Approximate Analysis of a Round Robin Scheduling Scheme for Network Coding*

Omer H. Abdelrahman and Erol Gelenbe

Department of Electrical and Electronic Engineering,
Imperial College London, UK
{oha06,e.gelenbe}@imperial.ac.uk

Abstract. Network coding (NC) has been proposed as a way to compress flows of packets by combining different packets, provided that there is sufficient redundancy through multiple transmission paths so that the receiving nodes can then decode the flows to reconstruct all of the individual packets. However NC does introduce additional computational overhead, and also creates different additional delays which are the subject of this paper. In particular, we evaluate the queueing and delay performance of NC at intermediate nodes of a store and forward packet network when the cross-encoding of packets is restricted within disjoint subsets of the traffic streams so that packets from different subsets cannot be encoded together. We propose a round robin scheduling scheme for serving these disjoint subsets, and present a queueing model for a single encoding node that captures the effect of NC. The model is analyzed approximately using a decoupling approach, and can be used to predict the additional delays incurred by packets in nodes that use NC. The accuracy of the analytical solution is validated via simulations.

Keywords: Network coding, performance evaluation, vacation models.

1 Introduction

Network coding (NC) [1] allows the cross encoding of packets at intermediate nodes before forwarding them towards their destination, and can offer greater communications efficiency and better usage of overall network bandwidth, at the cost of more processing and additional delays.

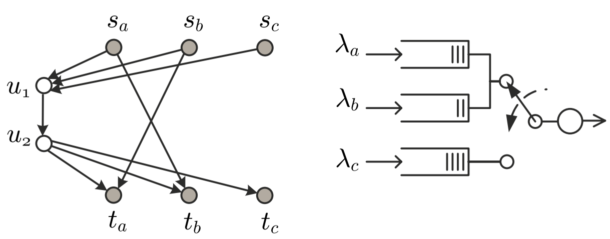
In [2], we analyzed the delay performance of NC at intermediate nodes of a store and forward network when the nodes encode together all traffic streams that pass through them. Our analysis showed that NC can improve overall performance significantly provided that it is used *opportunistically.* In this paper, we extend the previous analysis to a more general setting where encoding nodes perform NC within disjoint subsets of the packet streams so that packets from different subsets are not allowed to be mixed. This constraint arises when some source-destination pairs cannot use a sufficient number of redundant paths

* Research supported by the EU FP6 CASCADAS and FP7 DIESIS Projects of the EU Future and Emerging Technologies Program.

J.T. Bradley (Ed.): EPEW 2009, LNCS 5652, pp. 212–217, 2009.

for decoding purposes. To see this, consider the directed network depicted in Fig. 1(a), which has three independent unicast information flows a, b and c between the source destination pairs (s_i, t_i). Without NC, the three flows must share the link $u_1 \rightarrow u_2$ which becomes a bottleneck, while some of the links in the network will not be utilized. However, combining the three flows together at node u_1 will leave non of the receivers able to recover its desired information. We propose a scheduling scheme which can alleviate this problem. In particular, node u_1 can alternate between transmitting the coded flow $a \oplus b$ and the uncoded flow c along the bottleneck link to node u_2, which then forwards the coded flow to both t_a and t_b and the uncoded flow to t_c. Receivers t_a and t_b can then reconstruct the original flows from $\{b, a \oplus b\}$ and $\{a, a \oplus b\}$, respectively. Thus, combining NC and scheduling in this scenario can reduce traffic rate on the bottleneck link while distributing traffic on a larger number of paths. The equivalent queueing model representation of the encoding node u_1 with the proposed coding scheme is presented in Fig. 1(b).

The rest of this paper is organized as follows. In section 2, the queueing model for a single encoding node is presented along with the approximation technique. In section 3, the accuracy of the proposed analysis is validated by comparison with simulation results, and the proposed scheme's performance is compared to the performance of a peer non-coding technique. Conclusions are summarized in section 4.



(a) (b)

Fig. 1. (a) A directed network with three unicast sessions (s_i, t_i). (b) equivalent queueing model representation of the encoding node u_1 with the proposed coding scheme.

2 System Model

Consider a node receiving F distinct independent flows of packets, each assumed to be Poisson of arrival rate λ_i for the ith flow, which queue up in distinct buffers of infinite capacity. Assume that packet lengths in each stream are independent random variables, and that they are mutually independent between flows, with general distribution $S(x) = Prob[S \leq x]$, where S is the random variable representing packet length. Assume also that the node transmission time is directly proportional to packet length with a constant of proportionality of 1. When the server encodes n packets of different lengths, it packs the shorter packets with *zero*-bits to reach the length of the longest packet, and encodes the resulting packet bit by bit so that its length will be equal to the largest of the n packet lengths. The transmission time of a packet is then assumed to be proportional to the length of the largest packet.

