Prioritized Wireless Transmissions Using Random Linear Codes

Tuan Tran and Thinh Nguyen
School of EECS, Oregon State University
Corvallis, OR 97331, USA
{trantu, thinhq}@eecs.oregonstate.edu

***Abstract*—We investigate approximation algorithms for the problem of prioritized broadcast transmissions over independent erasure channels first described in [1]. In [1], the authors showed that under some settings, the achievable throughput regions for prioritized broadcast transmissions can be computed by a polynomial-time algorithm. In this paper, we study a class of approximate algorithms based on the Markov Chain Mote Carlo (MCMC) method, for obtaining the maximum sum of prioritized receiver's throughputs. Theoretical analysis and simulation results show the correctness and the convergence speed of the proposed algorithms.**

***Index Terms*—Network Coding, Random Network Coding, Prioritized Transmission.**

I. Introduction

Multimedia networking applications over wireless networks have gained much popularity in recent years. Unlike other types of traffic, multimedia traffics such as videos require some level of Quality of Service (QoS), e.g., minimum bandwidth, to meet user satisfactions. However, the current wireless networks such as WiFi were not designed for efficient provisioning of network resources in order to guarantee some specified QoS. To mitigate this lack of infrastructure support, scalable compression techniques are used to allow the sender to dynamically adjust the media bitrate to the current available bandwidth in real time. Notably, scalable video coding (SVC) is a class of techniques that allows a sender to adapt its video bit rate by partitioning a video bit stream into a base layer and several enhancement layers. The receiver is then able to view the video with higher quality when more layers are received. The base layer is the most important layer and must be present in order to have a reasonable video quality. The enhancement layers are organized in a hierarchical fashion such that the first enhancement layer must be present for the second enhancement layer to be useful, and the second enhancement layer must be present for the third enhancement layer to be useful, and so on. Naturally, this scenario leads to the general problem of prioritized transmissions where the goal is to transmit the most useful data under some resource constraints. In addition to multimedia networking applications, the prioritized transmission problem can also be used to model other applications that involve transmistting data with some hierarchical dependency, e.g. thread schedulers in multi-processor systems.

Prioritized Transmission. Formally, the prioritized transmission refers to the notion that, given M prioritized packets in the decreasing order of importance, a_1, a_2, $\ldots$, a_M, to be delivered to a receiver, then the packet a_i is useful to the receiver only if it has received all packets a_j with $j < i$ successfully. Thus, one metric to compare the performances between two prioritized transmission schemes, is to compare their useful throughput within a given time interval. The useful throughput can be abstractly represented $j-1$ where j is the position of the first lost packet. As described, it is trivial to show that, for the scenario that involves only a single sender and a single receiver, and the channel is modeled as a Bernoulli trial, the optimal scheme is the one that repeatedly transmits the highest priority packet until it is received correctly at the receiver, then transmits the next highest priority packet. The process repeats until there is no more packet to be transmitted, or the time exceeds a specified limit.

This problem, however is not trivial for the scenario in which there are one source and multiple receivers with a shared transmission medium. For example, consider a wireless broadcast scenario where multiple receivers want to receive a number of identical packets from a wireless broadcast base station. Due to the varied channel conditions at different receivers, the receivers will ultimately have different throughputs. Furthermore, because of the nature of the shared wireless medium, one transmission from the source to a particular receiver will take away resource from other receivers. Thus, the performances between two prioritized transmission schemes should not be based solely on the useful throughput of a particular receiver. Rather, it is better to base the decision on the sum throughput, i.e., throughput resulted from the summation of all the receiver's throughputs.

In this paper, we focus on the broadcast of prioritized information from a source to multiple receivers with following characteristics. Time is slotted and each packet is transmitted in one time slot. The source is allowed to use random linear network coding. Furthermore, the source is an oracle such that it knows precisely whether a packet is lost or received at any receiver in any future time slot. The task of the source is to schedule packet transmissions, i.e., selecting which packets in which time slots, in order to maximize the sum of the prioritized throughputs. The assumption of an oracle seems to make the problem less useful, however below are two examples for which the described model can be useful.

