ADAPTIVE COMMUNICATIONS AND SIGNAL PROCESSING LABORATORY
CORNELL UNIVERSITY, ITHACA, NY 14853

# Chaff-inserting Algorithms and Robust Detection Algorithms for Information Flows

**Ting He and Lang Tong**

Technical Report No. ACSP-TR-02-07-01

February 2007

## I. Introduction

In this work, we present several algorithms for scheduling the transmission of information flows while inserting the minimum number of chaff packets. In addition, we also propose detection algorithms based on these chaff-inserting algorithms.

## II. Problem Formulation

Consider an $n$-hop path illustrated in Fig. 1. Assume that neighbor nodes on the path can communicate reliably. Let $S_i$ ($i = 1, \ldots, n$) be the process of transmission epochs of the $i$th node, *i.e.*,

$$S_i = (S_i(1),\ S_i(2),\ S_i(3), \ldots), \quad i = 1,\ 2, \ldots,\ n,$$

where $S_i(k)$ ($k \geq 1$) is the $k$th transmission epoch[1] of node $R_{i-1}$ (consider node $S$ as $R_0$).

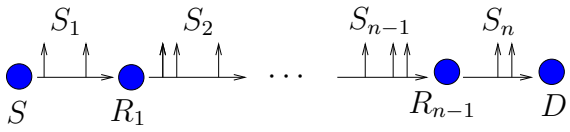


Fig. 1. Transmission activities along the path $S \to R_1 \to \ldots \to R_{n-1} \to D$.

If none of $S_i$ ($i = 1, \ldots, n$) belongs to the same information flow, assume them to be jointly independent. Otherwise, if $(S_i)_{i=1}^n$ is an $n$-hop information flow, then it can be decomposed into an information-carrying part $(X_i)_{i=1}^n$ and a chaff part $(W_i)_{i=1}^n$. That is, $S_i = X_i \bigoplus W_i$ ($i = 1, \ldots, n$)[2], where $(X_i)_{i=1}^n$ satisfies the following definition.

**Definition** A sequence of processes $(X_1, \ldots, X_n)$ is *a pure information flow* if there exist bijections $g_i : \mathcal{X}_i \to \mathcal{X}_{i+1}$ ($i = 1, \ldots, n-1$)[3] such that $g_i(s) - s \geq 0$ for all $s \in \mathcal{X}_i$, and $g_i$ satisfies certain communication constraints.

The bijection $g_i$ is a mapping between the transmission times of the same packets at nodes $R_{i-1}$ and $R_i$. The condition that $g_i$ is a bijection imposes a *packet-conservation* constraint, *i.e.*,

[1]Assume no simultaneous transmissions.

[2]The operator $\bigoplus$ is the superposition of processes $(a_1,\ a_2, \ldots)$ and $(b_1,\ b_2, \ldots)$, defined as $(a_i)_{i=1}^\infty \bigoplus (b_i)_{i=1}^\infty = (c_i)_{i=1}^\infty$ where $c_1 \leq c_2 \leq \ldots$ and $\{a_i\}_{i=1}^\infty \cup \{b_i\}_{i=1}^\infty = \{c_i\}_{i=1}^\infty$.

[3]We use $\mathcal{X}_i$ to denote the set of elements in $X_i$; similar rules hold for $S_i$ and $W_i$.

every information-carrying packet generates one and only one relay packet at each relay node. The condition $g_i(s) - s \geq 0$ is the *causality* constraint, which means that a packet cannot leave a node before it arrives. Communication constraints are additional constraints on $g_i$ which are imposed by the requirement of reliable communication. In this paper, we consider two types of commonly encountered constraints: bounded delay constraint and bounded memory constraint, as will be specified later.

It is worth emphasizing that an ($n$-hop) information flow is defined as $(S_i = X_i \bigoplus W_i)_{i=1}^n$; the chaff processes $W_i$ ($i = 1, \ldots, n$) are not subject to any of the above constraints.

Suppose that the detector starts at $t_0$ and takes observations for a duration $t$. We are interested in testing the following hypotheses:

$$\begin{aligned} \mathcal{H}_0: &\quad S_1, S_2, \ldots, S_n \text{ are jointly independent,} \\ \mathcal{H}_1: &\quad (S_i)_{i=1}^n \text{ contains an information flow,} \end{aligned}$$

by analyzing $S_i \cap [t_0, t_0 + t]$ ($i = 1, \ldots, n$)[4]. We say that $(S_i)_{i=1}^n$ *contains an information flow* if $\exists I \subseteq \{1, \ldots, n\}$ such that $(S_i)_{i \in I}$ is an information flow.

