case that with probability at least

$$(1-\lambda)^{K(4+R)}8^{-KN4}(b_{i,j,t_0}+R+4)^{-\alpha}R^{-2\alpha KNR},$$

queue $Q_{i,j}$ succeeds at least once during steps $t_0, \ldots, t_0+R+3$ and every other queue $Q_{i',j'}$ decides to send on step t (for $t \in \{t_0, \ldots, t_0+R+3\}$) only if $s_{i',j',t} \leq R^{2\alpha}$.

Suppose that $b_{i,j,t_0} \geq B$. We will show that unless we are in Case 2 or Case 3, we can identify a solid or delayed control queue that conflicts with $Q_{i,j}$. In order to do so, we need some definitions. We will say that a queue $Q_{i',j'}$ which conflicts with $Q_{i,j}$ is a *solid candidate* if $q_{i',j',t_0+4} > 0$, $s_{i',j',t_0+4} = 1$, and (therefore) $b_{i',j',t_0+4} = 0$. We will say that $Q_{i',j'}$ is a *delayed* candidate if it is long and has no conflicting blocking queues and has $b_{i',j',t_0+4} < Z$, and satisfies the following conditions.

1. If $Q_{i'',j'}$ is a working queue then either $q_{i'',j',t_0+4} = 1$ and $\lambda_{i'',j'} \leq 1/R^2$, or there is a queue $Q_{i'',j''}$ which does not conflict with a blocking queue and has $s_{i'',j'',t_0+4} = 1$ and

$$q_{i'',j'',t_0+4} \geq \min(R/2, s_{i',j',t_0+4} - 2, s_{i'',j',t_0+4} - 1).$$

2. If $Q_{i',j''}$ is a working queue then either $q_{i',j'',t_0+4} = 1$ and $\lambda_{i',j''} \leq 1/R^2$, or there is a queue $Q_{i'',j''}$ which does not conflict with a blocking queue and has $s_{i'',j'',t_0+4} = 1$ and

$$q_{i'',j'',t_0+4} \geq \min(R/2, s_{i',j',t_0+4} - 2, s_{i',j'',t_0+4} - 1).$$

We say that a solid candidate is *clear* if it has no conflicting blocking queues and we say that it is *unclear* otherwise. Note that each client and each server has at most one candidate, so if candidates are made into control queues then these control queues will not conflict. (To see this, note that each solid candidate succeeded on step t_0+3, so solid candidates cannot conflict with each other. Solid candidates and delayed candidates are blocking, so they cannot conflict with delayed candidates.) Note that clear solid candidates and delayed candidates do not conflict with blocking queues.

We now consider the possible cases (split by the number and type of solid candidates that exist):

1. If there is no solid candidate then we are in Case 3. Consider the run of the chain for steps $t_0, \ldots, t_0+3$ that we described earlier. Suppose that on step t_0+4, no message arrives, $Q_{i,j}$ decides to send if $q_{i,j,t_0+4} > 0$, and every other queue decides to send only if it is forced. The probability of this event is at least $(1-\lambda)^{K(5)}(8)^{-KN4}(b_{i,j,t_0}+5)^{-\alpha}2^{-KN}$. Since there are no solid candidates, $Q_{i,j}$ succeeds if it decides to send on step t_0+4. Furthermore, every queue $Q_{i',j'}$ other than $Q_{i,j}$ only decides to send on step t with $s_{i',j',t} \leq R^{2\alpha}$.
2. If $Q_{i',j'}$ is a long clear solid candidate then we can make $Q_{i',j'}$ a solid control queue.

3. If $Q_{i',j'}$ is an unclear solid candidate and there is no other solid candidate then we are in Case 3. Consider the run of the chain for step $t_0, \ldots, t_0 + 3$ that we described earlier. Suppose that no messages arrive on steps $t_0 + 4, \ldots, t_0 + 6$. If $q_{i',j',t_0+4} = 1$, then no other queues that conflict with either $Q_{i',j'}$ or $Q_{i,j}$ decide to send on steps $t_0 + 4$ through $t_0 + 6$ and after one step $q_{i',j',t_0+5} = 0$. Therefore, $Q_{i,j}$ can decide to send on step $t_0 + 6$, and it will succeed. Therefore, suppose that $q_{i',j',t_0+4} > 1$. If there is a blocking queue $Q_{i',j''}$ then $Q_{i',j'}$ and $Q_{i',j''}$ decide to send on steps $t_0 + 4$ and $t_0 + 5$. Otherwise, there is a blocking queue $Q_{i'',j'}$. If there is a blocking queue that client-conflicts with $Q_{i'',j'}$ then on step $t_0 + 4$ $Q_{i'',j'}$ decides to send and every blocking queue that client-conflicts with $Q_{i'',j'}$ decides to send. On step $t_0 + 5$ $Q_{i',j'}$ and $Q_{i'',j'}$ decide to send. Otherwise, $Q_{i',j'}$ and $Q_{i'',j'}$ decide to send on steps $t_0 + 4$ and $t_0 + 5$. On step $t_0 + 6$, $Q_{i,j}$ decides to send if $q_{i,j,t_0+6} > 0$. On each of the steps, every other queue decides to send only if it is forced. The probability of this event is at least $(1 - \lambda)^{K(7)}(8)^{-KN4}(b_{i,j,t_0} + 7)^{-\alpha} R^{-2\alpha KN(3)}$. Note that $Q_{i,j}$ succeeds if it decides to send on step $t_0 + 6$. Furthermore, every queue $Q_{i'',j''}$ other than $Q_{i,j}$ only decides to send on step t with $s_{i'',j'',t} \leq R^{2\alpha}$.