Example 1. (Thread scheduling): Consider a multi-processor system in which the processors share a common bus. When a processor uses the bus to transmit data, it prevents the processors from using it, thus it is a broadcast channel. Furthermore, a processor can be busy, e.g., due to other jobs that it cannot process the packets sent to it during some

978-1-4244-7190-4/10/$26.00

particular time slots. It is then reasonably to assume that this information is known to a global task scheduler. Thus, the task scheduler can be equivalently viewed as the oracle source. The goal for the task scheduler is to encode the data and schedule transmissions between the processors, to result in transmission the information with the shortest time.

Example 2. (Small-size cooperated wireless networks): Because of the small size, it is possible that a global scheduler can determine the transmit and receive schedules of all the nodes to avoid collisions. Furthermore, assuming that transmission errors due to the environment are negligible. Then if a sender knows the transmit and receive schedules of its neighbors, it can act as an oracle in an attempt to maximize the received throughputs of all its neighbors.

In [1], we have investigated the achievable throughput regions for the aforementioned broadcast scenario. Particularly, we have proposed a polynomial-time algorithm to check the optimality of an erasure pattern and use its transmission schedule to obtain the maximum throughput region for 2 and 3-receiver cases. In this paper, as an extended work, we are interested in finding the maximum sum throughput for more general case, i.e., scenarios consist of suboptimal erasure patterns with a large number of receivers and time slots. A heuristic approximation algorithm based on the MCMC method has been proposed to find the maximum sum throughput. Hereafter, for convenience, we use the term "throughput" for "sum throughput".

The organization of the paper as follows. In Section II, we provided some related work. In Section III we define notation and formally formulate the problem. The approximation algorithms are discussed in Section IV. In Section V we evaluate the performance of different approximation techniques through simulations. Finally, we conclude the paper with a few remarks in Section VI.

II. Related Work

Network coding (NC) techniques have been applied successfully to increase throughput in wireless ad hoc networks. A classical example first proposed by Wu *et al.* [2] for efficient information exchange in a wireless ad hoc network is shown in Fig. 1. Here, two nodes U_1 and U_2 want exchange their packets through U_3. Packet a sent by U_1 to U_2 is relayed through U_3, and packet b sent by U_2 to U_1 is relayed through U_3. As a result, U_3 has both a and b. In an existing wireless ad hoc network, U_3 has to perform two transmissions, one transmission for sending a to U_2, and another one for sending b to U_1. Now consider an NC scheme. Here, upon receiving a and b, U_3 can broadcast packet $a \oplus b$ to both U_1 and U_2. Since U_1 has a, it can recover b as $b = a \oplus (a \oplus b)$. Similarly, U_2 can recover a as $a = b \oplus (a \oplus b)$. The work related most to ours is that of Keller *et al.* in [3]. Both our work and theirs consider the broadcast problem where the source is an oracle. However, Keller *et al.* focus on minimizing decoding delay while our work focuses on the prioritized transmissions. Another related work to ours is that of Xue *et al.* [4]. In [4], Xue *et al.* used a Markov chain model to analyze the sum-rate of a 2-receiver unicast scenario using NC for a Bernoulli loss channel. On the other hand, our work focuses on the maximum throughput for the broadcast scenarios with a large number of receivers and time slots where the source is assumed to have complete knowledge of the channel.

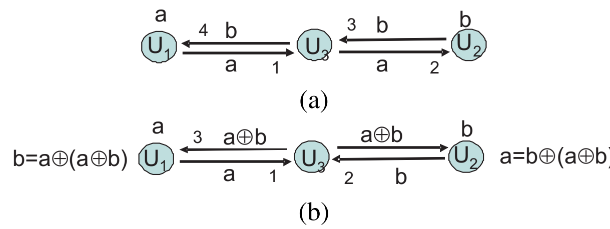


Fig. 1. Information exchange using (a) the store-and-forward scheme and (b) the NC scheme.