We divide the input flows $\{1, \ldots, F\}$ into J disjoint subsets which we denote as *coding classes*, and let C_j be the flows in class-j. We set the restriction that only those packets within the same class will be encoded together while packets of different classes will not be mixed. We assume that the server cycles the coding classes in a round robin manner and that it switches from one class to another instantaneously, i.e. we assume zero *switchover* time. When the server visits a coding class, it removes the head-of-the-line packets from non-empty queues in the class, forms a single coded packet and transmits the resulting packet on the output link. When the server arrives at a coding class and finds that all its queues are empty, it immediately switches to the next class.

We will analyze the performance of the system approximately by assuming that the complex model can be decomposed into F separate queues where each queue is treated as a *server of the walking type* [3,4]. In this single-server model, each time the queue is non-empty the server serves one customer (packet) for a service time S then becomes idle for a period T after which it examines the queue again. If it finds that the queue is empty then it takes off for a vacation time V after which it returns once again to examine the queue. Let U be the waiting time in a simple queueing system without vacations and with the same arrival process as the system considered, but with a modified service time $Y = S + T$. Then the following equality holds in distribution [3]:

$$W = U + \hat{V} \tag{1}$$

where $\hat{V}$ denotes the forward recurrence time of the vacation period V which has a distribution:

$$\hat{V}(x) = \frac{1}{E[V]} \int_0^x [1 - V(y)] \; dy \tag{2}$$

Thus the result summarized in (1) allows us to map all properties of interest of a queue with vacations in steady state to those of a system without vacations using the probability distribution of the vacation time V. Note also that (1) holds as long as the arrivals of customers to the queue constitute a renewal process, i.e. inter-arrival times are independent and identically distributed (iid) random variables [3].

We will study a queue, say the ith, in isolation from the others and consider that the queues interact with each other via the steady-state probabilities. Denote by C_{n_i} the coding class to which the ith queue belongs, and define $C_{n_i}(i) = \{x \in C_{n_i} | x \neq i\}$. If the ith queue is not empty when C_{n_i} is scheduled, then the subsequent service time S_i will be obtained from the maximum of the service times for the set of non-empty queues in the class including the ith queue. The server then moves to the next coding class and offers service in a similar manner. The vacation period T_i thus corresponds to the time interval from the server's departure from C_{n_i} until its next visit, which consists of the sum of service times at the other coding classes. Note that these service times can also be of zero duration if all other coding classes are empty. If, however, the server finds queue i empty upon scheduling C_{n_i}, then a sequence of service times V_i involving the other queues in the same class $\hat{S}_i$ and the other coding classes T_i will take place, some of these services possibly being of zero duration if all the other queues are empty. We will assume that the vacation periods V_i and T_i are iid random variables. The decoupling approximation we propose is as follows. Let q_i be the probability that in steady state the ith server does not participate in encoding a packet when the corresponding coding class is scheduled for service. Let Z_j be the set of all subsets of C_j, including C_j and the empty set. We will assume that the steady-state probability that any subset $Z \in Z_j$ is busy when visited by the server is given by $\prod_{k \in Z}[1 - q_k] \prod_{k \notin Z} q_k$. With these assumptions, the solution of the system can be summarized in the following steps:

Step 1. Assuming that the quantities q_i are known, find the distribution of the service and vacation times for each queue i as a function of q_k, $\forall k \neq i$:

$$Y_i = S_i + T_i \tag{3a}$$

$$S_i(x) = S(x) \sum_{Z \in Z_{n_i}(i)} S(x)^{|Z|} \prod_{k \in Z}[1 - q_k] \prod_{k \notin Z} q_k \tag{3b}$$

$$T_i = \sum_{j=1, j \neq n_i}^{J} T_{C_j} \tag{3c}$$

$$T_{C_j}(x) = \sum_{Z \in Z_j} S(x)^{|Z|} \prod_{k \in Z}[1 - q_k] \prod_{k \notin Z} q_k \tag{3d}$$

$$V_i = \hat{S}_i + T_i \tag{3e}$$

$$\hat{S}_i(x) = \sum_{Z \in Z_{n_i}(i)} S(x)^{|Z|} \prod_{k \in Z}[1 - q_k] \prod_{k \notin Z} q_k \tag{3f}$$

where $|Z|$ denotes the number of elements in set Z.

Step 2. For each subsystem i, determine the steady state probability q_i approximately using one of the following expressions:

- The probability of the equivalent ith server being empty when it returns from a vacation period [4]:

$$\hat{q}_i = \frac{1 - \lambda_i E[Y_i]}{1 + \lambda_i (E[V_i] - E[Y_i])} \tag{4a}$$

- The probability of the ith queue being empty or that it is busy but the equivalent server is idle due to a vacation time:

$$\hat{q}_i = \max\{0, 1 - \lambda_i E[Y_i]\} \tag{4b}$$

Step 3. Solve the system of non-linear equations $\hat{q}_i = q_i$, for $i = 1, \ldots, F$.

Now applying standard results for an $M/G/1$ queueing system and utilizing the decomposition property (1), we can write the mean waiting time in the ith queue as:

$$E[W_i] = \frac{\lambda_i E[Y_i^2]}{2(1 - \lambda_i E[Y_i])} + E[\hat{V}_i] \tag{5}$$

The mean response time is then $E[R_i] = E[W_i] + E[S_i]$.