4. If there are two unclear solid candidates, $Q_{i,j'}$ and $Q_{i',j}$, then we are in Case 3. Consider the run of the chain for step $t_0, \ldots, t_0 + 3$ that we described earlier. Suppose that no messages arrive on steps $t_0 + 4, \ldots, t_0 + 6$. If $q_{i,j',t_0+4} = 1$ or $q_{i',j,t_0+4} = 1$ then we can treat $Q_{i,j'}$ and $Q_{i',j}$ separately, using the analysis of the previous case. Also, if $Q_{i,j'}$ conflicts with blocking queue $Q_{i'',j''}$ and $Q_{i',j}$ conflicts with blocking queue $Q_{i''',j'''}$, with $i''' \neq i$, $i''' \neq i''$ and $i'' \neq i'$, then again we can treat $Q_{i,j'}$ and $Q_{i',j}$ separately, using the analysis of the previous case.

   Otherwise, note that $i''' \neq i$, since $Q_{i,j}$ cannot be a blocking queue (because $b_{i,j,t_0+4} \geq B$). Thus, for every blocking queue $Q_{i''',j'''}$ that conflicts with $Q_{i',j}$ and every blocking queue $Q_{i'',j''}$ that conflicts with $Q_{i,j'}$, either $i''' = i''$ or $i'' = i'$. Note that no blocking queue client-conflicts with $Q_{i,j'}$ in this case.

   If there is a blocking queue $Q_{i',j'}$, then suppose no messages arrive on steps $t_0 + 4, \ldots, t_0 + 9$. On step $t_0 + 4$, $Q_{i,j'}$, $Q_{i',j}$, $Q_{i',j'}$ decide to send, along with any working queues that client-conflict with $Q_{i,j'}$ or $Q_{i',j}$. On step $t_0 + 5$, $Q_{i',j}$ decides to send, along with any working queues that server-conflict with $Q_{i',j}$. (Note that after step $t_0 + 5$, any working queue $Q_{i^*,j}$ that server-conflicts with $Q_{i',j}$ has $s_{i^*,j,t_0+6} \geq 5$.) On step $t_0 + 6$ and $t_0 + 7$, $Q_{i',j}$ and $Q_{i',j'}$ decide to send, along with any blocking queues that client-conflict with $Q_{i',j}$. On step $t_0 + 8$, $Q_{i,j'}$ and $Q_{i',j'}$ decide to send, along with any forced queues, unless $q_{i,j',t_0+8} = 0$, in which case just $Q_{i',j'}$ decides to send. On step $t_0 + 9$, $Q_{i,j}$ decides to send if $q_{i,j,t_0+9} > 0$. (Note that no queue conflicting with $Q_{i,j}$ is forced at step $t_0 + 9$. On each of the steps, every other queue decides to send only if it is forced.