Receiver	Time slot 1	2	3	4
U_1	X	X	a_1	X
U_2	a_{12}	X	a_1	X
U_3	a_{12}	a_{13}	X	a_{13}
U_4	a_{12}	a_{13}	X	a_{13}

(a)

Receiver	Time slot 1	2	3	4
U_1	X	X	a_{12}	a_{12}
U_2	a_{12}	a_{12}	X	X
U_3	a_{12}	a_{12}	a_{12}	a_{12}
U_4	a_{12}	a_{12}	a_{12}	a_{12}

(b)

Fig. 2. (a) An optimal erasure pattern with an optimal transmission schedule produced by the OPT-Alg [1]. (b) A suboptimal erasure pattern with a suboptimal transmission schedule produced by the OPT-Alg. Recall that a coded packet a_{ij} is a linear combination of packets from a_i to a_j.

III. Model, Notation, and Previous Results

In this section, we review the system model, prioritized transmission problem and some useful definitions.

A. System Model

Assume that there is a source broadcasting M prioritized packets to N receivers over a shared channel. The prioritized packets in the decreasing order of importance, $a_1, a_2, \ldots, a_M$, are being delivered to all the receivers in T time slots. Assume that each packet takes one time slot to broadcast. A packet a_i is useful to a receiver only if it has received all packets a_j with $j < i$ successfully. Furthermore, let us assume that the number of broadcast packets M is less than or equal to the number of time slots T. The source is assumed to know the packet loss pattern at each receiver a priori.

B. Notation and Previous Results

Definition 1: A time slot is called a good time slot (GT) if the source can deliver a packet successfully to exactly one receiver during that time slot. A time slot is called a sharing good time slot (SGT) if the source can deliver a packet to two or more receivers successfully during that time slot.

Definition 2: An achievable throughput of a receiver U_i is r_i under a packet loss pattern, if it is possible for U_i to successfully decode consecutive packets $a_1, a_2, \ldots, a_{r_i}$.

Definition 3: An erasure pattern is said *optimal* if it is possible for every receiver to achieve a throughput that is equal to its total number of GTs and SGTs. Otherwise, an erasure pattern is said *suboptimal*.

Since an optimal erasure pattern provides the maximum possible throughput, in [1] we have proposed an algorithm (OPT-Alg) to check the optimality of an erasure pattern. To explain the optimality of the prioritized transmissions and definitions, two examples with $(N, T) = (4, 4)$ are illustrated in Fig. 2. Fig. 2(a) shows an optimal erasure pattern and a transmission schedule using the OPT-Alg. In this case, the maximum throughput is of 9 packets that is equal to the total number GTs and SGTs in the pattern. On the other hand, Fig. 2(b) represents a suboptimal erasure pattern with a transmission schedule using the OPT-Alg achieving a throughput of 8 packets. This is because receivers U_3 and U_4 have received duplicate packets. One should note that, however, in time slots 3 and 4, if we transmit coded packets a_{14}, then a higher throughput of 10 packets can be achieved.

► *Special scenarios*: It is straightforward to show that the OPT-Alg achieves the maximum throughput for the following patterns.

- Erasure patterns of 2-receiver and 3-receiver.
- Erasure patterns of N receivers satisfying $K_i = K_j$ for $\forall i \neq j \in \{1, 2, \dots, N\}$.
- Erasure patterns of N receivers in which any time slot is shared at most by two receivers.

In the next section, we describe approximate algorithms to obtain the maximum throughput for an arbitrary erasure pattern.

IV. Maximum Throughput Approximation

In this section we develop approximation techniques using the MCMC method to find the maximum throughput of an erasure pattern $E = (N, T)$ where N and T denote the number of receivers and time slots, respectively. First, a Simulated-Annealing based algorithm is proposed to find an optimal transmission scheme, and then it is improved by using some heuristic properties. But, before proceeding to the details of the algorithm, let us first introduce the MCMC method.

1) Markov Chain Monte Carlo: The main idea of the MCMC is to simulate an ergodic Markov chain (MC) such that its stationary distribution coincides with the target distribution [5]. For the sake of clarity, let us present a representative example to show how the MCMC works.

