Queue Stability Analysis in Network Coded Wireless Multicast Network

Nadieh Moghadam and Hongxiang Li

***Abstract*—This letter considers a single hop wireless multicast network. We first introduce a new two-level queuing system consisting of a main queue and a virtual queue, where each packet in the virtual queue is associated with a user index set. Then, we propose a network coding based packet scheduling method to maximize the system input rate under the queue stability constraint. Our analytical and simulation results demonstrate the effectiveness of the proposed solution.**

***Index Terms*—Multicast, network coding, stability, virtual queue.**

I. Introduction

Wireless multicast transmission achieves high spectrum and power efficiency because a single transmission will accommodate multiple receivers simultaneously. However, reliable multicast transmissions over packet erasure channels is quite challenging because of the varied channel conditions and the heterogeneity in the amount of information correctly received across all receivers. Fortunately, network coding (NC) offers a promising platform for multicast transmissions [1]. Particularly, it has been proven that one can approach the multicast capacity by using random linear network coding (RNC) techniques [2], where the transmitter forms output data by linearly combining the input data specified by independently and randomly chosen code coefficients from some finite field. A receiver is able to decode the original transmitted data when it receives a full set of independently encoded packets.

In the literature, most existing studies on RNC in multicast networks assume saturated transmitter queue that guarantees packet availability for transmission [3]–[5], where the queue length is considered to be infinite so that there is no buffer overflow. In this letter, we relax the unrealistic assumption of saturated queue and concern the queue stability of RNC based wireless multicast network. Specifically, we assume that packets randomly arrive at the transmitter and they are queued while waiting for transmission. A stability policy is provided in [6] for a network coded unicast network. In delay sensitive applications, [7] proposed a buffer-aware network coding method to reduce the delay caused by random packet arrivals. Additionally, stability properties are also evaluated in [8] for broadcast/multicast erasure channels with NC. In [8], the authors proposed a suboptimal virtual queue structure and provide detailed analysis for the case of two receivers. However, there is no closed form analysis for stability conditions and the proposed queueing model is quite complicated when the number of receivers is greater than two. To the best of our knowledge, queue stability in network coded multicast network has not been thoroughly studied. In [9], the authors investigated a similar problem but with a totally different queueing model. Specifically, they introduced a virtual queue for every receiver, while we only have one virtual queue for the entire system. Furthermore, their algorithm is limited to the special case of only two receivers. In this letter, we study new queuing structure and scheduling method to maximize the multicast network throughput under stability constraint. Specifically, our main contributions are:

- we propose a new network coded queueing model;
- we derive a scheduling algorithm that maximizes the packet arrival rate (input rate) while guaranteeing the stability of the transmitter queue;
- we extend the analytical results in [8] to arbitrary number of receivers and compare their performance with our proposed algorithm.

The rest of the letter is organized as follows. We first describe the system model in Section II. The performance analysis is provided in section III, followed by simulation validation in section IV. Finally, a conclusion is drawn in Section V.

II. System Model

We consider a single-hop wireless multicast network where the source (transmitter) multicasts data packets to N destinations (receivers) over erasure channels. Without loss of generality, we assume packets are independently generated according to a stationary process with arrival rate λ. Each packet transmission fails at receivers independently with packet error rate ϵ_i $(i = 1, 2, \ldots, N)$. The system is time-slotted and each packet transmission takes one time slot. We further assume that, at the beginning of each time slot, the transmitter reliably receives one-bit feedback message from each receiver indicating whether the previously transmitted packet has been received successfully.

A queue is considered stable if the arrival rate is less than the service rate. For queue stability analysis, we consider stationary operation when the queue distribution reaches a steady state. Let μ be the service rate of the source queue, the stability condition is given by $\lambda/\mu < 1$. In this letter, we propose a new two-level queue structure at the transmitter: the main queue Q_0 stores the newly arrived packets; the virtual queue Q_v stores those transmitted packets that are successfully received by at least one but not all receivers. For each packet $P_i \in Q_v$, an *index set* I_i is introduced which consists of the receivers who have not successfully received packet P_i. For simplicity,

Manuscript received October 18, 2015; accepted January 21, 2016. Date of publication February 3, 2016; date of current version May 6, 2016. This work was supported by the National Science Foundation under Grant ECCS-1509047. The associate editor coordinating the review of this paper and approving it for publication was N. Pappas. *(Corresponding author: Hongxiang Li.)*

The authors are with the Department of Electrical and Computer Engineering, University of Louisville, Louisville, KY 40292 USA (e-mail: n0moha07@louisville.edu; h.li@louisville.edu).