   The remaining possibility is that $Q_{i,j'}$ conflicts with blocking queue $Q_{i'',j'}$ and $Q_{i',j}$ conflicts with blocking queue $Q_{i'',j}$ such that $i'' \neq i$ and $i'' \neq i'$. (Note that there are no other blocking queues that conflict with $Q_{i,j'}$ or $Q_{i',j}$, or the situation could have been handled previously.) Suppose that no messages arrive on steps $t_0 + 4, \ldots, t_0 + 10$. For t in the range $t_0 + 4 \leq t \leq t_0 + 6$, $Q_{i'',j}$ and $Q_{i'',j'}$ decide to send on step t. $Q_{i,j'}$ decides to send on step t if $q_{i,j',t} > 0$ and $Q_{i',j}$ decides to send on step t if $q_{i',j,t} > 0$. On step $t_0 + 4$ any working queue that client conflicts with $Q_{i,j'}$, $Q_{i',j}$ or $Q_{i'',j}$ decides to send. On step $t_0 + 5$ any working queues that server-conflict with $Q_{i,j'}$ or $Q_{i',j}$ decide to send. On steps $t_0 + 5$ and $t_0 + 6$, any blocking queues that client-conflict with $Q_{i'',j}$ decide to send. If $q_{i',j,t_0+7} = 0$ then we proceed as follows. if $q_{i',j,t_0+7} > 0$ then $Q_{i',j}$ and $Q_{i'',j'}$ decide to send on step $t_0 + 7$. On step $t_0 + 8$, $Q_{i,j}$ decides to send if $q_{i,j,t_0+8} > 0$. On each of the steps, every other queue decides to send only if it is forced. (Note that no queue that conflicts with $Q_{i,j}$ is forced at step $t_0 + 8$.) If

$q_{i',j,t_0+7} > 0$ then $Q_{i',j}$ and $Q_{i'',j}$ decide to send on steps t_0+7 and t_0+8. If $q_{i',j,t_0+9} > 0$ then $Q_{i',j}$ and $Q_{i'',j'}$ decide to send on step t_0+9. On step t_0+10, $Q_{i,j}$ decides to send if $q_{i,j,t_0+10} > 0$. On each of the steps, every other queue decides to send only if it is forced. (Note that no queue that conflicts with $Q_{i,j}$ is forced at step t_0+10.)

The probability of this event is at least $(1-\lambda)^{K(11)}(8)^{-KN4}(b_{i,j,t_0}+11)^{-\alpha}R^{-2\alpha KN(7)}$. Note that every queue $Q_{i'',j''}$ other than $Q_{i,j}$ only decides to send on step t with $s_{i'',j'',t} \leq R^{2\alpha}$.

5. If $Q_{i',j'}$ is an unclear solid candidate and $Q_{i'',j''}$ is a short clear solid candidate then we are in Case 3. Consider the run of the chain for steps $t_0,\ldots,t_0+3$ that we described earlier. Suppose that on steps $t_0+4,\ldots,t_0+\lfloor(R-1)/2\rfloor+8$ no messages arrive. Suppose that on step t_0+4, every working queue that client-conflicts with $Q_{i'',j''}$ decides to send. On step t_0+5, $Q_{i'',j''}$ decides to send and every working queue that server-conflicts with $Q_{i'',j''}$ decides to send. For t in the range $\{t_0+6,\ldots,t_0+\lfloor(R-1)/2\rfloor+6\}$, $Q_{i'',j''}$ decides to send on step t if $q_{i'',j'',t} > 0$. (Thus $Q_{i'',j''}$ will empty its queue by step $t_0+\lfloor(R-1)/2\rfloor+6$.)

   If $Q_{i',j'}$ client-conflicts with $Q_{i,j}$ and with another blocking queue $Q_{i,j''}$ then $Q_{i',j'}$ and $Q_{i,j''}$ decide to send on steps $t_0+4,\ldots,t_0+\lfloor(R-1)/2\rfloor+6$ and so do any queues that client-conflict with them and are almost forced. $Q_{i,j}$ decides to send on step $t_0+\lfloor(R-1)/2\rfloor+7$ if $q_{i,j,t_0+\lfloor(R-1)/2\rfloor+7} > 0$. If $Q_{i',j'}$ server-conflicts with $Q_{i,j}$ and client-conflicts with a blocking queue $Q_{i',j''}$ then on steps $t_0+4,\ldots,t_0+\lfloor(R-1)/2\rfloor+5$, $Q_{i',j'}$ and $Q_{i',j''}$ decide to send and so does any other queue that client-conflicts with them and is almost forced. On step $t_0+\lfloor(R-1)/2\rfloor+6$, $Q_{i',j'}$ decides to send and so does any queue that server-conflicts with $Q_{i,j}$ and is almost forced. On step $t_0+\lfloor(R-1)/2\rfloor+7$ $Q_{i',j''}$ decides to send and so does $Q_{i,j}$ if $q_{i,j,t_0+\lfloor(R-1)/2\rfloor+7} > 0$.