The output rate from the encoding node (throughput) can also be obtained approximately as:

$$\gamma = \sum_{j=1}^{J} \sum_{n=1}^{|C_j|} \sum_{\substack{Z \subseteq C_j \\ |Z|=n}} \prod_{k \in Z} [1 - q_k] \prod_{k \notin Z} q_k \left(E[T_i] \mathbf{1}_{i \in C_j} + \int_0^{\infty} x \, n S(x)^{n-1} dS(x) \right)^{-1} \tag{6}$$

where $\mathbf{1}_x$ is the characteristic function which takes the value 1 if x is true and 0 otherwise.

3 Numerical Results

In this section we present numerical results which validate the accuracy of the mathematical model, and we compare the performance of the proposed scheme against a peer non-coding approach. The latter uses a technique known as store and forward in which packets received from different incoming links are stored in a single queue and served in a FIFO order. This can be modelled as the classical $M/G/1$ queueing system. For fair comparison of the two schemes, we provide the same total incoming traffic rate and packet length distribution.

In Fig. 2 we present results for the mean response time and bandwidth efficiency for the encoding node in Fig. 1(a). We assume that packet lengths are

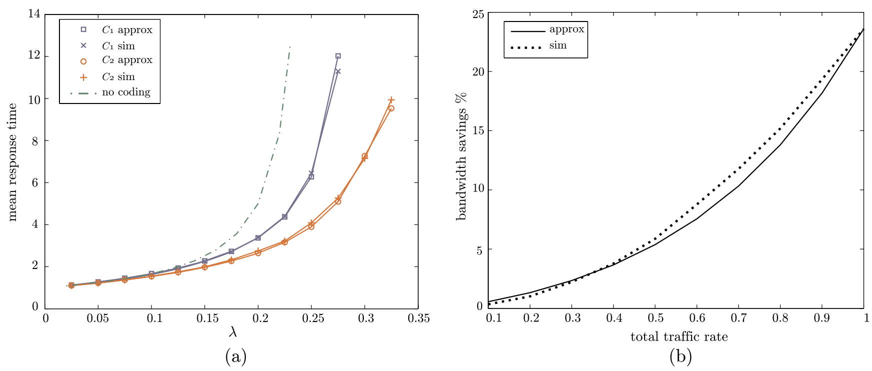


Fig. 2. (a) Mean response time and (b) bandwidth savings, for the encoding node u_1 in Fig. 1. The parameters are: $F = 3$, $J = 2$, $C_1 = \{a, b\}$, $C_2 = \{c\}$, arrival rates vector $\underline{\lambda} = \lambda[1.5 \quad 1.5 \quad 1]$ and exponential packet lengths with mean 1.

exponentially distributed and we vary the load on each queue from very light until a value that saturates the system. The figure indicates that the analytical results are in excellent agreement with those obtained from simulation. The model, however, tends to produce less accurate results when the queues are heavily loaded. This is because the queues are highly coupled in this instance, while the approximation is based on independence assumptions.

The mean response time results clearly demonstrate that the coding scheme outperforms the store and forward method, particularly when the node is heavily loaded. Moreover, since no packet loss is incurred, it follows that the scheme can deliver more packets per time unit, which translates to higher network throughput provided that destinations can reconstruct the original flows in a timely manner. Fig. 2(b) shows the percentage of saved bandwidth resulting from NC when the traffic rates do not saturate the store and forward node, i.e. $\sum_i \lambda_i E[S] < 1$. The figure indicates that coding can reduce bandwidth utilization by up to 25% when the node is heavily loaded. Note that bandwidth reductions of 37.5% could have been achieved if the flows were fully synchronized. In general, the bandwidth savings are more pronounced when the traffic rates within each coding class are balanced, since more coding opportunities arise in such instances.

4 Conclusions

In this paper, a round robin scheduling scheme based on NC has been proposed and analyzed approximately. The analysis was carried out using a decoupling approach, which was validated by simulation. The obtained results showed that the scheme outperforms the traditional store and forward technique, particularly when the system is heavily loaded. In future work, we will consider scheduling policies which aim at maximizing the throughput of the system, by employing priority levels, while maintaining acceptable QoS for the different flows. The analytical model, which seems to provide adequate delay approximation, will also be used to design efficient network algorithms such as routing and flow control under NC.

References

1. Ahlswede, R., Cai, N., Li, S.Y., Yeung, R.: Network information flow. IEEE Transactions on Information Theory 46(4), 1204–1216 (2000)
2. Abdelrahman, O.H., Gelenbe, E.: Queueing performance under network coding. In: Information Theory Workshop, 2009. ITW 2009. IEEE, Los Alamitos (2009)
3. Gelenbe, E., Iasnogorodski, R.: A queue with server of walking type (autonomous service). Annales de l'institut Henri Poincaré (B) Probabilités et Statistiques 16(1), 63–73 (1980)
4. Gelenbe, E., Mitrani, I.: Analysis and synthesis of computer systems. Academic Press, London (1980)