Consider a random variable X taking values in a sample space $\Omega = \{1, 2, \dots, m\}$ according to a target distribution $\{\pi : \pi_i = b_i/C, i \in \Omega\}$ where the size of the sample space $|\Omega|$ is very large and C is very complicated to compute. We now show how to construct an MC $\{X_n, n = 0, 1, \dots\}$ using a proposal transition matrix $Q = (q_{ij})$ as follows:

- Initialize $X_0 = i$. Set $n = 0$.
- When $X_n = i$, generates a new state Y according to the ith row of the transition matrix Q: $P(Y = j) = q_{ij}, j \in \Omega$.
- If $Y = j$

$$X_{n+1} = \begin{cases} Y & \text{w.p. } \alpha(i,j) \\ X_n & \text{w.p. } 1 - \alpha(i,j), \end{cases} \tag{1}$$

where $\alpha(i, j) = \min\left\{1, \frac{\pi_j q_{ji}}{\pi_i p_{ij}}\right\}$. We then can show that the process $\{X_n, n = 0, 1, \dots\}$ is an ergodic MC with its stationary distribution converges to π, the target distribution. As shown, using the MCMC method, we can generate samples according to the target distribution without computing C.

2) Simulated-Annealing Based Algorithm (SAB): Assume that receiver U_i has total K_i GTs and SGTs. We denote $K_m = \max_i K_i$ for $i = 1, 2, \dots, N$. Let Ω be the set of all possible transmission schemes, and let $S(x)$ be the throughput of a transmission scheme $x \in \Omega$. We represent each transmission scheme by a T-tuple ordered coded packets as $x = (c^1_{K_i}, c^2_{K_j}, \dots, c^T_{K_N})$ where $c^i_{K_j}$ is a coded packet transmitted at time slot i and generated from a batch of packets $a_1, a_2, \dots, a_{K_j}$. For example, considering the scenario illustrated in Fig. 2(b), one of the transmission schemes is $x = (c^1_2, c^2_2, c^3_2, c^4_2)$ where $c^i_n = \sum_{j=1}^{n} \gamma_j a_j$, and γ_j are drawn randomly from a large finite field. From now on, we specify a transmission scheme with its corresponding order of coded packets. Thus, the objective is to maximize the throughput

$$\max_{x \in \Omega} S(x) = \max_{x \in \Omega} \left\{ \sum_{i=1}^{N} r_i \right\}. \tag{2}$$

Recall that r_i is the total number of useful packets after decoding at receiver U_i. One should note that the number of possible transmission schemes in Ω is very large, $|\Omega| = (K_m)^T$. Hence, using exhaustive search, even for a small scenario, say $(N, T) = (5, 5)$, is infeasible for time-sensitive applications. Instead, we use MCMC-based technique by proposing a target distribution and Simulated-Annealing based algorithm to find a transmission scheme that approximately maximizes the throughput.

We first define the target distribution pdf to be the Boltzmann pdf:

$$f(x) = C e^{\frac{S(x)}{T_B}}, \tag{3}$$

where C is a normalized factor and T_B is temperature. We next define a neighbor of a transmission scheme in the sample space Ω:

Definition 4: A transmission scheme y is called a neighborhood of a transmission scheme x if x and y have only one different element.

In other words, y is generated from x by replacing an element of x with another one generated from a different set. For instance, the transmission scheme $x = (c^1_2, c^2_2, c^3_2, c^4_2)$ has a neighbor $y = (c^1_3, c^2_2, c^3_2, c^4_2)$ since x and y are different in the first element. Note that at any time slot j, a coded packet can only be generated with the maximum packet batch size is K_{jm}, $a_1, a_2, \dots, a_{K_{jm}}$, $K_{jm} = \max_i K_i$ for all receivers U_i having good condition at time slot j.

Finally, we propose a Simulated-Annealing based algorithm to sample from the target distribution. In particular, we propose a transition function $q(x, y)$ from x to one of its neighbors. Typically, an element of x is selected uniformly at random, and then it is replaced by one of the possible coded packets uniformly. We have

$$q(x, y) = q(y, x) = \frac{1}{T(K_m - 1)}. \tag{4}$$

Consequently, the acceptance probability that decides to move from the current state x to a new state y is given by

$$\begin{aligned} \alpha(x,y) &= \min\left\{1, \frac{f(y)q(y,x)}{f(x)q(x,y)}\right\} \\ &= \begin{cases} 1 & \text{if } S(y) \geq S(x) \\ e^{\frac{S(y)-S(x)}{T_B}} & \text{if } S(y) < S(x) \end{cases} \end{aligned} \tag{5}$$

The Boltzmann distribution becomes more and more concentrated around the global maximizer by gradually decreasing the temperature T_B. Pseudocode of the Simulated-Annealing based algorithm is described in Algorithm 1.

