Tree/Stack Splitting with Remainder for Distributed Wireless Medium Access Control with Multipacket Reception

Rung-Hung Gau, *Member, IEEE*

***Abstract*—In this paper, we propose the tree/stack splitting with remainder algorithm for distributed medium access control in a wireless network with multipacket reception. In order to reduce the length of a cycle and increase the network throughput, when the splitting with remainder algorithm is used, some nodes that attempt to transmit packets at the beginning of a cycle might have to postpone their packet retransmissions until the beginning of the next cycle. We demonstrate that the splitting with remainder algorithm outperforms the erasure algorithm and the probe algorithm. For the splitting with remainder algorithm, we analytically and accurately derive the network throughput and the average packet delay. We show that our analytical results are consistent with packet-based simulation results.**

***Index Terms*—Multipacket reception, tree/stack splitting algorithm, medium access control, wireless networks, cross-layer design.**

I. Introduction

In THIS paper, we propose a novel algorithm for medium access control in wireless networks with multipacket reception [5] [11]. According to the conventional $(0, 1, e)$ channel model, when two or more nodes simultaneously send packets to the access point, a collision occurs and the access point does not receive/decode any packets. With multipacket reception capability, when two or more nodes simultaneously send distinct packets to the access point, an access point could successfully receive/decode one or more packets. Therefore, in a wireless network, multipacket reception capability could be used to increase the throughput without requiring more bandwidth, which is usually seen as scarce resource. Multipacket reception can be realized by CDMA multiuser detection techniques [31] or Successive Interference Cancellation [32]. A multipacket reception channel is characterized by a matrix. As the classic splitting algorithm [2] [1], the proposed algorithm can be implemented in a centralized manner based on a tree or a distributed manner based on stacks. Gau and Chen [24] analytically derived the throughput and the average packet delay for the classic tree/stack splitting algorithm when the channel matrix of multipacket reception is a diagonal matrix. The classic tree/stack splitting algorithm is not designed for and therefore does not work in a wireless network with multipacket reception when the channel matrix is not a diagonal matrix. While there are many tree-based medium access control algorithms for the conventional collision channel (and channels with capture effects), to the best of our knowledge, in the literature, there are no distributed medium access control algorithms that are based on tree splitting and work for an arbitrary channel matrix of multipacket reception. In this paper, we study three novel splitting algorithms, each works for a wireless network with an arbitrary channel matrix of multipacket reception. In addition, for each of the three algorithms, it is unnecessary to know/estimate the value of the channel matrix of multipacket reception. Among the three splitting algorithms, the tree/stack splitting with remainder algorithm has the best performance in terms of network throughput and average packet delay. For the splitting with remainder algorithm, we mathematically derive the exact values of the network throughput and the average packet delay. We show that our analytical results are consistent with packet-based simulation results. In addition, in this paper, the network throughput and the average packet delay are functions of the arrival rate of new packets rather than the offered load [1], which depends on the medium access control algorithm.

The rest of the paper is organized as follows. In Section 2, related works are introduced. In Section 3, we present our system models. In Section 4, we introduce three splitting algorithms for medium access control in wireless networks with multipacket reception. In Section 5, we include our analytical results for the splitting with remainder algorithm, which has the best performance among the three splitting algorithms. In Section 6, we show packet-based simulation results and equation-based numerical results. In Section 7, we discuss how to extend our work to more general cases. Our conclusions are included in Section 8.

II. Related Work

Medium access control protocols can be classified into distributed protocols and polling-based protocols. There are three well-known classes of distributed medium access control protocols. They are Splitting, Aloha, and CSMA. Gallager [3] proposed the FCFS splitting algorithm. Garces and Garcia-Luna-Aceves [9] proposed using a deterministic tree-splitting algorithm for nodes in a wireless network to compete for the right to acquire the floor of a particular receiver's channel.

Manuscript received February 15, 2011; revised May 24, July 11, and August 23, 2011; accepted August 23, 2011. The associate editor coordinating the review of this paper and approving it for publication was D. Niyato.

R.-H. Gau is with the Department of Electrical Engineering, National Chiao Tung University, Hsinchu, Taiwan (e-mail: runghung@mail.nctu.edu.tw).

This research is supported in part by the National Science Council, Taiwan, R.O.C. under grant numbers NSC 97-2221-E-110-052-MY3 and NSC 100-2628-E-009-017-MY2.

Digital Object Identifier 10.1109/TWC.2011.083111.110284

Houdt and Blondia [7] proposed combining polling and a variant of the splitting algorithm for contention resolution in a wireless access network. The above works used the conventional $(0, 1, e)$ channel model. Sidi and Cidon [6] analyzed the performance of the FCFS splitting algorithm in channels with capture. Qin and Berry [12] proposed using opportunistic splitting algorithms that exploit channel state information. Yu and Giannakis [19] developed a medium access control protocol that exploits successive interference cancellation over multiple time slots in a tree splitting algorithm. Wang, Yu, and Giannakis [23] proposed combining the binary exponential backoff algorithm with a tree algorithm that relies on successive interference cancellation. In the above two papers, it is assumed that the access point knows the identities of all competing nodes in every time slot. In contrast, we do not make the assumption. Gore and Karandikar [29] proposed a FCFS splitting algorithm that adjusts the transmission power based on quaternary channel feedback. They focused on the capture effect rather than the general multipacket reception capability. Yim, Mehta, Molisch, and Zhang [26] proposed the DPMA (Dual Power Multiple Access) scheme, which is based on FCFS splitting and uses transmission power control to limit the reception power levels to two values that facilitate successive interference cancellation. They studied the case in which the buffer size is infinity and derived the stable throughput. In contrast, we study the case in which the multiple access channel is fully characterized by a matrix and the buffer size is finite.

Chlamtac and Farago [8] studied the multipacket reception capability provided by multiple independent collision channels. Adireddy and Tong [14] studied the use of distributed channel state information for random access in wireless networks with multipacket reception. In particular, they proposed a variant of slotted Aloha in which the transmission probability changes with the channel state. Naware, Mergen, and Tong [15] studied the effect of multipacket reception on stability and delay of slotted Aloha. Luo and Ephremides [16] studied the impacts of transmission power control on the performance of random multiple access with multipacket reception. Gau [17] [21] analytically derived the saturation throughput and the non-saturation throughput for slotted Aloha in wireless networks with multipacket reception. Dua [22] proposed a user-centric approach for evaluating the performance of slotted Aloha with multipacket reception in a wireless network in which the total number of nodes is finite but the buffer size at each node is infinity. Lotfinezhad, Liang, and Sousa [20] derived the optimal retransmission probabilities for slotted Aloha in wireless sensor networks with multipacket reception. Gau [27] used the Poisson random traffic model for performance evaluation of the slotted non-persistent CSMA protocol for wireless networks with multipacket reception. Celik, Zussman, Khan, and Modiano [28] proposed alternative backoff mechanisms for medium access control in wireless networks with multipacket reception capability and spatially distributed nodes. An introduction on CSMA protocols for wireless networks with multipacket reception can be found in [27] [28] and reference therein. Choudhury, Yang, and Vaidya [4] proposed a multi-hop RTS protocol for medium access control in wireless ad hoc networks with directional antennas. Guo, Wang, and Wu [25] studied the capacity of wireless ad hoc networks with multiple packet reception.

An introduction on polling-based protocols of medium access control with multipacket reception can be found in [11] [18] and reference therein. Shad, Todd, Kezys, and Litva [10] introduced dynamic slot allocation for cellular systems with antenna arrays to utilize the multipacket reception capability. Zhao and Tong [11] proposed the MQSR (Multi-Queue Service Room) protocol for medium access control in wireless networks with multipacket reception. In addition, they proposed a dynamic queue protocol [13] that achieves a performance comparable to that of MQSR with a much lower computational complexity. Gau and Chen [18] proposed and analytically evaluated the predictive multicast polling scheme which is a queue-aware medium access control scheme for wireless networks with multipacket reception.

III. System Models

We now state our system models. There are an access point and $N \geq 2$ nodes in the wireless network. Time is divided into time slots of equal length. In addition, the length of a time slot is one. The n-th time slot is the time interval $[n-1, n)$. The transmission time of a packet equals a time slot. In a time slot, a node either sends a packet or does not send any packets. The multipacket reception channel is fully characterized by a $N \times N$ matrix Λ such that $[\Lambda]_{i,j}$ is the probability that the access point will successfully receive/decode j packets in a time slot given that i packets are simultaneously transmitted in the time slot [5] [11]. Let z_i be the probability that the access point does not successfully receive/decode any packets in a time slot, when i nodes simultaneously transmit in the time slot. Then, $z_i = 1 - \sum_{j=1}^{N} [\Lambda]_{i,j} = 1 - \sum_{j=1}^{i} [\Lambda]_{i,j}$. Note that by definition $[\Lambda]_{i,j} = 0$, if $j > i$.

For each i, where $1 \leq i \leq N$, let $\{\eta_i(t), t \geq 0\}$ be a stochastic process such that $\eta_i(t)$ is the total number of packets that arrive at node i during the time interval $[0, t]$. We assume that $\eta_i(0) = 0, \forall i$. Namely, there are no packets in the network at time zero. In addition, we assume that if $i \neq j$, the two stochastic processes $\{\eta_i(t), t \geq 0\}$ and $\{\eta_j(t), t \geq 0\}$ are statistically independent and identical. As in a typical discrete-time system, a packet arriving within a time slot cannot be served until the beginning of the next time slot. Thus, without loss of essential generality, it is assumed that the arrival time of a packet is an integer. Let $G_{i,m}$ be a random variable that represents the total number of new packets that arrive at node i at time m. For each fixed value of i, it is assumed that $G_{i,1}, G_{i,2}, G_{i,3}, ...$ are IID (independent and identically distributed) random variables. Let $f(x) = P\{G_{i,m} = x\}$ be the probability that x packets arrive at a node in a time slot, $\forall x, i, m$. In this paper, we focus on the non-trivial case in which $0 < f(0) < 1$. Let $\lambda = N \cdot \sum_{x=0}^{\infty} x \cdot f(x)$ be the aggregated arrival rate of new packets in the network. In subsequent sections, the arrival rate of packets means the arrival rate of new packets. An admitted packet is retained in the buffer until successfully transmitted to the access point. Namely, retransmissions are used whenever appropriate.

As in [11], it is assumed that the access point knows if the current time slot is idle. There are three types of channel feedback from the access point: *idle*, *erasure*, and *partial success*. When none of the nodes send packets in a time slot, the channel feedback at the end of the time slot is *idle*. When nodes send packets in a time slot but the access point does not successfully receive/decode any packets at the end of the time slot, the channel feedback is *erasure*. To make the splitting with remainder algorithm compatible with a variety of physical-layer technologies that could realize multipacket reception, we assume that in this case the access point only knows that a group of nodes transmitted packets in the current time slot. Namely, the access point does not know the size of the group or the identities of the nodes in the group. Our analysis in this paper remains applicable when the size of the group is known but is not used in the splitting with remainder algorithm. When nodes send packets in a time slot and the access point successfully receives/decodes one or more packets at the end of the time slot, the channel feedback is *partial success*. In this case, the access point also broadcasts the list of nodes that have successfully sent packets to the access point in the current time slot. In addition, to be compatible with a wide range of physical-layer technologies, it is assumed that the access point does not know if any other nodes send but fail to deliver packets to the access point in the current time slot. Namely, even when the access point successfully decodes all packets transmitted in a time slot, the channel feedback is *partial success*. As in [1], it is assumed that channel feedback is immediate and always successfully received by all nodes in the network.