Digital Object Identifier 10.1109/LCOMM.2016.2524453

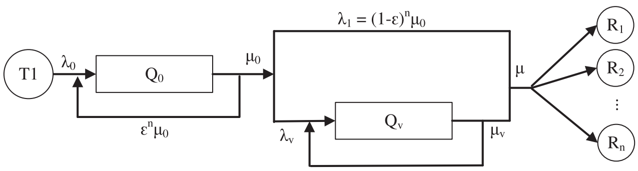

Fig. 1. Architecture of the queuing model.

Fig. 1 shows the architecture of the queuing model with common channel loss probabilities of ϵ. It should be noted that a transmitted packet from Q_0 remains in Q_0 if it is received by none of the receivers, which is shown in Fig. 1 as feedback for Q_0 with probability $\epsilon^n \mu_0$. In Fig. 1, μ_0 and μ_v are the service rates and λ_0 and λ_v are the input rates for Q_0 and Q_v respectively; and λ_1 is the rate by which a packet is received by all receivers at the same time. From Fig. 1 we can easily obtain:

$$\lambda_v = \mu_0 - \lambda_1 = (1 - (1 - \epsilon)^n)\mu_0 \quad (1)$$

where,

$$\mu_0 = \frac{\lambda_0}{1 - \epsilon^n} \quad (2)$$

Since the packets in the virtual queue still occupy memory of the transmitter, the total queue size, $|Q_T|$, is given by $|Q_T| = |Q_0| + |Q_v|$.

Based on the queue structure in Fig. 1, we have the following observations: (1) Sending a main queue packet (m-packet) maximizes the effective throughput because the m-packet is desired by all receivers; (2) While a virtual queue packet (v-packet) is only useful for a subset of receivers, its index set provides necessary and sufficient side information to employ network coding to reduce the overall queue length. Apparently, there is a tradeoff between queue size and throughput. To maximize the input rate and guarantee queuing stability, we propose a new Virtual Queue based Multicast Network Coding and Scheduling scheme (VQ-MNCS) with the following transmission priority policy:

1) **First priority**: Whenever there exist more than k packets in Q_v, the transmitter selects the maximum number of packets in Q_v whose index sets are mutually exclusive (i.e., no intersection between any two index sets). If there are more than one selection options, the set of packets with the largest aggregated receiver index set will be selected. Then, the selected packets are combined via RNC (i.e., XOR operation) and the network coded packet is sent out in the current time slot. Then, according to receiver feedback, each selected packet either updates its index set or in particular, if the packet is successfully decoded by all the intended receivers, its index set becomes empty and the packet leaves the queue system immediately.
2) **Second priority**: If the first priority is not applicable and Q_0 is nonempty, the head-of-line packet from Q_0 is sent to all receivers.
3) **Third priority**: If both the first and second priorities are not met, the packets selection and RNC combining process described in the first priority is performed, even though the number of packets in Q_v is less than k.

Algorithm 1 (VQ-MNCS Algorithm)

1: Set the value of threshold k.
2: **while** Q_T is nonempty **do**
3: **if** Q_0 is nonempty and $|Q_v| \leq k$ **then**
4: Transmits head-of-line packet $p_h \in Q_0$.
5: **if** p_h is received by all receivers **then**
6: p_h leaves the queuing system
7: **else if** p_h is not received by any receiver **then**
8: p_h stays put in Q_0.
9: **else**
10: p_h becomes a v-packet and enters Q_v with an associated index set.
11: **end if**
12: **else**
13: Find packet subset $P \subset Q_v$, whose index sets are mutually exclusive, such that its cardinality $|P|$ is maximized.
14: XOR-combine all packets in P and transmit the combined packet.
15: Based on receiver feedback, update the index sets of the involved v-packets.
16: **if** an updated index set becomes empty **then**
17: the corresponding packet leaves the queue.
18: **end if**
19 **end if**
20: **end while**

Based on the above policy, the packet encoding and scheduling method is summarized in Algorithm 1.

Remark 1: Threshold k is a key system parameter. Generally, the combining of v-packets (i.e., network coding) becomes more beneficial when k is large. On the other hand, larger k increases the queue length (i.e., more memory) and the complexity to find the optimal packet subset P.