Algorithm 1 : Simulated-Annealing based algorithm (SAB).

Input: Erasure pattern $E = (N, T)$.
Output: A maximum throughput transmission scheme.

STEP 1: Initialize the starting state X_0 and temperature T_0. Set $n = 0$.
STEP 2: Generate a new state Y from the proposal $q(X_n, y)$.
STEP 3:
if $S(Y) \geq S(X_n)$ **then**
  $X_{n+1} \leftarrow Y$
else
  $U \sim U(0,1)$ {Generate a uniform random variable.}
  if $U < \alpha(X_n, Y) = e^{\frac{S(Y)-S(X_n)}{T_n}}$ **then**
    $X_{n+1} \leftarrow Y$
  else
    $X_{n+1} \leftarrow X_n$
  end if
end if
STEP 4: Decrease the temperature $T_{n+1} \leftarrow \beta.T_n$ where $\beta < 1$, increase n by 1 and repeat from **STEP 2** until stopping.
STEP 5: Return a scheme x that produces the maximum throughput.

---

3) Improved SAB Algorithm (ISAB): One should note that the sample space forms an MC in which some of the states are optimal. Each time when a coded packet is transmitted, the system moves from the current state to the next state. Thus, the problem can be viewed as an optimization problem of a deterministic Markov chain decision process (DMDP). Before proceeding to describe the DMDP, we have the following definition for "extended neighbors" in ISAB algorithm.

Definition 5: A state s_e is called an extended neighbor of a state s_t if the transmission scheme s_e is generated from the transmission scheme s_t by changing the transmitted coded packets in such a way that a receiver with the smallest number of GTs and SGTs does not have any useless coded packets.

By this definition, two neighbor states may have more than one different elements. These states are connected by a dash line in Fig. 3. We then define the DMDP as follows:

- **State space:** Ω consists of all possible transmission schemes. A state is denoted by a T-tuple in which an element and its index correspondingly represent the coded packet and the time slot used for transmitting the packet.

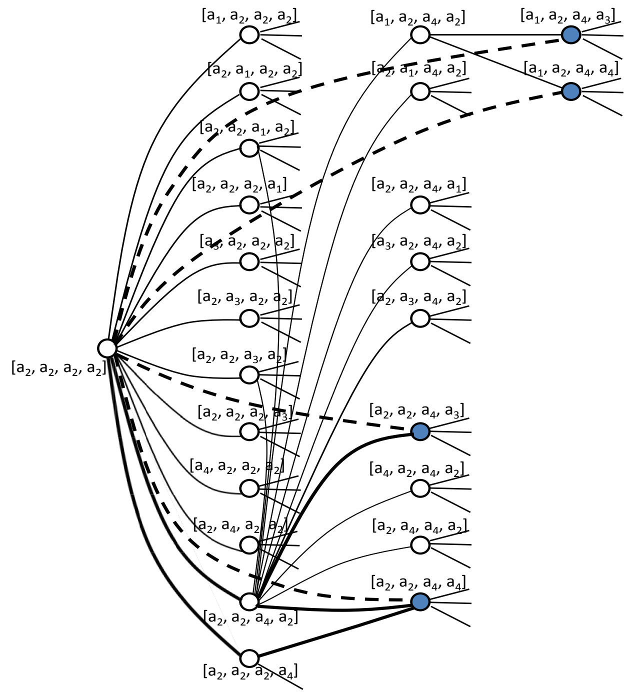


Fig. 3. A representation of the DMDP for the example in Fig. 2(b). The solid circles represent optimal states, i.e., transmission schemes achieve the maximum throughput; solid lines represent transitions in both SAB and ISAB while the dash lines represent transitions in the ISAB only.

For convenience, we eliminate the element superscript denoting the time slot index in each state.