   If $Q_{i',j'}$ does not client-conflict with a blocking queue then it server-conflicts with a blocking queue $Q_{i'',j'}$. If $Q_{i'',j'}$ does not client-conflict with a blocking queue then on step t_0+4 $Q_{i',j'}$ and $Q_{i'',j'}$ decide to send and nothing that client-conflicts with either of them decides to send. If there is a working queue that client-conflicts with $Q_{i',j'}$ then $Q_{i',j'}$ decides to send on step t_0+5 and so does any working queue that client-conflicts with it. Otherwise, $Q_{i',j'}$ does not decide to send on step t_0+5. Similarly, if there is a working queue that client-conflicts with $Q_{i'',j'}$ then $Q_{i'',j'}$ decides to send on step t_0+5 and so does any working queue that client-conflicts with it. Otherwise, $Q_{i'',j'}$ does not decide to send on step t_0+5. On steps $t_0+6,\ldots,t_0+\lfloor(R-1)/2\rfloor+6$ $Q_{i',j'}$ and $Q_{i'',j'}$ decide to send and so does any queue that server-conflicts with them and is almost forced. On step $t_0+\lfloor(R-1)/2\rfloor+7$ $Q_{i,j}$ decides to send if $q_{t_0+\lfloor(R-1)/2\rfloor+7} > 0$.

   If there is a blocking queue $Q_{i'',j''}$ that client-conflicts with $Q_{i'',j'}$ then for even ℓ in the range $0 \leq \ell \leq \lfloor(R-1)/2\rfloor+4$, on step $t_0+4+\ell$, $Q_{i'',j'}$ and $Q_{i'',j''}$ decide to send and any queue that client-conflicts with them and is almost forced decides to send. On step t_0+4 any working queue that conflicts with $Q_{i',j'}$ decides to send. If $q_{i',j',t_0+5} > 0$ then for any odd ℓ in the range $0 \leq \ell \leq \lfloor(R-1)/2\rfloor+4$, $Q_{i',j'}$ and $Q_{i'',j'}$ decide to send. On step $t_0+\lfloor(R-1)/2\rfloor+7$ or $t_0+\lfloor(R-1)/2\rfloor+8$ (whichever is of the same parity as t_0+4), $Q_{i,j}$ decides to send if its queue is non-empty. Every other queue only sends if it is forced. The probability of this event is at least $(1-\lambda)^{K(\lfloor(R-1)/2\rfloor+8)}(8)^{-KN4}(b_{i,j,t_0}+\lfloor(R-1)/2\rfloor+8)^{-\alpha}R^{-2\alpha KN(\lfloor(R-1)/2\rfloor+4)}$. On steps $t_0+6,\ldots,t_0+\lfloor(R-1)/2\rfloor+7$ nothing that conflicts with $Q_{i'',j''}$ decides to send, so it successfully sends its last message by step $t_0+\lfloor(R-1)/2\rfloor+6$. If $Q_{i',j',t_0+7} > 0$ then $Q_{i',j'}$ doesn't decide to send on both of steps $\lfloor(R-1)/2\rfloor+7$ and $t_0+\lfloor(R-1)/2\rfloor+8$. Therefore, if $Q_{i,j}$ decides to send on one of these steps it succeeds. Furthermore, every queue $Q_{i''',j'''}$ other than $Q_{i,j}$ only decides to send on step t with $s_{i''',j''',t} \leq R^{2\alpha}$.

6. If $Q_{i,j'}$ and $Q_{i',j}$ are short clear solid candidates then we are in Case 3. Consider the run of the chain for steps $t_0, \ldots, t_0 + 3$ that we described earlier. Suppose that on steps $t_0+4, \ldots, t_0+\lfloor (R-1)/2 \rfloor + 7$ no messages arrive. Suppose that on step t_0+4, every working queue that client-conflicts with $Q_{i,j'}$ or $Q_{i',j}$ decides to send. On step t_0+5, $Q_{i,j'}$ and $Q_{i',j}$ decide to send and every working queue that server-conflicts with one of them decides to send. For t in the range $\{t_0+6, \ldots, t_0+\lfloor (R-1)/2 \rfloor + 6\}$, $Q_{i,j'}$ decides to send on step t if $q_{i,j',t} > 0$ and $Q_{i',j}$ decides to send on step t if $q_{i',j,t} > 0$. On step $t_0+\lfloor (R-1)/2 \rfloor + 7$ $Q_{i,j}$ decides to send if $q_{i,j,t_0+\lfloor (R-1)/2 \rfloor + 7} > 0$. Every other queue only sends if it is forced. The probability of this event is at least $(1-\lambda)^{K(\lfloor (R-1)/2 \rfloor + 7)}(8)^{-KN4}(b_{i,j,t_0} + \lfloor (R-1)/2 \rfloor + 7)^{-\alpha} R^{-2\alpha KN(\lfloor (R-1)/2 \rfloor + 3)}$. On steps $t_0+6, \ldots, t_0+\lfloor (R-1)/2 \rfloor + 7$ nothing that conflicts with $Q_{i,j'}$ or $Q_{i',j}$ decides to send, so they successfully sends their last messages by step $t_0+\lfloor (R-1)/2 \rfloor + 6$. Therefore, if $Q_{i,j}$ decides to send on step $t_0+\lfloor (R-1)/2 \rfloor + 7$ it succeeds. Furthermore, every queue $Q_{i'',j''}$ other than $Q_{i,j}$ only decides to send on step t with $s_{i'',j'',t} \leq R^{2\alpha}$.