Remark 2: When the number of receivers n is large, finding the optimal subset P (line 13 of Algorithm 1) is nontrivial. In this case, to simplify the complexity, we can limit the size of P to j (i.e., $|P| = j$, denoted as $j - Best$) where the j index sets are mutually exclusive and their union yields the longest index set.

III. Performance Analysis

To better understand Algorithm 1, in this section we provide performance analysis to determine the maximum input rate under queuing stability constraint. For simplicity, we consider the case of three receivers. However, the results can be extended to an arbitrary number of users. In this letter, we focus on delay-sensitive multicast transmissions where each virtual queue packet will leave the queue after it is network coded and re-transmitted (regardless if it is received or not by all receivers). Under the queue stability condition, we assume there are L packets in the virtual queue, i.e., $Q_v = \{P_1, P_2, \ldots, P_L\}$ with associated index sets $S = \{I_1, I_2, \ldots, I_L\}$. Now we want to find out the probability of the existence of a packet with r specific indices, referred as $r - element$ index set in S in a n receiver scenario. First we define events X and Y as follows:

- **Event X:** a transmitted m-packet is not received by r specific users.

- **Event Y:** a transmitted m-packet is received by at least one but not all of the receivers (i.e., a m-packet becomes a v-packet).

Then, the probability that a v-packet's index set consists of r specific users is given by:

$$p_r = p_{(X|Y)} = \frac{p_{(X\cap Y)}}{p_{(Y)}} = \frac{\epsilon^r(1-\epsilon)^{n-r}}{1-\epsilon^n-(1-\epsilon)^n} \tag{3}$$

In the next step, our stability analysis focuses on a scenario with 3 receivers. In the 3 receiver scenario we only have three options for XOR-ing packets of Q_v. Accordingly, we have the following priority policy:

1) **First priority**: It is clear that XOR-ing three different packets with $1-element$ index set yield the best performance. Probability of the existence of these three packets in the Q_v is:

$$P_{1+1+1} = \binom{n}{3}\sum_{k_1=1}^{L-2}\sum_{k_2=1}^{L-k_1-1}\sum_{k_3=1}^{L-k_1-k_2} \binom{L}{k_1,k_2,k_3} p_1^{k_1+k_2+k_3}(1-3p_1)^{L-(k_1+k_2+k_3)} \tag{4}$$

2) **Second priority**: While the first priority is not applicable, XOR-ing one packet with $1-element$ index set and one packet with $2-element$ index set has the second highest performance and can be obtained from:

$$P_{1+2} = \binom{n}{1,2}\Bigg[\sum_{k_1=1}^{L-2}\sum_{k_2=1}^{L-k_1-1}\sum_{k_3=1}^{L-k_1-k_2} 2\binom{L}{k_1,k_2,k_3} p_1^{k_1+k_3}p_2^{k_2}(1-3p_1-p_2)^{L-(k_1+k_2+k_3)} + \sum_{k_1=1}^{L-1}\sum_{k_2=1}^{L-k_1}\binom{L}{k_1,k_2}p_1^{k_1}p_2^{k_2}(1-3p_1-p_2)^{L-(k_1+k_2)}\Bigg] \tag{5}$$

where p_1 and p_2 are the probabilities of the existence of a $1-element$ and $2-element$ index set in S and they can be obtained from (3).

3) **Third priority**: While both first and second priorities are not applicable our option would be XOR-ing two different packets with $1-element$ index sets:

$$P_{1+1} = \binom{n}{2}\sum_{k_1=1}^{L-1}\sum_{k_2=1}^{L-k_1}\binom{L}{k_1,k_2} p_1^{k_1}p_1^{k_2}(1-3p_1)^{L-(k_1+k_2)} \tag{6}$$

Now we can conclude that the total probability that packets leave Q_v is:

$$\mu_v = \frac{\lambda_v}{3}P_{1+1+1} + \frac{\lambda_v}{2}P_{1+1} + \frac{\lambda_v}{2}P_{1+2} + \lambda_v(1-P_{1+1+1}-P_{1+1}-P_{1+2})(1-\mu_0) \tag{7}$$

where the numbers in the denominators, 3, 2 and 2, are the numbers of combined packets with RNC. Where λ_v and μ_0 can be obtained from (1) and (2).