We adopt the convention in queueing theory so that a node is composed of a server and a queue with maximum size B. Therefore, the maximum number of waiting packets at a node equals $(B+1)$. Let $W_n^j \in \{0,1\}$ be a binary random variable such that $W_n^j = 1$ if and only if the server at node j is occupied/busy at time n. A server is said to be occupied by a packet if the packet is transmitting or the packet is waiting for being retransmitted. Let Q_n^j be a random variable that represents the total number of packets in the queue of node j at time n. Then, $0 \le Q_n^j \le B$. When the splitting with remainder algorithm is used, the time axis is partitioned into contention cycles, which will be defined in the next section. In a contention cycle, a node accepts at most B new packets. Admitted packets that arrive within a contention cycle cannot occupy the servers until the beginning of the next contention cycle. At the beginning of a contention cycle, if a node has waiting packets and the corresponding server is not occupied, a packet is moved from the queue to the server. In Figure 1, an example is used to illustrate the above variables. In this case, there are three nodes in the network, $B = 1$, a server is represented by an ellipse, and a queue is represented by a rectangle. At time T_n, all three servers are occupied and therefore $W_{T_n}^1 = W_{T_n}^2 = W_{T_n}^3 = 1$. Meanwhile, there are no packets in the queues of node 1 and node 3 but there is a packet in the queue of node 2. Thus, $Q_{T_n}^1 = Q_{T_n}^3 = 0$ and $Q_{T_n}^2 = 1$. At node 1, in the time interval $[T_n, T_{n+1}]$, a new packet arrives but no packets depart. Therefore, $W_{T_{n+1}}^1 = Q_{T_{n+1}}^1 = 1$. At node 2, in the time interval $[T_n, T_{n+1}]$, no new packets arrive but a packet departs. In addition, at time T_{n+1}, a packet moves from the queue to the server. As a result, $W_{T_{n+1}}^2 = 1$ and $Q_{T_{n+1}}^2 = 0$. At node 3, in the time interval $[T_n, T_{n+1}]$, no new packets arrive but a packet departs. Consequently, $W_{T_{n+1}}^3 = Q_{T_{n+1}}^3 = 0$.

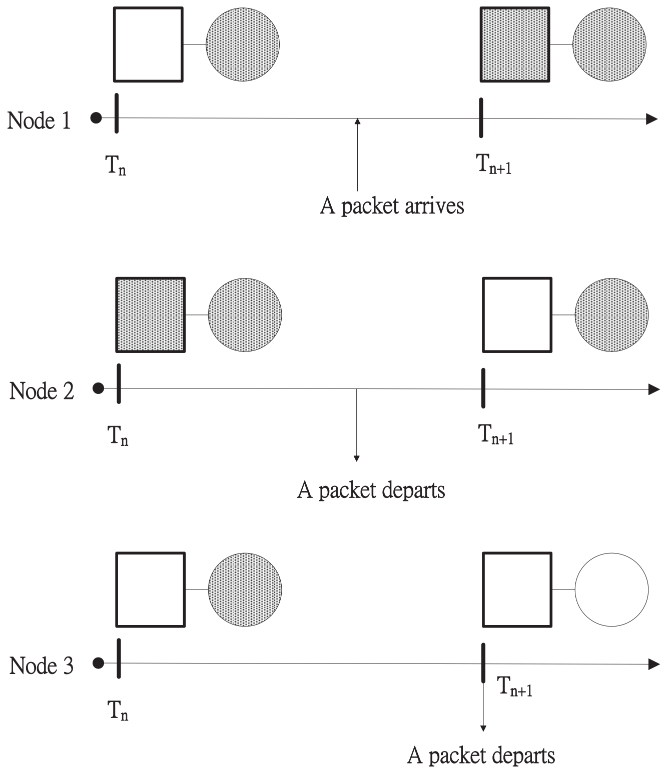


Fig. 1. An example of changes in queue state and server state.

IV. Splitting Algorithms for an Arbitrary Channel Matrix of Multipacket Reception

In this section, we introduce three medium access control algorithms, each works for an arbitrary channel matrix of multipacket reception and is based on the classic tree/stack splitting algorithm, which is not designed for wireless networks with multipacket reception.

A. The Tree/Stack Splitting with Remainder Algorithm

The (tree/stack) splitting with remainder algorithm is based on the classic tree/stack splitting algorithm [2] [1]. In particular, each node has a stack [33] and a pointer [33]. When the splitting with remainder algorithm is used, the time axis is partitioned into contention cycles. Let T_n be the time instance when the n-th contention cycle begins. Note that $T_1 = 0$. At the beginning of a contention cycle, at each node in the network, a single element is pushed into the empty stack. In addition, if the node has waiting packets, the pointer points to the unique element in the stack. In Figure 2, we show an example, which will be explained in detail later in the paper. At the beginning of a time slot, a node with waiting packets sends a packet if and only if its pointer points to the top element in its stack. Let S_m be the total number of nodes that send packets at time m. Let X_n be the total number of nodes with waiting packets at time T_n. Then, $S_{T_n} = X_n$. In addition, define the order of the n-th contention cycle to be X_n.

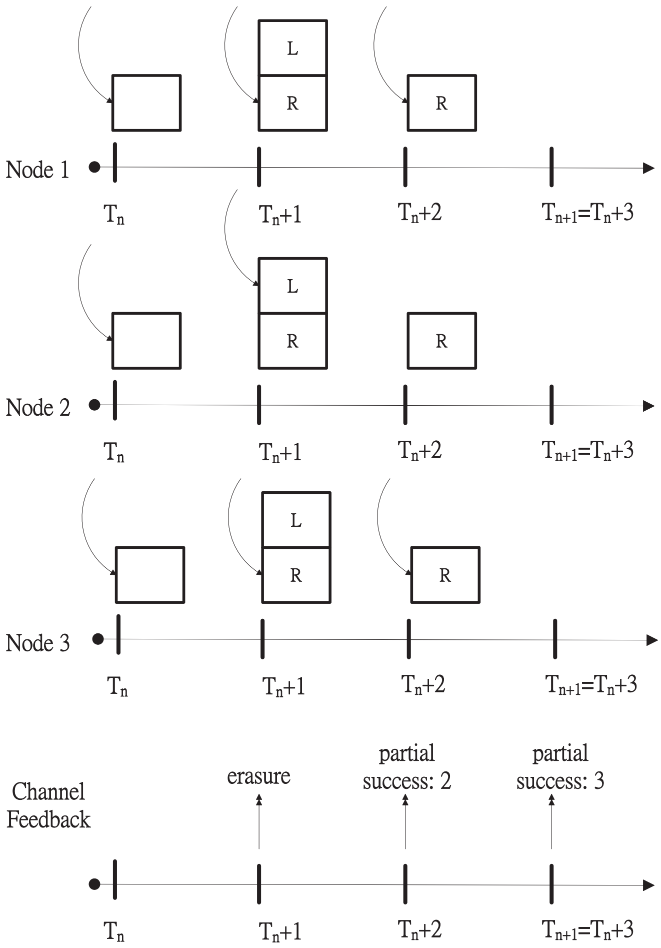


Fig. 2. An example of stack evolution in a contention cycle.

Within a contention cycle, the stacks evolve according to the channel feedback from the access point. If the channel feedback is *idle* or *partial success*, each node removes/pops the top element of its own stack. If the channel feedback at time $m+1$ is *erasure*, stacks in the network evolve according to the following procedure. First, each node in the network removes the top element in its stack and then creates two new elements. The new elements are called the right element and the left element, respectively. Second, each node in the network pushes the two elements into the stack such that the left element is on top of the right element. Last, each node that transmitted a packet at time m independently tosses a fair coin to join either the left subgroup or the right subgroup with equal probabilities. The pointer of each node in the left subgroup points to the left element in its stack. Similarly, the pointer of each node in the right subgroup points to the right element in its stack. A contention cycle terminates once all the stacks become empty. Note that T_{n+1} is the time instance when the n-th contention cycle terminates and the $(n+1)$-th contention cycle begins. Let $Y_n = T_{n+1} - T_n$ be the length of the n-th contention cycle.

When the splitting with remainder algorithm is used, some nodes that transmitted packets at the beginning of a contention cycle might fail to deliver packets to the access point within the contention cycle. Let R_m be the total number of packets that are successfully received by the access point at time m. Define $Z_n = X_n - \sum_{m=T_n+1}^{T_{n+1}} R_m$, $\forall n \geq 1$. Z_n is called the remainder of the n-th contention cycle. Note that Z_n is the total number of nodes that transmit packets at the beginning of the n-th contention cycle but fail to deliver packets to the access point within the time interval $[T_n, T_{n+1}]$. The Z_n nodes will retransmit packets in subsequent contention cycles. As the classic tree/stack splitting algorithm, when the splitting with remainder algorithm is used, for each type of evolution of stacks in a contention cycle, there exists a corresponding splitting tree in which the depth-first-search is used.

We include event-driven pseudo codes for the proposed algorithm in the Appendix. We now use Figure 2 to illustrate the stack evolution when the splitting with remainder algorithm is used. In this case, there are three nodes in the network and each node has waiting packets at time T_n. Therefore, at time T_n, each node pushes an element into the associated stack that was empty. In addition, for each node, the pointer points to the unique element in the stack. Then, all three nodes send packets during the time interval $[T_n, T_n + 1]$. The channel feedback at time $T_n + 1$ is *erasure*, which means that the access point does not receive/decode any packets at time $T_n + 1$. Upon receiving the channel feedback, each node pops the top element in its stack. Then, each node creates (based on the popped element) and pushes the element R and the element L into the stack in this order. In addition, for each node, the pointer points either to the element R in the stack or the element L in the stack with equal probabilities. In this example, the pointer of node 1 points to the element R, the pointer of node 2 points to the element L, and the pointer of node 3 points to the element R. Since the element L is the top element in the stack, only node 2 sends packets during the time interval $[T_n + 1, T_n + 2]$. The channel feedback at time $T_n + 2$ is *partial success: 2*, which means that the access point successfully receives/decodes a single packet from node 2 at time $T_n + 2$. Upon receiving the channel feedback, each node pops the top element in its stack, the element L. Since node 2 has successfully sent a packet to the access point, the pointer of node 2 points to NULL, which means that node 2 will not send any more packets during the n-th contention cycle. During the time interval $[T_n + 2, T_n + 3]$, both node 1 and node 3 send packets. The channel feedback at time $T_n + 3$ is *partial success: 3*, which means that the access point successfully receives/decodes a single packet from node 3. Upon receiving the channel feedback, each node pops the top element in the stack, which becomes empty at time $T_n + 3$. Therefore, at time $T_{n+1} = T_n + 3$, the n-th contention cycle terminates and the $(n+1)$-th contention cycle begins. In this example, $X_n = 3$, $Y_n = T_{n+1} - T_n = 3$, and $Z_n = X_n - 2 = 1$.

The splitting with remainder algorithm is a distributed protocol, since each node decides if it wants to compete for channel access in a slot.

B. The Erasure Algorithm and The Probe Algorithm

We now introduce two variants of the splitting with remainder algorithm. The first variant is called the erasure algorithm while the second variant is called the probe algorithm. Both variants work for an arbitrary channel matrix of multipacket reception. Each variant is very similar to the splitting with

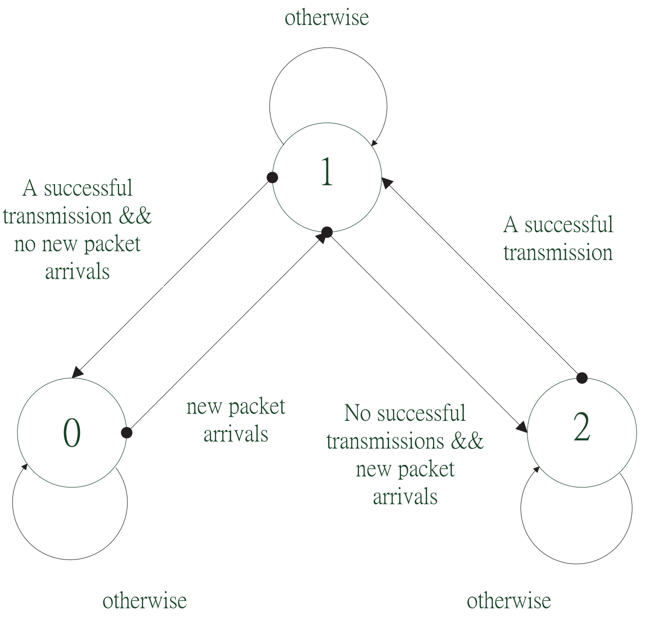

otherwise
1
A successful transmission && no new packet arrivals
A successful transmission
0
new packet arrivals
No successful transmissions && new packet arrivals
2
otherwise
otherwise

Fig. 3. Finite state machine for the system size of a node at the beginning of a contention cycle.

remainder algorithm except for the response to the channel feedback *partial success*. In particular, when the channel feedback is either *idle* or *erasure*, the response of each of the two variants is identical to that of the splitting with remainder algorithm (and that of the classic tree/stack splitting algorithm). For the erasure algorithm, the response when the channel feedback is *partial success* is identical to its own response when the channel feedback is *erasure*. For the probe algorithm, when the channel feedback is *partial success*, all the stacks in the network remain unchanged. In addition, the nodes that have successfully delivered packets to the access point in the current time slot set their pointers to null, while all other nodes keep their pointers unchanged. Therefore, in the subsequent time slot, only the nodes that transmit but fail to deliver packets to the access point in the current time slot will retransmit. If the channel feedback in the subsequent slot is *idle*, all nodes infer that all packet transmissions in the previous slot are successful. One more slot is used to further probe the previous channel status. For example, suppose in the first slot, node 1 and node 2 concurrently transmit packets but the access point only successfully receives/decodes the packet from node 1. If the probe algorithm is used, in the second time slot, only node 2 will retransmit for sure. When the erasure algorithm or the probe algorithm is used, each node that transmits at the beginning of a contention cycle will successfully deliver a packet to the access point within the contention cycle.