- **Action space:** $\mathcal{A}$ consists of all possible coded packets.

$$\mathcal{A} = \{c_i : c_i = \sum_{j=1}^{i} \gamma_j a_j\}, i = \{1, 2, \ldots, K_m\} \tag{6}$$

- **Transition probability:** A mapping $\tau(s,t)$ from $\Omega \times \mathcal{A}$ to Ω, which specifies the system state at time $t+1$ when the decision maker chooses action $a \in \mathcal{A}$ in state s at time t. If s and j are neighbors of each other, then

$$p_t(j|s,a) = \begin{cases} 1, & \text{if } \tau(s,a) = j \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

That said, if given an erasure pattern, the structure of the MC is known since once an action is taken, the transition from the current state to the next state is determined. Within a number of epochs, i.e., the number of iterations, in order to reach to an optimal state, it is important that how the start state is initialized, and how to choose a neighbor to move to for the next state. To illustrate this point, in Fig. 3 we represent the DMDP of the erasure pattern given in Fig. 2. In this, we assume that the initial state is $[a_2, a_2, a_2, a_2]$; and the optimal states are indicated by the solid circles. Note that in SAB each state has $T(K_m - 1)$ neighbors. As seen, with this initialization, we can reach to an optimal state in 2 epochs by SAB. However, within 2 epochs, we cannot reach to any optimal state if the initial state is $[a_1, a_1, a_1, a_1]$. Also, by using the extended neighbor concept in ISAB, by 1 epoch we can reach to the optimal state as shown by the dash lines. Thus, the following properties are taken into account in the ISAB algorithm for initialization and transition:

- *Initialization*: Information is filled in the order from the receiver U_j with the smallest K_j to the receiver U_l with the largest K_l. This is because of the prioritized data property; when the important data is transmitted first, all receivers possibly receive them with high probability, and then transmitting the enhancement data later could reduce the duplicate information.
- *Transition*: The elements having conflict, i.e., duplicate or useless data received at some receiver in a time slot, will be replaced with high probability to the next state. This allows the system to move quickly from the current state to an optimal state. There is still however a low probability that the system will move from the current state to a state with a lower throughput, that allows the MC to explore all the possible states, and thus, avoiding convergence to a local optimum.

4) Convergence Correctness: It is straightforward to show that the samples drawn from our proposed algorithms, i.e., SAB and ISAB, will converge to the stationary distribution which models the optimal solution with high probability. In particular, we prove that the samples generated by the proposed algorithms form an ergodic MC.

Theorem 4.1: Samples drawn from the SAB and ISAB algorithms form an MC whose states satisfy the detailed balance equation:

$$\pi(x)P(x,y) = \pi(y)P(y,x) \quad \forall x, y \in \Omega, \tag{8}$$

where $\pi(x)$, $\pi(y)$ are the stationary distributions of the states x and y; $P(x,y)$ and $P(y,x)$ are respectively the transition probabilities from x to y and vice versa.

Proof: The proof is omitted due to the space limit. ■

V. Evaluation

A. Simulation Setup

We evaluate the performance of different approximation techniques in terms of the achievable maximum throughput and runtime. In particular, we compare 5 different techniques: Exhaustive search, Uniform sampler, Metropolis-Hasting based sampler, Simulated-Annealing based sampler, and our proposed technique Improved Simulated-Annealing based sampler.

- *Exhaustive Search (Exh. Search)*: This algorithm tries all possible transmission schemes, then returns a scheme that produces the best result.
- *Simulated-Annealing based sampler (Simulated)*: This algorithm procedure is described in Algorithm 1.
- *Uniform sampler (Uniform)*: This variant of the Simulated has steps described in the Algorithm 1, except at the **STEP** 3 the acceptance probability is set equal to 1, $\alpha(x,y) = 1$. This algorithm uses less computation but it is not optimal in finding the optimal solution.
- *Metropolis-Hasting based sampler (Metropolis)*: This is another variant of the Simulated. It has the same procedure as described in Algorithm 1, except that the Boltzmann temperature is fixed for all iterations.