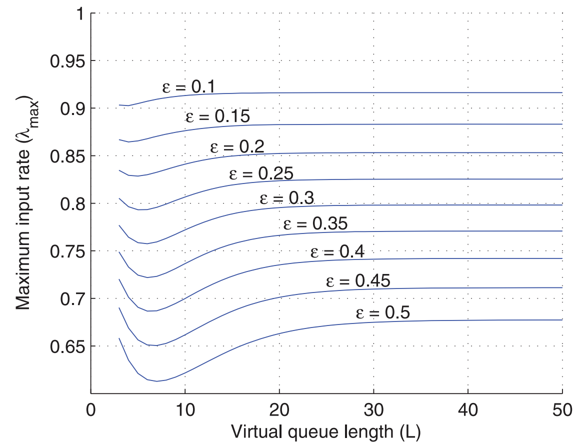

Fig. 2. Maximum input rate versus virtual queue length.

TABLE I
λ_{max} VERSUS ϵ RESULTED FROM FIG. 2

ϵ	10%	15%	20%	25%	30%	35%	40%	45%	50%
λ_{max}	0.916	0.883	0.853	0.825	0.798	0.771	0.742	0.711	0.677

It is obvious that the exiting packets from each queue enter the channel which is assumed to have the maximum capacity of one packet per time slot. Thus:

$$\mu_0 + \mu_v \le 1 \tag{8}$$

Plugging (1), (2) and (7) into (8), we can obtain a second order equation of λ_0. For any given ϵ the maximum input rate, λ_{max}, can be obtained from:

$$\lambda_{max} = \lim_{L\to\infty}\{\lambda_0 | 0 \le \lambda_0 \le 1\} \tag{9}$$

Fig. 2 shows the maximum input rate (λ_{max}) versus the virtual queue length (L) for different ϵ. We can see that, for given ϵ, λ_{max} asymptotically converges to its maximum value as L increases. In other words, when L is greater than some threshold, the maximum input rate is not dependent on the virtual queue length anymore. Table I shows λ_{max} for different ϵ. To analytically characterize the relationship between λ_{max} and ϵ, we use MATLAB to fit the data in Table I into a order-3 polynomial function, as shown in Equation (10).

$$\lambda_{max} = -1.39\epsilon^3 + 1.21\epsilon^2 - 0.9\epsilon + 1 \tag{10}$$

As we mentioned before, these results are only valid for the scenario with 3 receivers; however, for different number of receivers, the only difference is the number of probabilities explained before as P_{1+1+1}, P_{1+1} and P_{1+2}.

For an arbitrary number of receivers, the number of probabilities we have to calculate can be derived from the mathematical method called *"Partitions of Integers"*.

Definition: If a finite sequence $(a_1, a_2, \ldots, a_k)$ of positive integers satisfies $a_1 \ge a_2 \ldots \ge a_k$ and $a_1 + a_2 + \ldots + a_k = n$, we call that sequence a partition of the integer n or simply $p(n)$ [10].

Using this definition, for a generic multicast network with n receivers, the number of probabilities we need to calculate is $p(n) - 1$.

TABLE II
THE VALUES OF $p(n)$ FOR $n \leq 8$

n	2	3	4	5	6	7	8
$p(n)$	2	3	5	7	11	15	22

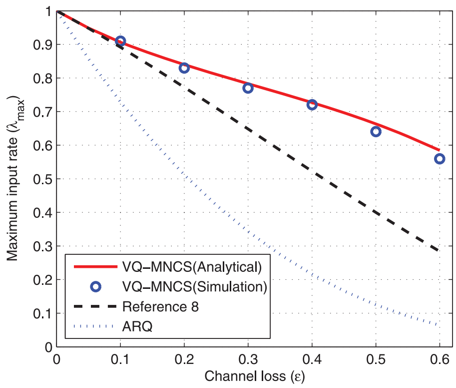

Fig. 3. Maximum input rate versus channel loss with 3 receivers.

It should be noted that beside $p(n) - 1$ number of possible cases, we also can add the partitions of integers less than n. Thus, the final number of possible cases, N_p is:

$$N_p = \sum_{k=2}^{n} \left[p(k) - 1 \right] \tag{11}$$

While an exact formula to calculate $p(n)$ indeed exists, it is by no means simple [11]. When n is small, Table II shows the values of $p(n)$ [10].

In order to calculate the complexity of VQ-MNCS Algorithm, we have to exhaustively search and compare all packets in the queue of length L to find packets that have mutually exclusive index sets. In the worst case, the total number of comparisons is:

$$C_T = \sum_{i=2}^{L} \binom{L}{i} \frac{i(i-1)}{2} \tag{12}$$

We can see that the complexity of the VQ-MNCS algorithm is proportional to $2^L L^2$. Furthermore, note that the complexity of comparing two index sets depends on the length of the packets indices. In a n user scenario, the maximum length of an index set is $n - 1$. Therefore, in the worst case, the algorithm complexity is $O((n-1)2^L L^2) \simeq O(n2^L L^2)$.