V. Analytical Results for The Splitting with Remainder Algorithm

In this section, we derive analytical results for the splitting with remainder algorithm when $B = 1$. Recall that T_n is the time instance when the n-th contention cycle begins. Consider a node. The system size of a node is defined as the total number of packets at the node. In Figure 3, we show how

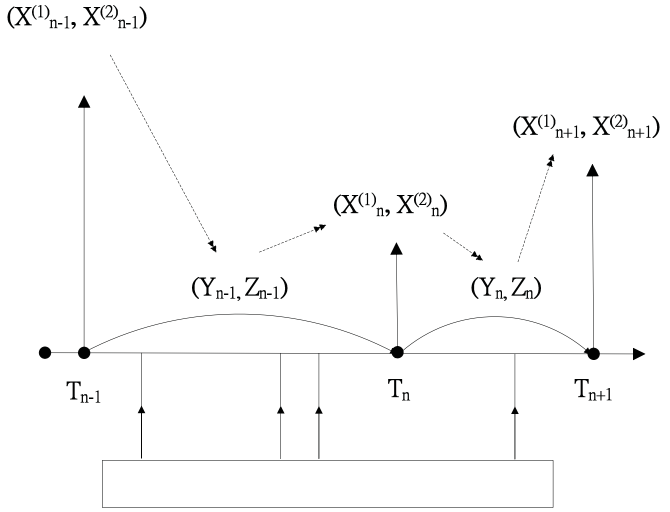

$(X^{(1)}_{n-1}, X^{(2)}_{n-1})$
$(X^{(1)}_{n+1}, X^{(2)}_{n+1})$
$(X^{(1)}_{n}, X^{(2)}_{n})$
(Y_{n-1}, Z_{n-1})
(Y_{n}, Z_{n})
T_{n-1}
T_n
T_{n+1}

Fig. 4. An illustration of modeling the system evolution based on an embedded discrete-time Markov chain.

the system size of the node might change from time T_n to time T_{n+1}. In particular, when the system size at time T_n is zero, the system size at time T_{n+1} will become one if at least one new packet arrives at the node within the time interval $(T_n, T_{n+1}]$. Otherwise, the system size will remain zero. When the system size at time T_n is one, there are three types of state transition. The system size at time T_{n+1} will become zero if within the time interval $(T_n, T_{n+1}]$ the node successfully transmits a packet to the access point and no new packets arrive at the node. The system size at time T_{n+1} will become two if within the time interval $(T_n, T_{n+1}]$ the node fails to deliver a packet to the access point and at least one new packet arrives at the node. Otherwise, the system size will remain one. When the system size at time T_n is two, the system size at time T_{n+1} will become one if the node successfully transmits a packet to the access point within the time interval $(T_n, T_{n+1}]$. Otherwise, the system size will remain two.

A. An Embedded Discrete-Time Markov Chain and Its State Transition Probabilities

Let $X_n^{(k)}$ be the total number of nodes with system size k at time T_n. Since there are N nodes in the wireless network and $B = 1$, $X_n^{(k)} \in \{0, 1, 2, .., N\}$, $\forall n \geq 1, k \in \{0, 1, 2\}$. We use Figure 4 to illustrate how we analyze the system evolution in order to derive the network throughput. First, given the value of $(X_n^{(1)}, X_n^{(2)})$, based on the splitting with remainder algorithm, we can derive the joint probability distribution function of the random variables Y_n and Z_n, where Y_n is the length and Z_n is the remainder of the n-th contention cycle. Next, given the value of $(X_n^{(1)}, X_n^{(2)}, Y_n, Z_n)$, we can derive the joint probability distribution function of the random variables $X_{n+1}^{(1)}$ and $X_{n+1}^{(2)}$. Note that $X_{n+1}^{(1)} + X_{n+1}^{(2)} \geq R_n$. Namely, the total number of nodes that transmit packets at the beginning of the $(n+1)$-th contention cycle is greater than or equal to the remainder of the n-th contention cycle. In addition, some packets that arrived within the n-th contention cycle might be transmitted at the beginning of the $(n+1)$-th contention cycle.

Recall that $X_n = X_n^{(1)} + X_n^{(2)}$. Based on the splitting with remainder algorithm, only the packets that occupy servers at time T_n can contend for channel access during the n-th contention cycle. Therefore, $P\{Y_n = k, Z_n = r | X_n^{(1)} = m_1, X_n^{(2)} = m_2\} = P\{Y_n = k, Z_n = r | X_n = m_1 + m_2\}$. Define $\Phi_1(m, k, r) = P\{Y_n = k, Z_n = r | X_n = m\}$, $\forall n \geq 1$. We now derive the value of $\Phi_1(m, k, r)$. First, $\Phi_1(0, 1, 0) = 1$. For a contention cycle of order zero, with probability one, the length is one and the remainder is zero. Second, $\Phi_1(m, 1, m - k) = [\Lambda]_{m,k}$, $\forall m \geq 1, 1 \leq k \leq m$. Consider a contention cycle of order $m \geq 1$. With probability $[\Lambda]_{m,k}$, the access point successfully receives $k \geq 1$ packets in the first time slot of the contention cycle. If the access point successfully receives $k \geq 1$ packets in the first time slot of the contention cycle, the length of the contention cycle is one and the corresponding remainder is $m - k$. Third, $\forall m \geq 1, k \geq 2, 0 \leq r < m$,

$$\begin{aligned} & \Phi_1(m, k, r) \\ = \ & z_m \times \sum_{n_l=0}^{m} C_{n_l}^{m} \cdot (\frac{1}{2})^m \cdot \sum_{k_l=1}^{k-2} \sum_{r_l=0}^{r} \Phi_1(n_l, k_l, r_l) \times \\ & \Phi_1(m - n_l, k - 1 - k_l, r - r_l) \end{aligned} \quad (1)$$

We now elaborate on the above equation. Recall that z_m in the above equation is the probability that the access point does not successfully receive any packets in a time slot given that m nodes simultaneously transmit packets in the time slot, $\forall m \geq 1$. When $X_n = m$ nodes simultaneously transmit packets at time T_n, the probability that the channel feedback at time $T_n + 1$ is *erasure* equals z_m. If the channel feedback at time $T_n + 1$ is *erasure*, each of the X_n nodes joins either the left subgroup or the right subgroup with equal probabilities. With probability $C_{n_l}^m \cdot (\frac{1}{2})^m$, n_l nodes join the left subgroup, while $m - n_l$ nodes join the right subgroup. Given that there are n_l nodes in the left subgroup, the probability that the length of the contention cycle of the left subgroup is k_l and the remainder of the contention cycle of the left subgroup is r_l equals $\Phi_1(n_l, k_l, r_l)$. Similarly, given that there are $m - n_l$ nodes in the right subgroup, the probability that the length of the contention cycle of the right subgroup is $k - 1 - k_l$ and the remainder of the contention cycle of the right subgroup is $r - r_l$ equals $\Phi_1(m - n_l, k - 1 - k_l, r - r_l)$.

Let $F_n^{(1)}$ be a random variable that represents the total number of nodes, each of system size one at time T_n but fails to successfully deliver packets to the access point within the n-th contention cycle $(T_n, T_{n+1}]$. Let $F_n^{(2)}$ be a random variable that represents the total number of nodes, each of system size two at time T_n but fails to successfully deliver packets to the access point within the n-th contention cycle $(T_n, T_{n+1}]$. By definition, $0 \leq F_n^{(1)} \leq X_n^{(1)}$, $0 \leq F_n^{(2)} \leq X_n^{(2)}$, and $Z_n = F_n^{(1)} + F_n^{(2)}$. Therefore, $\max(Z_n - X_n^{(2)}, 0) \leq F_n^{(1)} \leq \min(Z_n, X_n^{(1)})$. Define $\phi(m_1, m_2, r, r_1) = P\{F_n^{(1)} = r_1 | (X_n^{(1)}, X_n^{(2)}) = (m_1, m_2), Z_n = r\}$, $\forall n \geq 1$. Based on combinatorics, we have

$$\begin{aligned} & \phi(m_1, m_2, r, r_1) = \frac{C_{r_1}^{m_1} \times C_{r - r_1}^{m_2}}{\sum_{g = \max(r - m_2, 0)}^{\min(r, m_1)} C_g^{m_1} \times C_{r-g}^{m_2}}, \\ & \forall \max(r - m_2, 0) \leq r_1 \leq \min(r, m_1) \end{aligned} \quad (2)$$

Furthermore, as long as $P\{Y_n = k, Z_n = r, (X_n^{(1)}, X_n^{(2)}) = (m_1, m_2)\} > 0$, based on symmetry, we have

$$\begin{aligned} & P\{F_n^{(1)} = r_1 | Y_n = k, Z_n = r, (X_n^{(1)}, \\ & X_n^{(2)}) = (m_1, m_2)\} \\ = \ & P\{F_n^{(1)} = r_1 | Z_n = r, (X_n^{(1)}, X_n^{(2)}) = (m_1, m_2)\}, \\ & \forall k \geq 1 \end{aligned} \quad (3)$$

The above equation states that given the value of $(X_n^{(1)}, X_n^{(2)}, Z_n)$, the random variable $F_n^{(1)}$ is statistically independent of the random variable Y_n, the length of the n-th contention cycle.

Define $\Phi_2(m_1, m_2, k, r, n_1, n_2) = P\{(X_{n+1}^{(1)}, X_{n+1}^{(2)}) = (n_1, n_2) | Y_n = k, Z_n = r, (X_n^{(1)}, X_n^{(2)}) = (m_1, m_2)\}$, $\forall n \geq 1$. In addition, define $\Theta_{a,b,n} = |\{j | j \in \{1, 2, 3, .., N\}, W_{T_n}^j + Q_{T_n}^j = a, W_{T_{n+1}}^j + Q_{T_{n+1}}^j = b\}|$, $\forall a, b \in \{0, 1, 2\}, n \geq 1$. These random variables will be used to derive the value of $\Phi_2(m_1, m_2, k, r, n_1, n_2)$. Note that $\Theta_{a,b,n}$ represents the total number of nodes, each has a waiting packets at the beginning of the n-th contention cycle but has b waiting packets at the beginning of the $(n + 1)$-th contention cycle. Define $q = 1 - f(0)$ and $\eta(k) = (1 - q)^k = (f(0))^k$. Namely, for each fixed node, $\eta(k)$ is the probability that no packets arrive at the node in k time slots. Based on Figure 3, we now derive the probability mass functions for $\Theta_{a,b,n}$'s. First,

$$\begin{aligned} & P\{\Theta_{0,1,n} = h | (X_n^{(1)}, X_n^{(2)}) = (m_1, m_2), Y_n = k, \\ & F_n^{(1)} = r_1, F_n^{(2)} = r_2\} \\ = \ & P\{\Theta_{0,1,n} = h | X_n^{(0)} = N - m_1 - m_2, Y_n = k\} \\ = \ & C_h^{N - m_1 - m_2} \cdot (1 - \eta(k))^h \cdot (\eta(k))^{N - m_1 - m_2 - h} \end{aligned} \quad (4)$$

Recall that a node with system size zero at time T_n will become a node with system size one at time T_{n+1} if and only if at least one packet arrives within Y_n time slots. Thus, we have the above equation.