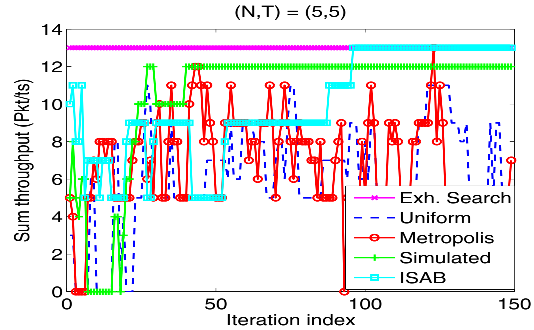
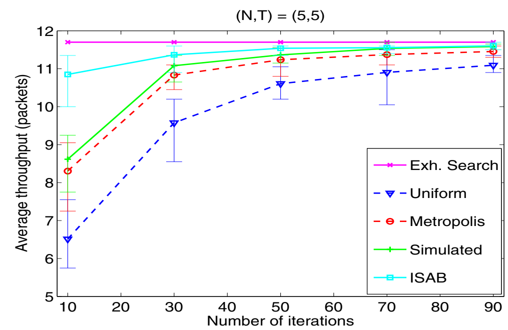
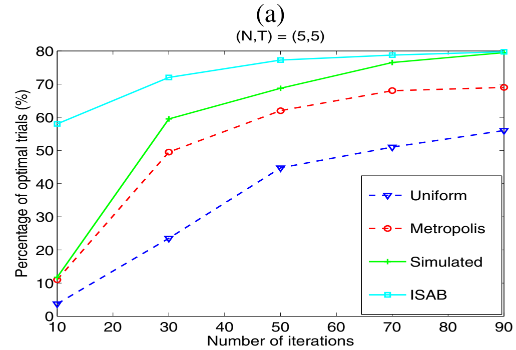


Fig. 4. Sample trace in terms of achievable throughput versus iteration index for an erasure pattern (N, T) = (5, 5); iterations $I = 150$.



(a)



(b)

Fig. 5. Simulation results for the erasure pattern (N, T) = (5, 5); (a) Average throughput versus the number of iterations; (b) Percentage of trials achieving the optimal solution versus the number of iterations.

- *Improved Simulated-Annealing based sampler (ISAB)*: The algorithm is described in the subsection IV-3.

To compare different techniques, we generate several patterns randomly, and then compute the average of the results. Since the achievable maximum throughput is a function of the number of receivers, time slots, and iterations, in our graphs we vary one parameter and keep the other two constant. Also, we do not plot performance of the Exhaustive Search for large erasure patterns due to its runtime increases exponentially.

B. Main Results

In the first experiment, we show the sample trace of different techniques with an erasure pattern $(N,T) = (5,5)$, and the number of iterations $I = 150$ (less than 1% of the expected transmission schemes). The sample traces in terms of throughput are illustrated in Fig. 4. As seen, in the first 100 iterations, ISAB is the only technique that finds an optimal state produced by the Exhaustive Search technique (indicated by the horizontal line on the top). As seen, the sample trace produced by the ISAB algorithm starts at very high throughput state due to a good initialization, and then quickly converges to an optimal state. On contrary, the other techniques cannot find any optimal state in 120 iterations. Particularly, the SAB converges to a local optimum. It takes a while before the ISAB converges to an optimal state, then it stays there with probability closed to 1.

We now examine how fast the approximation techniques converge to the optimal solution. In order to evaluate this, we fix the erasure pattern size $(N, T) = (5, 5)$ and vary the number of iterations. Specifically, convergence speed of each technique is evaluated by two metrics: average throughput and percentage of trials achieving the optimal solution. These are correspondingly shown in Figs. 5(a) and (b). Note that each point of the graph is computed by averaging the results from 100 trials. As seen, in Fig. 5(a), the Exhaustive Search achieves the optimal solution indicated by the horizontal line on the top. While in the ISAB, by starting from a good initial state and good choices of state transitions, it achieves the best performance among the approximation techniques. In addition, all the approximation techniques converge to the optimal solution, even with different rates. Especially, the ISAB converges fastest with about 100 iterations, about 1% of the sample space size. This is intuitively because the ISAB has higher flavor to move to the transmission schemes which have high throughput. Also, this state selection procedure has the ISAB obtained smaller variance comparing to the other techniques. Fig. 5(b) represents the percentage of the trials that find the optimal solution versus the number of iterations. As shown, the ISAB has almost 60% of the trials can find the optimal solution within 10 iterations; while the second best techniques, Simulate and Metropolis, achieve only 10%, and the Uniform achieves the worst performance with only 5%. As seen, the chance of finding an optimal solution increases when increasing the number of iterations.