7. If $Q_{i',j'}$ is a short clear solid candidate and there is no other solid candidate, then there are many cases. In each case, we will say that a queue *other-conflicts* with $Q_{i,j}$ if it conflicts with $Q_{i,j}$ but not with queue $Q_{i',j'}$. The cases follow.

   7a. No blocking queue other-conflicts with $Q_{i,j}$. We split this case up as follows.

      7a1. No working queue other-conflicts with $Q_{i,j}$. In this case, we are in Case 3. Consider the run of the chain for steps $t_0, \ldots, t_0 + 3$ that we described earlier. Suppose that on steps $t_0 + 4, \ldots, t_0 + \lfloor (R-1)/2 \rfloor + 7$ no messages arrive. Suppose that on step t_0+4, every working queue that client-conflicts with $Q_{i',j'}$ decides to send. On step t_0+5, $Q_{i',j'}$ decides to send and every working queue that server-conflicts with $Q_{i',j'}$ decides to send. For t in the range $\{t_0+6, \ldots, t_0+\lfloor (R-1)/2 \rfloor + 6\}$, $Q_{i',j'}$ decides to send on step t if $q_{i,j',t} > 0$. On step $t_0+\lfloor (R-1)/2 \rfloor + 7$ $Q_{i,j}$ decides to send if $q_{i,j,t_0+\lfloor (R-1)/2 \rfloor + 7} > 0$. Every other queue only sends if it is forced. The probability of this event is at least

      $$(1-\lambda)^{K(\lfloor (R-1)/2 \rfloor + 7)}(8)^{-KN4}(b_{i,j,t_0} + \lfloor (R-1)/2 \rfloor + 7)^{-\alpha} R^{-2\alpha KN(\lfloor (R-1)/2 \rfloor + 3)}.$$

      On steps $t_0+6, \ldots, t_0+\lfloor (R-1)/2 \rfloor + 7$ nothing that conflicts with $Q_{i',j'}$ decides to send, so it successfully sends its last messages by step $t_0+\lfloor (R-1)/2 \rfloor + 6$. Therefore, if $Q_{i,j}$ decides to send on step $t_0+\lfloor (R-1)/2 \rfloor + 7$ it succeeds. Furthermore, every queue $Q_{i'',j''}$ other than $Q_{i,j}$ only decides to send on step t with $s_{i'',j'',t} \leq R^{2\alpha}$.

      7a2. There is a working queue that other-conflicts with $Q_{i,j}$. Every working queue that other-conflicts with $Q_{i,j}$ client-conflicts with another potentially working queue. In this case, we are in Case 3. The proof is the same as that of Case 7a1 except that on step t_0+4 every working queue $Q_{i'',j''}$ that other-conflicts with $Q_{i,j}$ decides to send, and every potentially working queue that client-conflicts with $Q_{i'',j''}$ decides to send.

      7a3. There are two working queues, $Q_{i'',j}$ and $Q_{i''',j}$ that other-conflict (and in particular, server-conflict) with $Q_{i,j}$. Neither of them client-conflicts with a forcing queue. In this case we are in Case 3. The proof is the same as that of Case 7a1 except that on step t_0+4 every working queue that other-conflicts with $Q_{i,j}$ decides to send.

      7a4. There is a working queue $Q_{i'',j}$ that other-conflicts (and in particular, server-conflicts) with $Q_{i,j}$ and does not client-conflict with a potentially working queue. Every other working queue that other-conflicts with $Q_{i,j}$ client-conflicts with a forcing queue. In this Case we are in Case 2. Note that $b_{i'',j,t_0+4} \geq Z$ (otherwise $Q_{i'',j}$ would be blocking). Consider the run of the chain for steps $t_0, \ldots, t_0+3$ that we described earlier.

Suppose that on step t_0+4 no messages arrive and $Q_{i'',j}$ decides to send. No other queue decides to send unless it is forced. The probability of this event is at least $(1-\lambda)^{K5}8^{-KN4}(b_{i,j,t_0}+5)^{-\alpha}2^{-KN}$. $Q_{i'',j}$ succeeds on step t_0+5 and every other queue $Q_{i''',j'''}$ decides to send on step t (for $t \in \{t_0,\ldots,t_0+4\}$) only if $s_{i''',j''',t} \leq 8$.