Second,

$$\begin{aligned} & P\{\Theta_{1,0,n} = h | (X_n^{(1)}, X_n^{(2)}) = (m_1, m_2), Y_n = k, \\ & F_n^{(1)} = r_1, F_n^{(2)} = r_2\} \\ = \ & P\{\Theta_{1,0,n} = h | X_n^{(1)} - F_n^{(1)} = m_1 - r_1, Y_n = k\} \\ = \ & C_h^{m_1 - r_1} \cdot (\eta(k))^h \cdot (1 - \eta(k))^{m_1 - r_1 - h} \end{aligned} \quad (5)$$

A node with system size one at time T_n will become a node with system size zero at time T_{n+1} if and only if within the n-th contention cycle, the node successfully transmits a packet to the access point and no new packets arrive at the node.

Similarly,

$$\begin{aligned} & P\{\Theta_{1,2,n} = h | (X_n^{(1)}, X_n^{(2)}) = (m_1, m_2), Y_n = k, \\ & F_n^{(1)} = r_1, F_n^{(2)} = r_2\} \\ = \ & P\{\Theta_{1,2,n} = h | F_n^{(1)} = r_1, Y_n = k\} \\ = \ & C_h^{r_1} \cdot (1 - \eta(k))^h \cdot (\eta(k))^{r_1 - h} \end{aligned} \quad (6)$$

We now derive the value of $\Phi_2(m_1, m_2, k, r, n_1, n_2)$.

$$\begin{aligned}
& \Phi_2(m_1, m_2, k, r, n_1, n_2) \\
= \; & P\{(X_{n+1}^{(1)}, X_{n+1}^{(2)}) = (n_1, n_2) | Y_n = k, Z_n = r, \\
& (X_n^{(1)}, X_n^{(2)}) = (m_1, m_2)\} \\
= \; & \sum_{r_1=\max(r-m_2,0)}^{\min(r,m_1)} P\{F_n^{(1)} = r_1 | Y_n = k, Z_n = r, \\
& (X_n^{(1)}, X_n^{(2)}) = (m_1, m_2)\} \times \\
& P\{(X_{n+1}^{(1)}, X_{n+1}^{(2)}) = (n_1, n_2) | Y_n = k, Z_n = r, \\
& (X_n^{(1)}, X_n^{(2)}) = (m_1, m_2), F_n^{(1)} = r_1\} \\
= \; & \sum_{r_1=\max(r-m_2,0)}^{\min(r,m_1)} P\{F_n^{(1)} = r_1 | Z_n = r, \\
& (X_n^{(1)}, X_n^{(2)}) = (m_1, m_2)\} \times \\
& P\{(X_{n+1}^{(1)}, X_{n+1}^{(2)}) = (n_1, n_2) | Y_n = k, Z_n = r, \\
& (X_n^{(1)}, X_n^{(2)}) = (m_1, m_2), F_n^{(1)} = r_1\} \qquad (7)
\end{aligned}$$

The first equality is based on the definition of $\Phi_2(\cdot)$. The second equality is based on that $P\{Y = y | X = x\} = \sum_z P\{Z = z | X = x\} \times P\{Y = y | Z = z, X = x\}$, for discrete random variables X, Y, and Z. The third equality is based on equation 3.

Based on Figure 3, $X_{n+1}^{(1)} = \Theta_{0,1,n} + (X_n^{(1)} - \Theta_{1,0,n} - \Theta_{1,2,n}) + \Theta_{2,1,n}$, $X_{n+1}^{(2)} = F_n^{(2)} + \Theta_{1,2,n}$, and $\Theta_{2,1,n} = X_n^{(2)} - F_n^{(2)}$. Then, based on equation 7, we have

$$\begin{aligned}
& \Phi_2(m_1, m_2, k, r, n_1, n_2) \\
= \; & \sum_{r_1=\max(r-m_2,0)}^{\min(r,m_1)} \phi(m_1, m_2, r, r_1) \times \\
& P\{\Theta_{1,2,n} = n_2 - r + r_1, \Theta_{0,1,n} - \Theta_{1,0,n} = \\
& n_1 - m_1 + n_2 - m_2 | (X_n^{(1)}, X_n^{(2)}) = (m_1, m_2), \\
& Y_n = k, F_n^{(1)} = r_1, F_n^{(2)} = r - r_1\} \qquad (8)
\end{aligned}$$

Conditioning on $(X_n^{(1)}, X_n^{(2)}, Y_n, F_n^{(1)}, F_n^{(2)})$, the three random variables, $\Theta_{0,1,n}$, $\Theta_{1,0,n}$, and $\Theta_{1,2,n}$, are statistically independent, $\forall n \geq 1$. Therefore, based on equation 8, we have

$$\begin{aligned}
& \Phi_2(m_1, m_2, k, r, n_1, n_2) \\
= \; & \sum_{r_1=\max(r-m_2,0)}^{\min(r,m_1)} \phi(m_1, m_2, r, r_1) \times \\
& P\{\Theta_{1,2,n} = n_2 - r + r_1 | (X_n^{(1)}, X_n^{(2)}) = \\
& (m_1, m_2), Y_n = k, F_n^{(1)} = r_1, F_n^{(2)} = r - r_1\} \times \\
& \sum_{\theta=0}^{m_1-r_1} [P\{\Theta_{1,0,n} = \theta | (X_n^{(1)}, X_n^{(2)}) = (m_1, m_2), \\
& Y_n = k, F_n^{(1)} = r_1, F_n^{(2)} = r - r_1\} \times \\
& P\{\Theta_{0,1,n} = \theta + n_1 - m_1 + n_2 - m_2 | (X_n^{(1)}, X_n^{(2)}) \\
& = (m_1, m_2), Y_n = k, F_n^{(1)} = r_1, F_n^{(2)} = r - r_1\}] \\
& \qquad (9)
\end{aligned}$$

Note that we have derived the values of all the terms in the right-hand side of the above equation.

Let $\Omega = \{(i, j) | i, j \geq 0, i + j \leq N\}$. It can be proved that the discrete-time stochastic process $\{(X_n^{(1)}, X_n^{(2)})\}_{n=1}^{\infty}$ is a discrete-time Markov chain (DTMC) [30] with the finite state space Ω. We now derive the state transition probabilities for the DTMC $\{(X_n^{(1)}, X_n^{(2)})\}_{n=1}^{\infty}$.

$$\begin{aligned}
& P\{(X_{n+1}^{(1)}, X_{n+1}^{(2)}) = (n_1, n_2) | (X_n^{(1)}, X_n^{(2)}) = \\
& (m_1, m_2)\} \\
= \; & \sum_{k=1}^{\infty} \sum_{r=0}^{m_1+m_2} P\{Y_n = k, Z_n = r | (X_n^{(1)}, X_n^{(2)}) = \\
& (m_1, m_2)\} \times P\{(X_{n+1}^{(1)}, X_{n+1}^{(2)}) = (n_1, n_2) | \\
& Y_n = k, Z_n = r, (X_n^{(1)}, X_n^{(2)}) = (m_1, m_2)\} \\
= \; & \sum_{k=1}^{\infty} \sum_{r=0}^{m_1+m_2} \Phi_1(m_1 + m_2, k, r) \times \\
& \Phi_2(m_1, m_2, k, r, n_1, n_2) \qquad (10)
\end{aligned}$$

Define the steady-state probability distribution of the discrete-time Markov chain $\{(X_n^{(1)}, X_n^{(2)})\}_{n=1}^{\infty}$ as follows:

$$\begin{aligned}
& v(m_1, m_2) = \lim_{n \to \infty} P\{(X_n^{(1)}, X_n^{(2)}) = (m_1, m_2)\}, \\
& \forall m_1, m_2 \geq 0, m_1 + m_2 \leq N \qquad (11)
\end{aligned}$$

It can be proved that the DTMC $\{(X_n^{(1)}, X_n^{(2)})\}_{n=1}^{\infty}$ is irreducible, aperiodic, and positive-recurrent. Therefore, the steady-state probability distribution $v(m_1, m_2)$ exists. In addition, $0 < v(m_1, m_2) < 1$, $\forall m_1, m_2 \geq 0, m_1 + m_2 \leq N$. Given the state transition probabilities of the DTMC $\{(X_n^{(1)}, X_n^{(2)})\}_{n=1}^{\infty}$, we can derive the steady-state probability distribution by solving the associated stationary equations [30].

B. The Network Throughput

Let $l(k) = E[Y_n | X_n^{(1)} + X_n^{(2)} = k]$ be the expected value of the length of the n-th contention cycle, given that k nodes concurrently transmit at the beginning of the n-th contention cycle, $\forall n \geq 1$. Note that the value of $E[Y_n | X_n^{(1)} + X_n^{(2)} = k]$ is independent of the value of n. When $X_n^{(1)} + X_n^{(2)} = 0$, $Y_n = 1$ for sure. Thus, $l(0) = 1$. When $1 \leq k \leq N$, similar to equation 1, we have the following recursive equation:

$$l(k) = 1 + z_k \sum_{m=0}^{k} C_m^k \cdot (\frac{1}{2})^k \cdot [l(m) + l(k - m)] \qquad (12)$$

Recall that z_k in the above equation is the probability that the access point does not successfully receive any packets in a time slot given that k nodes simultaneously transmit packets in the time slot, $\forall k \geq 1$. Note that when $k \geq 1$ nodes transmit packets at the beginning of a cycle and the channel feedback of the first time slot of the cycle is *erasure*, the k nodes will be randomly partitioned into two subgroups.

We now use theory of regenerative processes [30] to derive the network throughput as follows. Let A_n be the n-th time instance when the n-th contention cycle of order zero begins. When the splitting with remainder algorithm is used, at the beginning of each contention cycle of order zero, the system statistically regenerates itself. Namely, the network evolution from time A_n is statistically identical to that from time zero.

Thus, $A_2 - A_1$, $A_3 - A_2$, $A_4 - A_3$, ... are IID random variables. Namely, $\{A_n\}_{n=1}^{\infty}$ is a renewal process [30]. Let $E[L] = E[A_{n+1} - A_n]$ be the expected length of a renewal interval. In order to calculate the value of $E[L]$, we create an auxiliary continuous-time stochastic process $\{G(t), t \geq 0\}$ as follows. $G(t) = X_n = X_n^{(1)} + X_n^{(2)}$, if $t \in [T_n, T_n + Y_n)$. Since $\{(X_n^{(1)}, X_n^{(2)})\}_{n=1}^{\infty}$ is a DTMC and Y_n statistically depends only on $(X_n^{(1)}, X_n^{(2)})$, the continuous-time stochastic process $\{G(t), t \geq 0\}$ is a semi-Markov process [30]. Then, based on [30],

$$E[L] \quad = \quad \sum_{m_1=0}^{N} \sum_{m_2=0}^{N-m_1} \frac{v(m_1, m_2)}{v(0,0)} \cdot l(m_1 + m_2) \quad (13)$$

We now elaborate on the above equation. Let $C(m_1, m_2)$ be the expected value of the total number of contention cycles that begins with system state (m_1, m_2) in the renewal interval $[A_n, A_{n+1}]$, $\forall m_1, m_2$. Since $\{(X_n^{(1)}, X_n^{(2)})\}_{n=1}^{\infty}$ is a DTMC, $\frac{v(m_1,m_2)}{v(0,0)} = \frac{C(m_1,m_2)}{C(0,0)}$, $\forall m_1, m_2$. Since A_n is the starting time of the n-th contention cycle that begins with state $(0, 0)$, during a renewal cycle $[A_n, A_{n+1})$, there is a unique contention cycle that begins with state $(0, 0)$ and therefore $C(0, 0) = 1$. Thus, on average, there are $\frac{v(m_1,m_2)}{v(0,0)}$ contention cycles that begins with state (m_1, m_2) in $[A_n, A_{n+1}]$. The average length of a contention cycle that begins with state (m_1, m_2) equals $l(m_1 + m_2)$. Thus, the average length of a renewal cycle equals $\sum_{m_1=0}^{N} \sum_{m_2=0}^{N-m_1} \frac{v(m_1,m_2)}{v(0,0)} \cdot l(m_1 + m_2)$.