We now compare the runtime of different techniques by fixing the number of time slots $T = 5$ and vary the number of receivers. In particular, for each instance of (N, T) we generate many patterns at random, and run each technique many times for a pattern. The plotted curves are averaged by the achieved results of each technique. Figs. 6 represents the runtime of different techniques. The runtime is the actual time that a technique needed to find an optimal state. As seen in Fig. 6, when the number of receivers increases, the runtimes of other techniques increase exponentially while that of the ISAB maintains almost constant.

In order to compare the performances of different techniques in large erasure patterns, we do not find the optimal solution using the Exhaustive search since the sample space increases exponentially. Especially, we fix the number of time slots $T = 5$, the time duration for running each algorithm is $t = 100$ milliseconds, and vary the number of receivers N. The average throughputs of different techniques are plotted in Fig. 7. As seen, when increasing the number of receivers, the performance gaps between the ISAB and the other techniques increase, meaning that the ISAB is scalable better with the problem size.

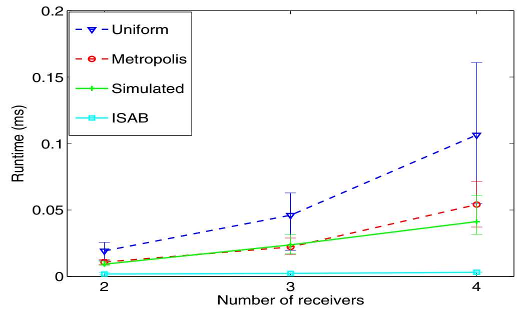


Fig. 6. Runtime versus the number of receivers; the number of time slots $T = 5$.

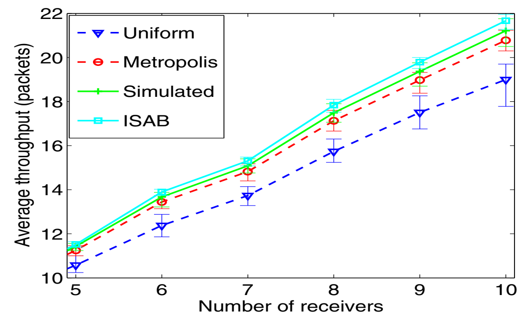


Fig. 7. Average throughput versus the number of receivers; number of receivers $T = 5$; time duration $t = 100$ (ms).

VI. Conclusions

In this paper we have investigated the approximation techniques to find the maximum throughput of prioritized transmissions from a source to multiple receivers via a shared and lossy channel. We show that the problem can be formulated as a combinatoric optimization problem and use a generic approximation method to find a closed-optimal solution. Particularly, an improved algorithm of the Simulated-Annealing based algorithm has been proposed, i.e., ISAB, to find an approximation optimal solution with the runtime much faster than the Exhaustive search. Several simulations have been provided to verify our proposed algorithms.

References

[1] T. Tran, T. Nguyen, and R. Raich, "On achievable throughput region of prioritized transmissions via network coding," in *The 18th International Conference on Computer Communications and Networks (ICCCN)*, Aug. 2009.

[2] Y. Wu, P. A. Chou, and S.-Y. Kung, "Information exchange in wireless networks with network coding and physical-layer broadcast," in *Technical Report MSR-TR-2004-78, Microsoft Research*, Aug. 2004.

[3] L. Keller, E. Drinea, and C. Fragouli, "Online broadcasting with network coding," in *Network Coding, Theory and Applications, NetCod 2008*, Jan. 2008.

[4] F. Xue and X. Yang, "Network coding and packet-erasure broadcast," in *SECON Workshops 2008*, Jun. 2008.

[5] M. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equations of state calculations by fast computing machines," in *Journal of Chemical Physics,*, 1953.