To reduce the computational complexity, we introduce a suboptimal $j - Best$ algorithm in which we combine no more than j packets with mutually exclusive index sets. Compared to the exponential complexity in (12), the complexity of $j - best$ algorithm is reduced to polynomial $O(nL^4)$.

IV. SIMULATION RESULTS

In this section, we validate the effectiveness of the proposed VQ-MNCS algorithm by simulations and compare its performance with existing solutions.

For the case of three receivers described in section III, Fig. 3 illustrates the maximum input rate versus ϵ when the queuing system is stable. We can see that the simulation results match our analysis very well. As expected, the input rate decreases with ϵ. More importantly, our VQ-MNCS solution outperforms the traditional ARQ and the one proposed in [8], and the performance gain increases with ϵ. Similarly, Fig. 4 compares our VQ-MNCS based algorithms with [8] and ARQ when the number of receivers is 6. In particular, we apply the $j - Best$ methods ($j = 2, 3, 4$). As expected, the maximum input rate increases with the value of j. With the relaxation described in section III, we can see that the green curve upper bounds all other transmission schemes.

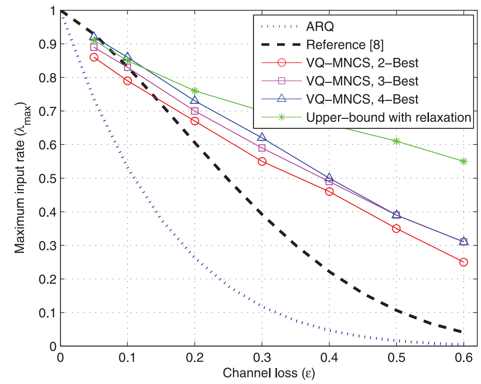

Fig. 4. Maximum input rate versus channel loss with 6 receivers.

V. CONCLUSION

In this letter we proposed a new queueing model for multicast network, where one virtual queue consists of packets that are received by only partial users. Based on this new queuing model, we developed a network coding based packet scheduling algorithm. The results show the effectiveness of our proposed solution.

REFERENCES

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, “Network information flow,” *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[2] T. Ho et al., “A random linear network coding approach to multicast,” *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

[3] P. Chau, S. Kim, Y. Lee, and J. Shin, “Hierarchical random linear network coding for multicast scalable video streaming,” in *Proc. Annu. Summit Conf. Asia-Pacif. Signal Inf. Process. Assoc. (APSIPA)*, 2014, pp. 1–7.

[4] J. Qureshi, C. H. Foh, and J. Cai, “An efficient network coding based retransmission algorithm for wireless multicast,” in *Proc. IEEE 20th Int. Symp. Pers. Indoor Mobile Radio Commun.*, Sep. 2009, pp. 691–695.

[5] M. Amerimehr, F. Ashtiani, and S. Valaee, “Maximum stable throughput of network coded multiple broadcast sessions for wireless tandem random access networks,” *IEEE Trans. Mobile Comput.*, vol. 13, no. 6, pp. 1256–1267, Jun. 2014.

[6] S. Athanasiadou, M. Gatzianas, L. Georgiadis, and L. Tassiulas, “Stable and capacity achieving XOR-based policies for the broadcast erasure channel with feedback,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2013, pp. 2905–2909.

[7] W. Chen, K. B. Letaief, and Z. Cao, “Buffer-aware network coding for wireless networks,” *IEEE/ACM Trans. Netw.*, vol. 20, no. 5, pp. 1389–1401, Oct. 2012.

[8] Y. E. Sagduyu and A. Ephremides, “On broadcast stability of queue-based dynamic network coding over erasure channels,” *IEEE Trans. Inf. Theory*, vol. 55, no. 12, pp. 5463–5478, Dec. 2009.

[9] N. Moghadam and H. Li, “Improving queue stability in wireless multicast with network coding,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2015, pp. 3382–3387.

[10] M. Bona, *Introduction to Enumerative Combinatorics*. New York, NY, USA: McGraw-Hill, 2007.

[11] G. E. Andrews, *The Theory of Partitions*. Cambridge, U.K.: Cambridge Univ. Press, 1998.