Recall that R_m is the total number of packets that are successfully received by the access point at time m. Let λ_D be the network throughput. In particular, λ_D is defined to be $\lim_{n\to\infty} \frac{E[\sum_{m=1}^{n} R_m]}{n}$. Define $u(m) = E[X_n^{(1)} + X_n^{(2)} - Z_n | X_n^{(1)} + X_n^{(2)} = m]$. Recall that Z_n is the remainder of the n-th contention cycle. In addition, $X_n^{(1)} + X_n^{(2)}$ is the total number of nodes that send packets at the beginning of the n-th contention cycle. According to the splitting with remainder algorithm, $u(m)$ is the expected value of the total number of packets that are successfully received by the access point within the n-th contention cycle, given that $X_n^{(1)} + X_n^{(2)} = m$. We now derive the value of $u(m)$. First, it is clear that $u(0) = 0$. On the other hand, $\forall 1 \leq m \leq N$,

$$\begin{aligned} u(m) \quad &= \quad z_m \times \sum_{k=0}^{m} C_k^m \times (\frac{1}{2})^m \times [u(k) + u(m-k)] \\ &\qquad + \sum_{j=1}^{m} [\Lambda]_{m,j} \times j \end{aligned} \quad (14)$$

We now elaborate on the above equation. Recall that z_m is the probability that the access point does not successfully receive any packets in a time slot given that m nodes simultaneously transmit packets in the time slot, $\forall m \geq 1$. When $X_n = m$, with probability z_m, the first channel feedback in the n-th contention cycle is *erasure*. In this case, each of the m nodes either joins the left subgroup or the right subgroup. With probability $C_k^m \cdot (\frac{1}{2})^m$, k nodes join the left subgroup, while $(m - k)$ nodes join the right subgroup. On average, the nodes in the left subgroup successfully deliver $u(k)$ packets to the access point within the n-th contention cycle. Similarly, the nodes in the right subgroup successfully deliver $u(m - k)$ packets to the access point within the n-th contention cycle. On the other hand, when $X_n = m$, with probability $[\Lambda]_{m,j}$, in the first time slot of the n-th contention cycle, the access point successfully receives $j \geq 1$ packets.

When the splitting with remainder algorithm is used, at the beginning of each contention cycle of order zero, the system statistically regenerates itself. Therefore, $\{R_n\}_{n=1}^{\infty}$ is a regenerative process with respect to the renewal process $\{A_n\}_{n=1}^{\infty}$. In a regenerative cycle $(A_n, A_{n+1}]$, on average, there are $\frac{v(m_1,m_2)}{v(0,0)}$ contention cycles, each begins with state (m_1, m_2). In a contention cycle that begins with state (m_1, m_2), on average, the access point successfully receives $u(m_1 + m_2)$ packets from $(m_1 + m_2)$ nodes. Therefore, on average, the access point successfully receives $\sum_{m_1=0}^{N} \sum_{m_2=0}^{N-m_1} \frac{v(m_1,m_2) \cdot u(m_1+m_2)}{v(0,0)}$ packets in a regenerative cycle. According to the theory of regenerative processes, the network throughput equals the expected value of the total number of successfully received packets at the access point in a regenerative cycle divided by the expected value of the length of a regenerative cycle. Thus, we have the following formula for calculating the network throughput.

$$\lambda_D = \frac{1}{E[L]} \sum_{m_1=0}^{N} \sum_{m_2=0}^{N-m_1} (\frac{v(m_1, m_1)}{v(0,0)}) \cdot u(m_1 + m_2) \quad (15)$$

C. The Average System Size and The Average Packet Delay

Recall that N is the total number of nodes in the wireless network. Let W_n be a random variable that represents the total number of busy/occupied servers in the network at time n. In particular, $W_n = \sum_{j=1}^{N} W_n^j$. When the splitting with remainder algorithm is used, at the beginning of each contention cycle that begins with state $(0, 0)$, the system statistically regenerates itself. Therefore, the stochastic process $\{W_n\}_{n=0}^{\infty}$ is a regenerative process with respect to the renewal process $\{A_n\}_{n=1}^{\infty}$. Let $E[W]$ be the long-term average number of busy servers in the network in a time slot. In particular, $E[W] = \lim_{n\to\infty} \frac{E[\sum_{m=0}^{n-1} W_m]}{n}$.

Let $\alpha(n)$ be an integer such that $X_{\alpha(n)} = X_{\alpha(n)}^{(1)} + X_{\alpha(n)}^{(2)} = n$. Namely, the order of the $\alpha(n)$-th contention cycle equals n. Let $s_1(n)$ be the average number of busy servers during a contention cycle of order n. In particular, $s_1(n) = \frac{1}{E[Y_{\alpha(n)}]} \cdot E[\sum_{m=T_{\alpha(n)}}^{T_{\alpha(n)+1}-1} W_m] = \frac{1}{l(n)} \cdot E[\sum_{m=T_{\alpha(n)}}^{T_{\alpha(n)+1}-1} W_m]$. Define $\xi(k, r) = E[Z_n = r | X_n = k]$, $\forall n \geq 1$. Recall that $\Phi_1(k, y, r) = E[Y_n = y, Z_n = r | X_n = k]$, $\forall n \geq 1$. Then, $\xi(k, r) = \sum_{y=1}^{\infty} \Phi_1(k, y, r)$. We now derive the value of $s_1(m)$. First, $s_1(0) = 0$. This is due to that when $X_n = 0$, $Y_n = 1$ and $W_{T_n} = 0$ for sure. In addition, $\forall 1 \leq m \leq N$, we have

$$\begin{aligned} & s_1(m) \cdot l(m) \\ = \quad & m \cdot 1 + z_m \sum_{k=0}^{m} C_k^m \cdot (\frac{1}{2})^m \cdot \{[s_1(k) \cdot l(k) + \\ & (m-k) \cdot l(k)] + [s_1(m-k) \cdot l(m-k) + \\ & \sum_{r=0}^{k} \xi(k, r) \cdot r \cdot l(m-k)]\} \end{aligned} \quad (16)$$

We now elaborate on the above equation. By definition, $s_1(m) \cdot l(m)$ is the expected value of the sum of the number of busy/occupied servers over a contention cycle of order m. By definition, the total number of busy servers in the first time slot of a contention cycle of order m is m. Recall that z_m is the probability that the access point does not successfully receive any packets in a time slot given that m nodes simultaneously transmit packets in the time slot, $\forall m \geq 1$. With probability z_m, the first channel feedback in a contention cycle of order m is *erasure*. In this case, a contention cycle contains two time intervals, in addition to the first time slot. The two time intervals are called the left time interval and the right time interval, respectively. With probability $C_k^m \cdot (\frac{1}{2})^m$, there are k nodes in the left subgroup and $(m-k)$ nodes in the right subgroup. Given that there are k nodes in the left subgroup and $(m-k)$ nodes in the right subgroup, the expected value of the sum of the number of busy servers over the left time interval of the contention cycle equals $s_1(k)l(k) + (m-k)l(k)$. In addition, with probability $\xi(k,r)$, the remainder of the splitting with remainder algorithm associated with the left subgroup equals r. Therefore, the expected value of the sum of the total number of busy servers over the right time interval of the contention cycle equals $s_1(m-k)l(m-k) + \sum_{r=0}^{k} \xi(k,r) \cdot r \cdot l(m-k)$.

According to theory of regenerative processes, the expected value of the number of busy servers in the network in a time slot equals the expected value of the sum of the number of busy servers over a regenerative cycle divided by the expected value of the length of a regenerative cycle. Thus, similar to the derivation of the network throughput, we have

$$\begin{aligned} E[W] &= \frac{1}{E[L]} \sum_{m_1=0}^{N} \sum_{m_2=0}^{N-m_1} \{(\frac{v(m_1,m_2)}{v(0,0)}) \times \\ & \qquad s_1(m_1+m_2) \times l(m_1+m_2)\} \end{aligned} \tag{17}$$

Let $E[Q]$ be the expected value of the total number of waiting packets in the N queues of the network in a time slot. Let $s_2(n)$ be the average queue size at a node during a contention cycle of order n, when the queue size of the node at the beginning of the contention cycle is zero. We now derive the value of $s_2(n)$. First, $s_2(0) = 0$. This is due to that when $X_m = 0$, $Y_m = 1$ for sure. Define $g(m,k) = P\{Y_n = k | X_n = m\}$, $\forall n \geq 1$. Then, $g(m,k) = \sum_{r=0}^{m} \Phi_1(m,k,r)$, $\forall m,k$. Recall that for a node, q is the probability that at least one new packet arrives at the node within a time slot. Consider a tagged node with queue size zero at time T_n. $g(m,y)$ is the probability that the length of a contention cycle of order m equals y. For the tagged node, $(1-q)^{k-1} \cdot q$ is the probability that the earliest packet arrival time after time T_n is $T_n + k$. Given that $Y_n = y$ and the earliest packet arrival time after time T_n is $T_n + k$, the sum of queue size over the n-th contention cycle is $(y-k)$. Therefore,

$$\begin{aligned} s_2(m) \cdot l(m) &= \sum_{y=1}^{\infty} g(m,y) \sum_{k=1}^{y-1} [(1-q)^{k-1} \cdot q] \times \\ & \qquad (y-k), \forall 1 \leq m \leq N \end{aligned} \tag{18}$$

When $B = 1$, there are at most two waiting packets at a node. If there are two waiting packets at a node at the beginning of a contention cycle, the queue size of the node is always one within the contention cycle. According to theory of regenerative processes, the expected value of the queue size in a time slot equals the expected value of the sum of the queue size over a regenerative cycle divided by the expected value of the length of a regenerative cycle. Thus, similar to the derivation of the network throughput, we have

$$\begin{aligned} & E[Q] \\ = \; & \frac{1}{E[L]} \sum_{m_1=0}^{N} \sum_{m_2=0}^{N-m_1} \{(\frac{v(m_1,m_2)}{v(0,0)}) \times [(N-m_2) \times \\ & s_2(m_1+m_2) \times l(m_1+m_2) + \\ & m_2 \times 1 \times l(m_1+m_2)]\} \end{aligned} \tag{19}$$

Let $E[D]$ be the average system delay. The system delay of a packet is the total amount of time that a packet stays in the system. Note that the system delay includes the queueing delay and the retransmission delay. Then, according to the well-known Little's formula [30], we have $E[D] = \frac{E[W]+E[Q]}{\lambda_D}$.

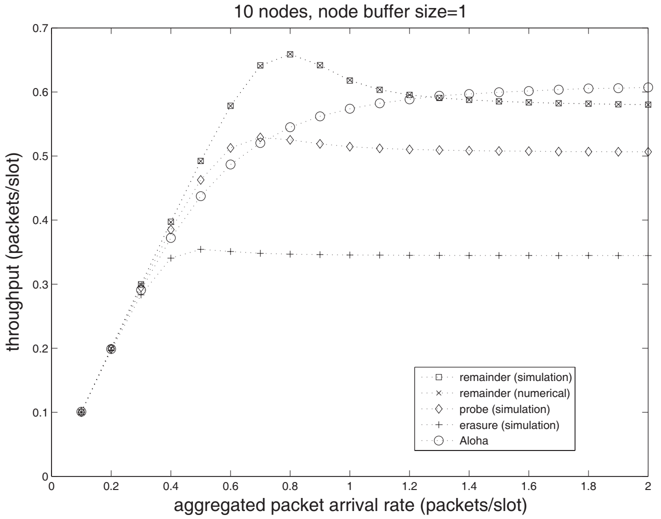


Fig. 5. The throughput of the splitting algorithms for channel 1, 10 nodes.