7a5. If there is a working queue $Q_{i,j''}$ that other-conflicts (and in particular, client-conflicts) with $Q_{i,j}$ and does not client-conflict with a potentially working queue but server-conflicts with a forcing queue then we are in Case 3. The proof is the same as that of Case 7a1 except that on step t_0+4, $Q_{i,j''}$ decides to send and nothing that client-conflicts with the forcing queue decides to send.

7a6. If there is a working queue $Q_{i,j''}$ that other-conflicts (and in particular, client-conflicts) with $Q_{i,j}$ and does not client-conflict with a potentially working queue and does not server-conflict with a forcing queue then we are in Case 2. The proof similar to that of Case 7a4.

7b. There is a blocking queue which other-conflicts with $Q_{i,j}$. We split this case up as follows.

7b1. There are two blocking queues, $Q_{i,j'}$ and $Q_{i,j''}$ that other-conflict (and in particular, client-conflict) with $Q_{i,j}$. In this case, we are in Case 3. The proof is the same as that of Case 7a1 except that on steps $t_0+4,\ldots,t_0+\lfloor(R-1)/2\rfloor+6$, $Q_{i,j'}$ and $Q_{i,j''}$ decide to send, colliding with each other and with any other queues that send that other-conflict with $Q_{i,j}$. On step $t_0+\lfloor(R-1)/2\rfloor+6$, every queue that other-conflicts with $Q_{i,j}$ and is almost forced decides to send.

7b2. There are two blocking queues, $Q_{i',j}$ and $Q_{i'',j}$ that other-conflict (and in particular, server-conflict) with $Q_{i,j}$. There is no blocking queue that client-conflicts with either of these queues. In this case, we are in Case 3. The proof is the same as that of Case 7a1 except that, if there is a working queue that client-conflicts with $Q_{i',j}$ then it sends on step t_0+4 and $Q_{i',j}$ sends on step t_0+4. Otherwise, $Q_{i',j}$ doesn't send on step t_0+4. Similarly, if there is a working queue that client-conflicts with $Q_{i'',j}$ then it sends on step t_0+4 and $Q_{i'',j}$ sends on step t_0+4. Otherwise, $Q_{i'',j}$ doesn't send on step t_0+4. On steps $t_0+5,\ldots,t_0+\lfloor(R-1)/2\rfloor+6$, $Q_{i,j'}$ and $Q_{i,j''}$ decide to send, colliding with each other and with any other queues that decide to send that other-conflict with $Q_{i,j}$. On step $t_0+\lfloor(R-1)/2\rfloor+6$, every queue that other-conflicts with $Q_{i,j}$ and is almost forced decides to send.

7b3. There is a blocking queue $Q_{i',j}$ which other-conflicts (and in particular, server-conflicts) with $Q_{i,j}$ and there is a blocking queue $Q_{i',j'}$. In this case, we are in Case 3. The proof is the same as that of Case 7a1 except that on steps $t_0+4,\ldots,t_0+\lfloor(R-1)/2\rfloor+5$ $Q_{i',j}$ and $Q_{i',j'}$ decide to send. On step $t_0+\lfloor(R-1)/2\rfloor+5$ any queue $Q_{i',j''}$ which is almost forced decides to send. On step $t_0+\lfloor(R-1)/2\rfloor+6$ $Q_{i',j}$ decides to send and any queue $Q_{i'',j}$ which is almost forced decides to send and no other queue $Q_{i',j''}$ decides to send. On step $t_0+\lfloor(R-1)/2\rfloor+7$, $Q_{i',j'}$ decides to send.

7b4. There is a blocking queue $Q_{i,j'}$ which other-conflicts (and in particular, client-conflicts) with $Q_{i,j}$. It does not client-conflict with any other blocking queue. It server-conflicts with the blocking queue $Q_{i',j'}$. $Q_{i',j'}$ does not client-conflict with any other blocking queue. In this case, we are in Case 3. The proof is similar to the proof of Case 7b2.

7b5. There is a blocking queue $Q_{i,j'}$ which other-conflicts (and in particular, client-conflicts) with $Q_{i,j}$. It does not client-conflict with any other blocking queue. It server-conflicts with the blocking queue $Q_{i',j'}$. $Q_{i',j'}$ client-conflicts with blocking queue $Q_{i',j''}$. In this case, we are in Case 3. The proof is the same as that of Case 7a1

except that if there is a working queue that client-conflicts with $Q_{i,j'}$ then it sends on step t_0+4 and $Q_{i,j'}$ sends on step step t_0+4. Otherwise, $Q_{i,j'}$ does not send on step t_0+4. For even ℓ in the range $0 \le \ell \le \lfloor (R-1)/2 \rfloor + 2$, $Q_{i',j'}$ and $Q_{i',j''}$ both send, and so does any queue $Q_{i',j'''}$ which is almost forced. For odd ℓ in the range $0 \le \ell \le \lfloor (R-1)/2 \rfloor + 2$, on step $t_0+4+\ell$, nothing that client-conflicts with $Q_{i',j'}$ sends. $Q_{i',j'}$ and $Q_{i,j'}$ both send.

7b6. There is a short blocking queue $Q_{i',j'}$ that other-conflicts with $Q_{i,j}$. $Q_{i',j'}$ does not conflict with any blocking queues. In this Case, we are in Case 3. The proof is the same as the proof of Case 6, because queue $Q_{i',j'}$ can be treated as a short clear solid candidate.

7b7. There is a long blocking queue $Q_{i',j'}$ that other-conflicts with $Q_{i,j}$. $Q_{i',j'}$ does not conflict with any blocking queues. $b_{i',j',t_0+4} \ge Z$. In this case we are in Case 2. The proof is similar to that of Case 7a4.

7b8. There is a long blocking queue $Q_{i',j'}$ that other-conflicts with $Q_{i,j}$. $Q_{i',j'}$ does not conflict with any blocking queues. $b_{i',j',t_0+4} < Z$. $Q_{i',j'}$ satisfies the following conditions:

1. If $Q_{i'',j'}$ is a working queue then either $q_{i'',j',t_0+4} = 1$ and $\lambda_{i'',j'} \le 1/R^2$, or there is a queue $Q_{i'',j''}$ which does not conflict with a blocking queue and has $s_{i'',j'',t_0+4} = 1$ and

$$q_{i'',j'',t_0+4} \ge \min(R/2, s_{i',j',t_0+4} - 2, s_{i'',j',t_0+4} - 1).$$

2. If $Q_{i',j''}$ is a working queue then either $q_{i',j'',t_0+4} = 1$ and $\lambda_{i',j''} \le 1/R^2$, or there is a queue $Q_{i'',j''}$ which does not conflict with a blocking queue and has $s_{i'',j'',t_0+4} = 1$ and

$$q_{i'',j'',t_0+4} \ge \min(R/2, s_{i',j',t_0+4} - 2, s_{i',j'',t_0+4} - 1).$$

We conclude that $Q_{i',j'}$ is a delayed candidate. Note that if there is a delayed candidate $Q_{i',j'}$ then we can make $Q_{i',j'}$ a delayed control queue. If $Q_{i'',j'}$ is working and $q_{i'',j',t_0+4} > 1$ or $\lambda_{i'',j',t_0+4} > 1/R^2$ then, there is a queue $Q_{i'',j''}$ which does not conflict with a blocking queue and has $s_{i'',j'',t_0+4} = 1$ and $q_{i'',j'',t_0+4} \ge \min(R/2, s_{i',j',t_0+4} - 2, s_{i'',j',t_0+4} - 1)$. We make $Q_{i'',j''}$ a solid control queue if it is long, and a temporary control queue otherwise. (Note that $Q_{i'',j''}$ is blocking, so it doesn't conflict with a candidate.) If $Q_{i',j''}$ is working and $q_{i',j'',t_0+4} > 1$ or $\lambda_{i',j'',t_0+4} > 1/R^2$ then, there is a queue $Q_{i'',j''}$ which does not conflict with a blocking queue and has $s_{i'',j'',t_0+4} = 1$ and $q_{i'',j'',t_0+4} \ge \min(R/2, s_{i',j',t_0+4} - 2, s_{i',j'',t_0+4} - 1)$. As before, we make $Q_{i'',j''}$ a solid control queue if it is long, and a temporary control queue otherwise.

7b9. There is a long blocking queue $Q_{i',j'}$ that other-conflicts with $Q_{i,j}$. $Q_{i',j'}$ does not conflict with any blocking queues. $Q_{i'',j'}$ is a working queue such that ($q_{i'',j',t_0+4} > 1$ or $\lambda_{i'',j} > 1/R^2$) and there is no forced queue $Q_{i'',j''}$. (Or, similarly, $Q_{i',j''}$ is a working queue such that ($q_{i',j'',t_0+4} > 1$ or $\lambda_{i',j''} > 1/R^2$) and there is no forced queue $Q_{i'',j''}$.) Then we are in Case 3. The proof is the same as that of Case 7a1 except that on step t_0+4, $Q_{i'',j'}$ decides to send and nothing that conflicts with it decides to send. As of step t_0+5, $Q_{i'',j'}$ is a blocking queue. Thus, we are in one of the cases 7b1–7b5. (As in Cases 7b2, 7b4 and 7b5, if there is a working queue that client-conflicts with $Q_{i',j'}$ then it sends on step t_0+4 (while $Q_{i'',j'}$ is succeeding) and $Q_{i',j'}$ sends on step t_0+4. Otherwise, $Q_{i',j'}$ doesn't send on step t_0+4. Now if

there is a working queue that client-conflicts with $Q_{i'',j'}$ then we start at step t_0+4 of those cases. Otherwise, we start at step t_0+5.)