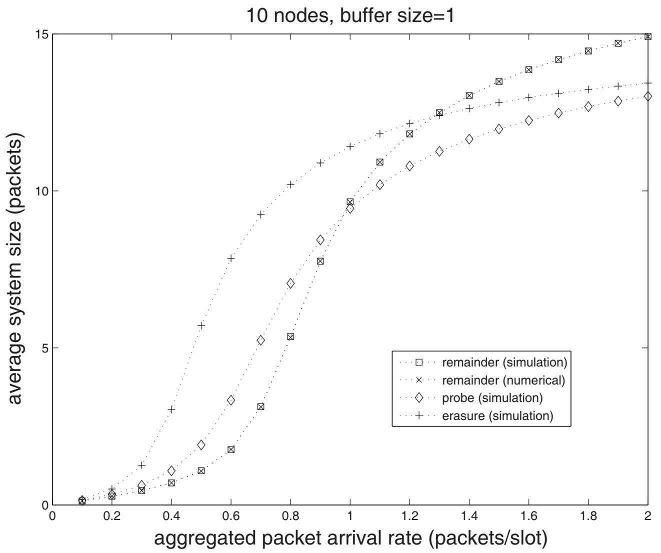


Fig. 6. The average system size of the splitting algorithms for channel 1, 10 nodes.

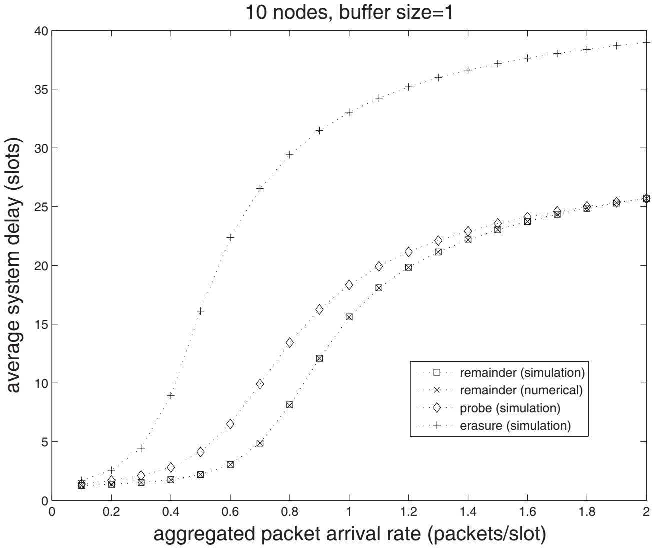


Fig. 7. The average system delay of the splitting algorithms for channel 1, 10 nodes.

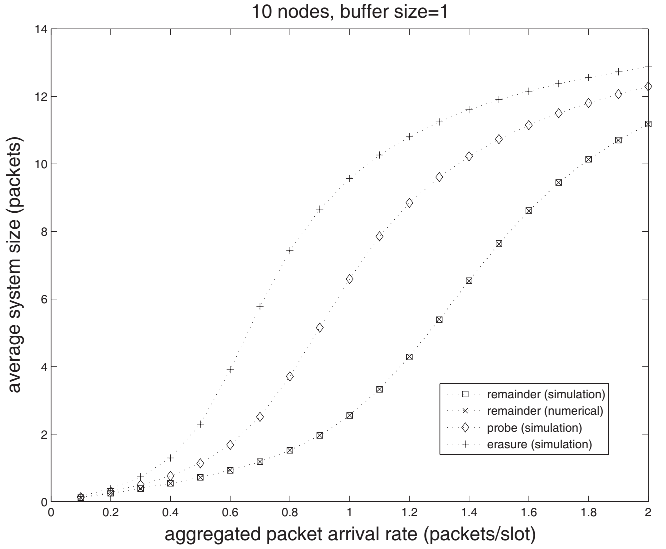


Fig. 9. The average system size of the splitting algorithms for channel 2, 10 nodes.

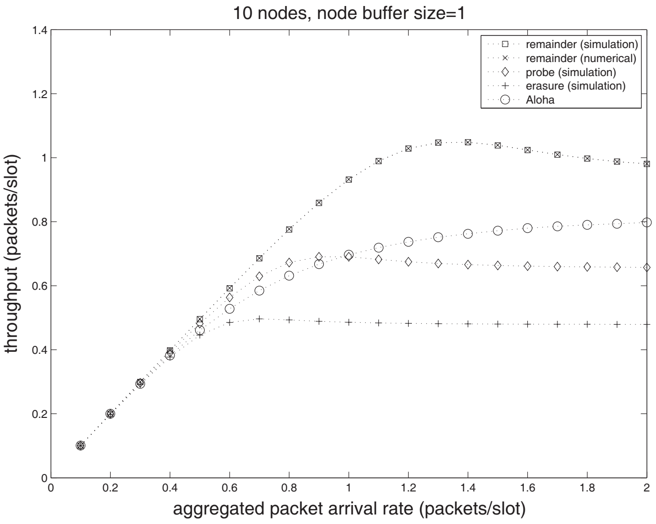


Fig. 8. The throughput of the splitting algorithms for channel 2, 10 nodes.

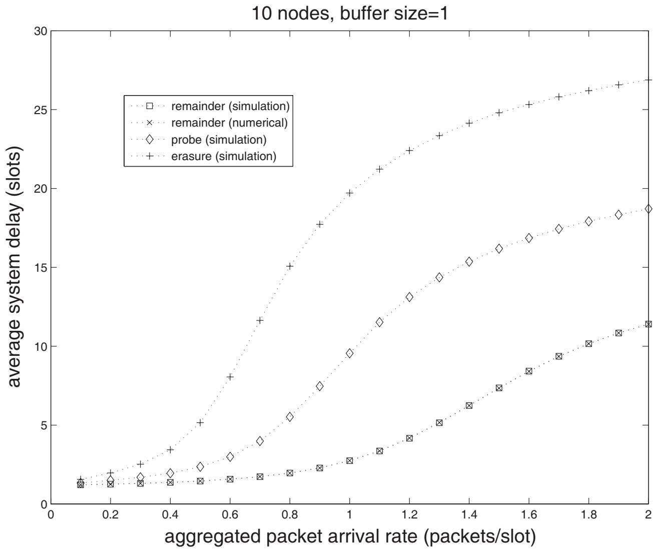


Fig. 10. The average system delay of the splitting algorithms for channel 2, 10 nodes.

VI. Numerical and Simulation Results

We wrote one C program to derive numerical results based on the equations in the paper and three C programs to perform packet-based simulation for the three tree-based algorithms. For each simulation instance, there are one million contention cycles. In addition, we wrote a C program to perform packet-based simulation for the slotted Aloha algorithm with retransmission probability equals $\frac{1}{N}$, where N is the total number of nodes in the network. For illustration purposes, we studied the cases in which $N = 10$, $B = 1$, and $f(1) = 1 - f(0)$. Recall that Λ is the channel matrix. We first study the case when $[\Lambda]_{1,1} = 0.9$, $[\Lambda]_{2,1} = 0.8$, $[\Lambda]_{2,2} = 0.1$, $[\Lambda]_{3,1} = 0.7$, $[\Lambda]_{3,2} = 0.1$, $[\Lambda]_{3,3} = 0.1$, and $z_i = 1$, $\forall i \geq 4$. In this case, when three nodes concurrently transmit packets, most likely, the access point will successfully receive only one packet and the channel feedback will be *partial success*.

In Figure 5, we show the network throughput for the three tree-based algorithms and the slotted Aloha algorithm. As shown in the figure, for the network throughput, our equation-based numerical results are consistent with packet-based simulation results. When the aggregated packet arrival rate is no greater than 1.2, in terms of throughput, the splitting with remainder algorithm outperforms the slotted Aloha algorithm. Among the three tree-based algorithms, regardless of the aggregated packet arrival rate, the splitting with remainder algorithm always has the largest throughput, while the erasure algorithm always has the smallest throughput. For example, when the total arrival rate of new packets is 0.8, the throughput for the splitting with remainder algorithm is about 0.618, the throughput for the probe algorithm is about 0.525, the throughput for the erasure algorithm is about 0.346, and the throughput for the slotted Aloha algorithm is about 0.5449. We now explain why the splitting with remainder algorithm is superior to the other two tree-based algorithms. Consider a time slot in which the channel feedback is *partial success*.

Since $z_i = 1$, $\forall i \geq 4$, we know for sure that at most three nodes send packets in the time slot. Note that *partial success* implies that the access point successfully receives at least one packet. Suppose that in fact only a node sends a packet in the time slot. In this case, when the erasure algorithm is used, each node will waste two time slots before reducing the stack size by one. When the probe algorithm is used, each node will waste one time slot before reducing the stack size by one. In contrast, when the splitting with remainder algorithm is used, each node immediately reduces the stack size. When the aggregated packet arrival rate is small, it is very likely that only one node sends a packet in a time slot. Therefore, the splitting with remainder algorithm outperforms the erasure algorithm and the probe algorithm, when the network load is small.

In Figure 6, we show the average system size for the three tree-based algorithms. In particular, when the total arrival rate of new packets is greater than or equal to 1.0, the average system size of the splitting with remainder algorithm is larger than that of the probe algorithm. On the other hand, when the total arrival rate of new packets is less than 1.0, the average system size of the splitting with remainder algorithm is smaller than that of the probe algorithm. In Figure 7, we show the average packet delay for the three tree-based algorithms. In particular, among the three tree-based algorithms, the splitting with remainder algorithm always has the smallest expected value of packet delay, when the total arrival rate is less than 2.0. As shown in the figure, for the average packet delay, our equation-based numerical results are consistent with packet-based simulation results.

We also study the case when $[\Lambda]_{1,1} = 0.9$, $[\Lambda]_{2,1} = 0.1$, $[\Lambda]_{2,2} = 0.8$, $[\Lambda]_{3,1} = 0.1$, $[\Lambda]_{3,2} = 0.1$, $[\Lambda]_{3,3} = 0.7$, and $z_i = 1$, $\forall i \geq 4$. In comparison with the previous channel matrix, this channel matrix is more similar to a diagonal matrix with rank three. In Figure 8, we show the network throughput for the three tree-based algorithms and the slotted Aloha algorithm. When the aggregated packet arrival rate is no greater than 3.0, in terms of throughput, the splitting with remainder algorithm outperforms the slotted Aloha algorithm. For example, when the total arrival rate of new packets is 1.4, the throughput for the splitting with remainder algorithm is about 1.048, the throughput for the probe algorithm is about 0.666, the throughput for the erasure algorithm is about 0.480, and the throughput for the slotted Aloha algorithm is about 0.7624. In Figure 9, we show the average system size for the three tree-based algorithms. In Figure 10, we show the average packet delay for the three tree-based algorithms. For this channel matrix, among the three tree-based algorithms, the splitting with remainder algorithm always has the largest network throughput, the smallest average system size, and the smallest expected value of packet delay.

VII. Discussions

In this section, we analyze the computational complexity of the proposed splitting with remainder algorithm. In addition, we discuss how to extend the above work to more general cases.

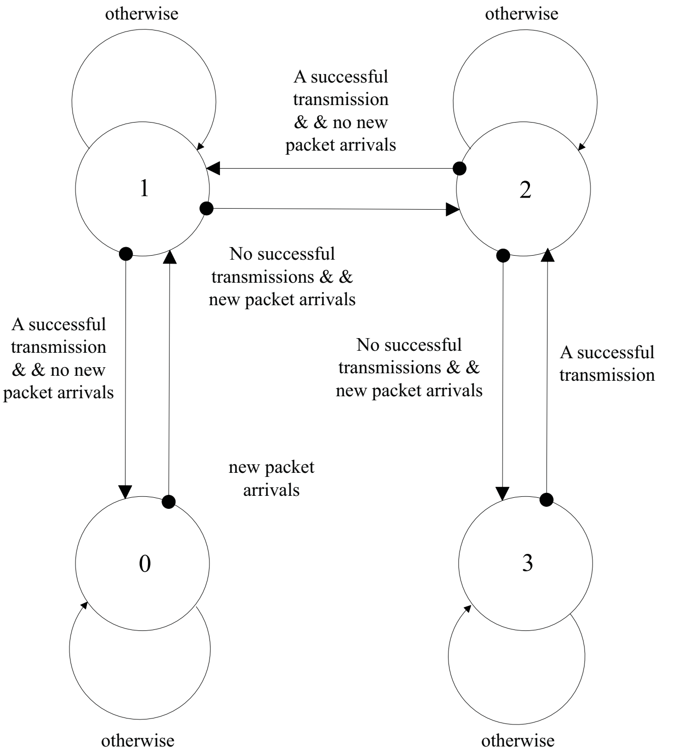

otherwise
otherwise
A successful transmission & & no new packet arrivals
1
2
No successful transmissions & & new packet arrivals
A successful transmission & & no new packet arrivals
No successful transmissions & & new packet arrivals
A successful transmission
new packet arrivals
0
3
otherwise
otherwise

Fig. 11. Finite state machine for the system size of a node at the beginning of a contention cycle, when $B = 2$.

A. Computational Complexity

We now analyze the computational complexity of the proposed splitting with remainder algorithm. Note that the proposed algorithm is a distributed algorithm. In particular, each node listens to the channel feedback from the access point but two nodes do not exchange any control messages. A packet of channel feedback is composed of two parts. The first part represents the class of the channel feedback, while the second part is used only when the channel feedback is *partial success*. In particular, the second part is composed of N bits such that the i-th bit is set to one if and only if the access point has successfully received a packet from node i. There are three classes of channel feedback and $\lceil \log_2(3) \rceil = 2$. Therefore, the length of a channel feedback packet is $(N+2)$ bits. For a node, it takes $O(1)$ time to create an element, to pop an element from the stack, to push an element into the stack, or to update the stack pointer. Therefore, given the channel feedback, it takes a node $O(1)$ time to update the stack and the pointer. Let $H_{i,t}$ be a random variable that represents the total number of elements in the stack of node i at time t. In general, $H_{i,t}$ is unbounded. We now show that $E[H_{i,t}] < \infty$, $\forall i, t$. According to the splitting with remainder algorithm, $H_{i,t+1} - H_{i,t} \leq 1$. Let $c(t)$ be the integer such that $T_{c(t)} \leq t < T_{c(t)+1}$. Then, $H_{i,t} \leq Y_{c(t)}$. Recall that $l(m) < \infty$ is the average length of a contention cycle of order m. Then, $E[H_{i,t}] \leq \max_{m:1 \leq m \leq N} l(m)$. Since $l(m)$ is an increasing function of m, $E[H_{i,t}] \leq l(N) < \infty$. Namely, the average space requirement is upper bounded by $l(N)$.

B. When $B = 2$

We now study the case when $B = 2$. In Figure 11, we show how the system size of a node (in terms of total number of waiting packets) might change from time T_n to time T_{n+1}, when $B = 2$. Let $X_n^{(k)}$ be the total number of nodes, each has k packets at time T_n, $\forall n \geq 1, k \in \{0, 1, 2, .., B+1\}$. Let $\mathbf{Z}^+$ be the set composed of all non-negative integers. Similar to the case in which $B = 1$, it can be proved that when $B = 2$, $\{(X_n^{(1)}, X_n^{(2)}, X_n^{(3)})\}_{n=1}^{\infty}$ is a discrete-time Markov chain with a finite state space $\Omega_{1,2} = \{(x_1, x_2, x_3)|x_1 + x_2 + x_3 \leq N, x_1, x_2, x_3 \in \mathbf{Z}^+\}$. Following the procedures in section V.A and making necessary modifications, in principle, we can derive the state transition probabilities for the DTMC $\{(X_n^{(1)}, X_n^{(2)}, X_n^{(3)})\}_{n=1}^{\infty}$. Due to limit of space, we have to leave the lengthy details as future work. Define $v^{(2)}(m_1, m_2, m_3) = \lim_{n\rightarrow\infty} P\{(X_n^{(1)}, X_n^{(2)}, X_n^{(3)}) = (m_1, m_2, m_3)\}$, $\forall (m_1, m_2, m_3) \in \Omega_{1,2}$. Let $\lambda_D^{(2)}$ be the network throughput, when $B = 2$. Define $\widehat{N} = \sum_{(m_1,m_2,m_3):(m_1,m_2,m_3)\in\Omega_{1,2}} v^{(2)}(m_1, m_2, m_3) \times u(m_1 + m_2 + m_3)$ and $\widehat{D} = \sum_{(m_1,m_2,m_3):(m_1,m_2,m_3)\in\Omega_{1,2}}$ $v^{(2)}(m_1, m_2, m_3) \times l(m_1+m_2+m_3)$. Then, similar to equation 13 and equation 15, we have $\lambda_D^{(2)} = \frac{\widehat{N}}{\widehat{D}}$. Note that $l(k)$ has been derived in equation 12 and $u(k)$ has been derived in equation 14.

C. Heterogeneous Source Traffic

We now study the case in which $B = 1$ but the nodes are classified into two types according to their packet arrival rates. Let λ_1 and λ_2 be two distinct positive real numbers. In particular, for a type-i node, the arrival rate of new packets equals λ_i, $\forall i \in \{1, 2\}$. Let N_i be the total number of nodes in class i, $\forall i \in \{1, 2\}$. Let $X_n^{(i,k)}$ be the total number of type-i nodes, each has k packets at time T_n, $\forall n \geq 1, i \in \{1, 2\}, k \in \{0, 1, 2, .., B+1\}$. Then, it can be proved that $\{(X_n^{(1,0)}, X_n^{(1,1)}, X_n^{(1,2)}, X_n^{(2,1)}, X_n^{(2,2)})\}_{n=1}^{\infty}$ is a discrete-time Markov chain with a finite state space $\Omega_{2,1} = \{(x_{1,0}, x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2})|x_{1,0} + x_{1,1} + x_{1,2} = N_1, x_{2,1} + x_{2,2} \leq N_2, x_{1,0}, x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2} \in \mathbf{Z}^+\}$. In addition, for a node, the state transition diagram is similar to that in Figure 3. Note that when the arrival rates of new packets for all the nodes are identical, the state of the network at the beginning of a contention cycle is represented by a two-dimensional vector. In contrast, when the nodes are classified into two types according to their packet arrival rates, the state of the network at the beginning of a contention cycle is represented by a five-dimensional vector. Thus, if the Gauss-Seidel method is used to solve the stationary equation, the computational complexity for throughput calculation is at least $O((N_1^2 \cdot N_2^2)^2) = O((N_1 \cdot N_2)^4)$. Therefore, it may be desired to have low-complexity algorithms that produce good approximations.

D. On Deriving Approximations with Lower Computational Complexity

In this subsection, we outline an approach for deriving approximations for the network performance with lower computational complexity, when the splitting with remainder algorithm is used and $B \geq 2$. Recall that X_n is the total number of nodes with waiting packets at the beginning of the n-th contention cycle and Z_n is the remainder of the n-th contention cycle. Let Ξ_n be the total number of waiting packets in the queues of the nodes in the network at the beginning of the n-th contention cycle. Let q_n and r_n be the integers such that $\Xi_n = q_n \cdot X_n + r_n$ and $0 \leq r_n < X_n$. To efficiently derive approximations for the network performance, it is assumed that the system is homogenized at the beginning of each contention cycle. As a result, at the beginning of the n-th contention cycle, among the X_n nonempty nodes, there are r_n nodes, each has $q_n + 1$ waiting packets. In addition, at the beginning of the n-th contention cycle, there are $(X_n - r_n)$ nodes, each has q_n waiting packets. The state at the beginning of the n-th contention cycle is represented by (X_n, Ξ_n).

We now derive relations between (X_n, Ξ_n) and (X_{n+1}, Ξ_{n+1}). Let $M_{0,n}$ be the total number of new packets that arrive at the $N - X_n$ nodes, each was empty at the beginning of the n-th contention cycle, during the n-th contention cycle. Let $M_{1,n}$ be the total number of new packets that arrive at the X_n nodes, each had waiting packets at the beginning of the n-th contention cycle, during the n-th contention cycle. Then,

$$X_{n+1} = Z_n + \min(X_n - Z_n, \Xi_n + M_{1,n}) + \min(N - X_n, M_{0,n}) \quad (20)$$

We now elaborate on the above equation. The Z_n nodes that transmit packets at the beginning of the n-th contention cycle but fail to deliver packets to the access point during the n-th contention cycle will compete for channel access at the beginning of the $(n+1)$-th contention cycle. Therefore, we have the first term in the right-hand side of the equation. The second term is due to that for the $X_n - Z_n$ nodes that successfully deliver packets to the access point during the n-th contention cycle, packets will be moved from the queues to the servers at the end of the n-th contention cycle, as long as there are enough packets in the queues. Note that $\Xi_n + M_{1,n}$ is the total number of packets in the queues of the X_n nodes just before the end of the n-th contention cycle. The third term is due to that for the $N - X_n$ nodes that are empty at the beginning of the n-th contention cycle, packets will be moved from the queues to the servers at the end of the n-th contention cycle, as long as enough new packets arrive at the $N - X_n$ nodes during the n-th contention cycle. Note that $M_{0,n}$ is the total number of packets that arrive at the $N - X_n$ nodes during the n-th contention cycle.

In addition, since $X_n + \Xi_n$ is the total number of packets in the network at the beginning of the n-th contention cycle, $M_{0,n} + M_{1,n}$ is the total number of new packet arrivals during the n-th contention cycle, and $X_n - Z_n$ is the total number of packets that depart from the network during the n-th contention cycle, we have $X_{n+1} + \Xi_{n+1} = X_n + \Xi_n + M_{0,n} + M_{1,n} - (X_n - Z_n)$. Thus,

$$\Xi_{n+1} = \Xi_n + M_{0,n} + M_{1,n} + Z_n - X_{n+1} \quad (21)$$

It can be proved that the discrete-time stochastic process $\{(X_n, \Xi_n)\}_{n=1}^{\infty}$ is a discrete-time Markov chain with the finite state space $\Omega = \{(x, \xi)|x \in \{0, 1, 2, .., N\}, \xi \in \{0, 1, 2, .., NB\}\}$. Thus, $|\Omega| = (N+1) \cdot (NB+1) = O(N^2 \cdot B)$. When $B \geq 2$ and N is much larger than

B, $O(N^2 B)$ is much smaller than $O(N^{B+1})$. Based on the discrete-time Markov chain $\{(X_n, \Xi_n)\}_{n=1}^{\infty}$, with minor modifications, many results in the previous sections could be reused to calculate the approximated values for the network throughput and the average packet delay for the splitting with remainder algorithm when $B \geq 2$.

VIII. Conclusion

In this paper, we have proposed the tree/stack splitting with remainder algorithm for distributed medium access control in a wireless network with multipacket reception. In order to reduce the length of a cycle and increase the network throughput, when the splitting with remainder algorithm is used, some nodes that attempt to transmit packets at the beginning of a cycle might have to postpone their packet retransmissions until the beginning of the next cycle. We have demonstrated that the splitting with remainder algorithm outperforms the probe algorithm and the erasure algorithm. For the splitting with remainder algorithm, we have analytically and accurately derived the network throughput and the average packet delay. We have shown that our analytical results are consistent with packet-based simulation results. The model and analytical results developed in this paper could be used to numerically obtain the optimal throughput of the network. Future works include designing an adaptive algorithm in which parameters can be dynamically adjusted to achieve the optimal throughput. Another direction of future research is combining transmission power control with the splitting with remainder algorithm to further improve the network performance. Furthermore, changing the probability for a node to join the left subgroup based on the knowledge of the channel matrix and the channel feedback is also a promising direction for future research. In the discussion section, we have shown the first steps to extend our analysis to two other cases. We plan to work out all the details in the near future. It is expected that when the buffer size is greater than two, the computational complexity to derive exact results is high. Therefore, another direction of future research is to design low-complexity algorithms to derive good approximations for the cases in which the buffer size is large.