7b10. There is a long blocking queue $Q_{i',j'}$ that other-conflicts with $Q_{i,j}$. $Q_{i',j'}$ does not conflict with any blocking queues. $Q_{i'',j'}$ is a working queue such that ($q_{i'',j',t_0+4} > 1$ or $\lambda_{i'',j} > 1/R^2$) and $Q_{i'',j''}$ has $s_{i'',j'',t_0+4} = 1$ but it conflicts with a blocking queue Q'. (Similarly, $Q_{i',j''}$ is a working queue such that ($q_{i',j'',t_0+4} > 1$ or $\lambda_{i',j''} > 1/R^2$) and $Q_{i'',j''}$ has $s_{i'',j'',t_0+4} = 1$ but it conflicts with a blocking queue Q'.) Then we are in Case 3. The proof is the same as that of Case 7a1 except that on step t_0+4, Q' decides to send (and collides with $Q_{i'',j''}$). On step t_0+5 $Q_{i'',j'}$ decides to send and nothing that conflicts with it decides to send. As of step t_0+6, $Q_{i'',j'}$ is a blocking queue. Thus, we are in one of the cases 7b1-7b5 as in case 7b9.

7b11. There is a long blocking queue $Q_{i',j'}$ that other-conflicts with $Q_{i,j}$. $Q_{i',j'}$ does not conflict with any blocking queues. For every working queue $Q_{i'',j'}$ such that $q_{i'',j',t_0+4} > 1$ or $\lambda_{i'',j'} > 1/R^2$ (there is at least one such $Q_{i'',j'}$), there is a forced queue $Q_{i'',j''}$ that does not collide with any blocking queue and has $q_{i'',j'',t_0+4} < \min(R/2, s_{i',j',t_0+4} - 2, s_{i'',j',t_0+4} - 1)$. (Similarly, For every working queue $Q_{i',j''}$ such that $q_{i',j'',t_0+4} > 1$ or $\lambda_{i',j''} > 1/R^2$ (there is at least one such $Q_{i',j''}$), there is a forced queue $Q_{i'',j''}$ that does not collide with any blocking queue and has $q_{i',j'',t_0+4} < \min(R/2, s_{i',j',t_0+4} - 2, s_{i',j'',t_0+4} - 1)$.) Then we are in Case 3. The proof is similar to that of Case 7a1 except that on step t_0+4 all workers $Q_{i''',j'}$ with $q_{i''',j',t_0+4} = 1$ and $\lambda_{i''',j'} \leq 1/R^2$ decide to send. On every step all of the forced queues that are described above decide to send and every working queue that client-conflicts with one of the forced queues and is almost forced decides to send. If one of the forced queues, $Q_{i'',j''}$ has a collision on a step, then, on the next step, $Q_{i'',j'}$ decides to send and none of the queues that conflict with $Q_{i'',j'}$ decides to send. Otherwise, one of the forced queues, $Q_{i'',j''}$ exhausts its queue and on the next step $Q_{i'',j'}$ decides to send and none of the queues that conflict with $Q_{i'',j'}$ decides to send. $Q_{i'',j'}$ is then a blocking queue and so we are in one of the cases 7b1-7b5 as in Case 7b9.

If none of the big backoff counters put us into Case 2 or Case 3, then the control queues that we identify by considering the above cases do not conflict and therefore we can divide the queues into categories such that all of the Case 4 conditions are satisfied.

DISTRIBUTION:

1	Leslie Goldberg Department of Computer Science University of Warwick Coventry CV4 7AL ENGLAND		
1	Phil Mackenzie 805 E. State St. Boise, ID 83712		
1	MS 0321	9200	W. J. Camp
1	MS1111	9221	S. S. Dosanjh
1	MS1110	9205	R. C. Allen, Jr.
5	MS1110	9222	D. E. Womble
5	MS1110	9223	D. S. Greenberg
1	MS1111	9225	G. S. Heffelfinger
1	MS0441	9226	R. W. Leland
2	MS 1436	4523	C. E. Meyers
5	MS 0899	4414	Technical Library (5)
1	MS 9018	8940-2	Central Technical Files
2	MS 0619	12690	Review & Approval Desk (2) for DOE/OSTI