Appendix

A. *Event-Driven Pseudo Codes for The Splitting with Remainder Algorithm*

```
onContentionCycleBegin() {
  element=createStackElement();
  if (thisNodeHasWaitingPackets()==true) {
    push(element); UpdateStackPointer(TOP); // points to the unique element in the stack
  }
  else {
    push(element); UpdateStackPointer(NULL); // points to NULL
  }
  createEvent(aSlotBegins);
}

onSlotBegin() {
  if (stackPointerPointsToTopElement()==true) transmitPacket();
}

onReceiveChannelFeedback(int feedback) {
  if (feedback==idle) {
    // after poping, the pointer always points to the top element in the stack
    popElementAndUpdateStackPointer();
  }
  else if (feedback==partial success) {
    popElementAndUpdateStackPointer();
  }
  else {
    topElement=popElementAndUpdateStackPointer();
    splitTopElement(topElement,leftElement, rightElement);
    push(rightElement);
    push(leftElement);
    if (fairCoinTossing()==HEAD) { // HEAD appears with probability 0.5
      UpdateStackPointer(TOP); // point to the leftElement
    }
    else {
      UpdateStackPointer(TOP2); // point to the rightElement
    }
  }
  if (stackIsEmpty()==true)
    createEvent(aContentionCycleBegins);
}
```

B. *Definitions of Variables*

For the convenience of readers, we summarize the definitions of key variables used in the above analysis as follows.

N: total number of nodes in the network excluding the access point

Λ: a $N \times N$ matrix such that $[\Lambda]_{i,j}$ is the probability that the access point will successfully receive/decode j packets in a time slot given that i packets are simultaneously transmitted in the time slots

z_i: the probability that the access point does not successfully receive/decode any packets in a time slot, when i nodes simultaneously transmit in the time slot

$f(x)$: the probability that x packets arrive at a node in a time slot

λ: the aggregated packet arrival rate in the network

B: the buffer size at a node (a node has at most $(B+1)$ waiting packets)

W_n^j: a binary random variable such that $W_n^j = 1$ if and only if the server at node j is occupied/busy at time n

Q_n^j: a random variable that represents the total number of packets in the queue of node j at time n

T_n: the time instance when the n-th contention cycle begins

X_n: the total number of nodes with waiting packets at time T_n, which is also the order of the n-th contention cycle

Y_n: the length of the n-th contention cycle

R_m: the total number of packets that are successfully received by the access point at time m

Z_n: the remainder of the n-th contention cycle ($Z_n = X_n - \sum_{m=T_n+1}^{T_{n+1}} R_m$)

$X_n^{(k)}$: the total number of nodes with system size k at time T_n (the total number of nodes, each has k waiting packets at time T_n)

$\Phi_1(m, k, r) = P\{Y_n = k, Z_n = r | X_n = m\}$

$F_n^{(1)}$: a random variable that represents the total number of nodes, each has one waiting packet at time T_n but fails to successfully deliver packets to the access point within the n-th contention cycle

$F_n^{(2)}$: a random variable that represents the total number of nodes, each has two waiting packets at time T_n but fails to successfully deliver packets to the access point within the n-th contention cycle

$\phi(m_1, m_2, r, r_1) = P\{F_n^{(1)} = r_1 | (X_n^{(1)}, X_n^{(2)}) = (m_1, m_2), Z_n = r\}$

$\Phi_2(m_1, m_2, k, r, n_1, n_2) = P\{(X_{n+1}^{(1)}, X_{n+1}^{(2)}) = (n_1, n_2) | Y_n = k, Z_n = r, (X_n^{(1)}, X_n^{(2)}) = (m_1, m_2)\}$

$\Theta_{a,b,n}$: the total number of nodes, each has a waiting packets at the beginning of the n-th contention cycle but has b waiting packets at the beginning of the $(n+1)$-th contention cycle.

$q = 1 - f(0)$

$\eta(k) = (f(0))^k$

$v(m_1, m_2) = \lim_{n\to\infty} P\{(X_n^{(1)}, X_n^{(2)}) = (m_1, m_2)\}$

$l(k)$: the expected value of the length of the n-th contention cycle, given that k nodes concurrently transmit at the beginning of the n-th contention cycle

A_n: the n-th time instance when the n-th contention cycle of order zero begins (the time instance when the n-th regenerative cycle begins)

$E[L] = E[A_{n+1} - A_n]$

λ_D: the network throughput

$u(m)$: the expected value of the total number of packets that are successfully received by the access point within the n-th contention cycle, given that $X_n^{(1)} + X_n^{(2)} = m$

$W_n = \sum_{j=1}^{N} W_n^j$

$E[W]$: the long-term average number of busy servers in the network per time slot

$s_1(n)$: the average number of busy servers during a contention cycle of order n

$\xi(k, r) = E[Z_n = r | X_n = k]$

$E[Q]$: the expected value of the total number of waiting packets in the N queues of the network in a time slot

$s_2(n)$: the average queue size at a node during a contention cycle of order n, when the queue size of the node at the beginning of the contention cycle is zero.

$g(m, k) = P\{Y_n = k | X_n = m\}$

$E[D]$: average system delay for a packet (the expected value of the total amount of time a packet spends in the system until it is successfully received by the access point)

References

[1] D. Bertsekas and R. Gallager, *Data Networks*, 2nd edition. Prentice Hall, 1992.

[2] J. J. Capetanakis, "Tree algorithm for packet broadcast channel," *IEEE Trans. Inf. Theory*, vol. 25, pp. 505-515, Sep. 1979.

[3] R. G. Gallager, "Conflict resolution in random access broadcast networks," in *Proc. AFOSR Workshop Communication Theory Applications*, Sep. 1978, pp. 74-76.

[4] R. R. Choudhury, Xue Yang, R. Ramanathan, and N. H. Vaidya, "On designing MAC protocols for wireless networks using directional antennas," *IEEE Trans. Mobile Comput.*, vol. 5, no. 5, pp. 477-491, May 2006.

[5] S. Ghez, S. Verdu, and S. C. Schwartz, "Stability properties of slotted ALOHA with multipacket reception capability," *IEEE Trans. Automatic Control*, vol. 33, pp. 640-649, July 1988.

[6] M. Sidi and I. Cidon, "Splitting protocols in presence of capture," *IEEE Trans. Inf. Theory*, vol. 31, pp. 295-301, Mar. 1985.

[7] B. Van Houdt and C. Blondia, "Analysis of an identifier splitting algorithm combined with polling (ISAP) for contention resolution in a wireless access network," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 11, pp. 2345-2355, Nov. 2000.

[8] I. Chlamtac and A. Farago, "An optimal channel access protocol with multiple reception capability," *IEEE Trans. Computers*, vol. 43, no. 4, pp. 480-484, Apr. 1994.

[9] R. Garces and J. J. Garcia-Luna-Aceves, "Collision avoidance and resolution multiple access for multichannel wireless networks," in *Proc. IEEE INFOCOM 2000*, pp. 595-602.

[10] F. Shad, T. D. Todd, V. Kezys, and J. Litva, "Dynamic slot allocation (DSA) in indoor SDMA/TDMA using a smart antenna base station," *IEEE/ACM Trans. Networking*, vol. 9, no. 1, pp. 69-81, Feb. 2001.

[11] Q. Zhao and L. Tong, "A multiqueue service room MAC protocol for wireless networks with multipacket reception," *IEEE/ACM Trans. Networking*, vol. 11, no. 1, pp. 125-137, Feb. 2003.

[12] X. Qin and R. Berry, "Opportunistic splitting algorithms for wireless networks," in *Proc. IEEE INFOCOM 2004*, vol. 3, pp. 1662-1672.

[13] Q. Zhao and L. Tong, "A dynamic queue protocol for multiaccess wireless networks with multipacket reception," *IEEE Trans. Wireless Commun.*, vol. 3, no. 6, pp. 2221-2231, Nov. 2004.

[14] S. Adireddy and L. Tong, "Exploiting decentralized channel state information for random access," *IEEE Trans. Inf. Theory*, vol. 51, no. 2, pp. 537-561, Feb. 2005.

[15] V. Naware, G. Mergen, and L. Tong, "Stability and delay of finite-user slotted ALOHA with multipacket reception," *IEEE Trans. Inf. Theory*, vol. 51, no. 7, pp. 2636-2656, July 2005.

[16] J. Luo and A. Ephremides, "Power levels and packet lengths in random multiple access with multiple-packet reception capability," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 414-420, Feb. 2006.

[17] R.-H. Gau, "Performance analysis of slotted Aloha in interference-dominating wireless ad-hoc networks," *IEEE Commun. Lett.*, vol. 10, no. 5, pp. 402-404, May 2006.

[18] R.-H. Gau and K.-M. Chen, "Predictive multicast polling for wireless networks with multipacket reception and queueing," *IEEE Trans. Mobile Comput.*, vol. 5, no. 6, pp. 725-737, June 2006.

[19] Y. Yu and G. B. Giannakis, "High-throughput random access using successive interference cancellation in a tree algorithm," *IEEE Trans. Inf. Theory,* vol. 53, no. 12, pp. 4628-4639, Dec. 2007.

[20] M. Lotfinezhad, B. Liang, and E. S. Sousa, "Adaptive cluster-based data collection in sensor networks with direct sink access," *IEEE Trans. Mobile Comput.*, vol. 7, no. 7, pp. 884-897, July 2008.

[21] R.-H. Gau, "Performance analysis of finite-user slotted Aloha in wireless networks with multipacket reception and random traffic," *IEEE Commun. Lett.*, vol. 12, no. 2, pp. 140-142, Feb. 2008.

[22] A. Dua, "Random access with multi-packet reception," *IEEE Trans. Wireless Commun.*, vol. 7, no. 6, pp. 2280-2288, June 2008.

[23] X. Wang, Y. Yu, and G. B. Giannakis, "Design and analysis of cross-layer tree algorithms for wireless random access," *IEEE Trans. Wireless Commun.*, vol. 7, no. 3, pp. 909-919, Mar. 2008.

[24] R.-H. Gau and K.-M. Chen, "Probability models for the splitting algorithm in wireless access networks with multi-packet reception and finite nodes," *IEEE Trans. Mobile Comput.*, vol. 7, no. 12, pp. 1519-1535, Dec. 2008.

[25] M.-F. Guo, X. Wang, and M.-Y. Wu, "On the capacity of k-MPR wireless networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 7, pp. 3878-3886, July 2009.

[26] R. Yim, N. B. Mehta, A. F. Molisch, and J. Zhang, "Dual power multiple access with multipacket reception using local CSI," *IEEE Trans. Wireless Commun.*, vol. 8, no. 8, pp. 4078-4088, Aug. 2009.

[27] R.-H. Gau, "Modeling the slotted nonpersistent CSMA protocol for wireless access networks with multiple packet reception," *IEEE Commun. Lett.*, vol. 13, no. 10, pp. 797-799, Oct. 2009.

[28] G. D. Celik, G. Zussman, W. F. Khan, and E. Modiano, "MAC for networks with multipacket reception capability and spatially distributed nodes," *IEEE Trans. Mobile Comput.*, vol. 9, no. 2, pp. 226-240, Feb. 2010.

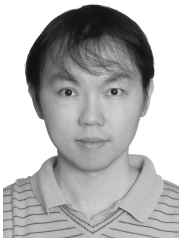
[29] A. D. Gore and A. Karandikar, “Power-controlled FCFS splitting algorithm for wireless networks,” *IEEE Trans. Veh. Technol.*, vol. 59, no. 2, pp. 842-856, Feb. 2010.

[30] S. M. Ross, *Stochastic Processes*, 2nd edition. John Wiley and Sons Inc., 1996.

[31] S. Verdu, *Multiuser Detection*. Cambridge University Press, 1998.

[32] D. Tse and P. Viswanath, *Fundamentals of Wireless Communications*. Cambridge University Press, 2005.

[33] E. Horowitz, S. Sahni, and D. P. Mehta, *Fundamentals of Data Structures in C++*, 2nd edition. Silicon Press, 2007.



Rung-Hung Gau received the BS degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1994 and the MS degree in electrical engineering from University of California at Los Angeles, Los Angeles, California, USA, in 1997. He received the PhD degree in electrical and computer engineering from Cornell University, Ithaca, New York, USA, in 2001. He is currently an associate professor in the Department of Electrical Engineering, National Chiao Tung University, Hsinchu, Taiwan. His research interests include cross-layer medium access control in wireless networks, network coding, mobility management, multicast flow control, and wireless ad-hoc sensor networks. His URL is http://cmbsd.cm.nctu.edu.tw/∼runghung/.