Assume that the detector knows $\Delta$ but not $t_0$ or traffic before $t_0$. This is a nonparametric hypothesis testing problem; no statistical assumptions are made at this point (although additional assumptions under $\mathcal{H}_0$ are needed for detailed analysis).

To characterize the amount of chaff, we introduce the following definition.

**Definition** If $(S_i)_{i=1}^n$ is an information flow, then its *chaff-to-traffic ratio* (CTR) in the interval $[t_0, t_0 + t]$ is defined as

$$\text{CTR}(t; t_0) = \frac{\sum_{i=1}^n |\mathcal{W}_i \cap [t_0, t_0 + t]|}{\sum_{i=1}^n |\mathcal{S}_i \cap [t_0, t_0 + t]|},$$

*i.e.,* $\text{CTR}(t; t_0)$ is the fraction of chaff packets in the interval $[t_0, t_0 + t]$.

As for communication constraints, we consider the following two types of information flows.

[4]Given a process $S = (s_j)_{j=1}^\infty$, $S \cap [a, b]$ is the truncated process defined as $(s_j)_{j=k}^l$, where $s_{k-1} < a \leq s_k$, and $s_l \leq b < s_{l+1}$.

**Definition** A sequence of processes $(X_1, \ldots, X_n)$ is *a pure information flow with bounded delay* $\Delta$ if it is a pure information flow, and in addition to the packet-conservation and the causality constraints, the mapping $g_i$ ($i = 1, \ldots, n-1$) satisfies that $g_i(s) - s \leq \Delta$ for all $s \in \mathcal{X}_i$.

**Definition** A sequence of processes $(X_1, \ldots, X_n)$ is *a pure information flow with bounded memory* $M$ if it is a pure information flow, and in addition to the packet-conservation and the causality constraints, the mapping $g_i$ ($i = 1, \ldots, n-1$) satisfies that for any $t \geq 0$,

$$0 \leq |\mathcal{X}_i \cap [0,\, t]| - |\mathcal{X}_{i+1} \cap [0,\, t]| \leq M.$$

The condition $g_i(s) - s \leq \Delta$ is a *bounded delay* constraint which requires every information-carrying packet to be relayed within delay $\Delta$. The condition $|\mathcal{X}_i \cap [0,t]| - |\mathcal{X}_{i+1} \cap [0,t]| \in [0, M]$ is a *bounded memory* constraint which requires the number of information-carrying packets stored at a relay node (*i.e.,* the total arrivals minus the total departures) to be bounded by $M$.

Note that chaff packets do not have to satisfy any of the above constraints.

## III. Optimal Chaff-inserting Algorithms

In this section, we will show how to schedule the transmission of information-carrying packets according to given transmission pattern such that the number of inserted chaff packets is minimized.

### *A. Inserting Chaff for Bounded Delay Flows*

*1) Two-hop Flows:* Consider scheduling a 2-hop information flow under the bounded delay constraint. To this end, Blum *et al.* in [1] proposed a greedy algorithm called "Bounded-Greedy-Match" (BGM). Given $(S_1,\, S_2)$, BGM sequentially matches every packet in $S_1$ with the first unmatched packet within delay $\Delta$ in $S_2$, and labels all the unmatched packets as chaff.

We combine the insertion of chaff and the transmission of information-carrying packets into the implementation presented in Table I.

This implementation of BGM uses two pointers $m$ and $n$ to record the current epochs examined in $S_1$ and $S_2$, and keeps updating $m$ and $n$ depending on whether the match is successful or not. Its complexity is $O(|S_1| + |S_2|)$.

TABLE I

BOUNDED-GREEDY-MATCH (BGM).

Bounded-Greedy-Match($S_1,\ S_2,\ \Delta$):

```
m = n = 1;
while m ≤ |S_1| and n ≤ |S_2|
  if s_n^(2) − s_m^(1) < 0
    s_n^(2) = chaff; n = n + 1;
  else if s_n^(2) − s_m^(1) > Δ
      s_m^(1) = chaff; m = m + 1;
    else
      (s_m^(1), s_n^(2)) = arrival and departure times of a packet;
      m = m + 1; n = n + 1;
    end
  end
end
```

It is shown in [1] that BGM inserts the minimum number of chaff packets in embedding an information flow with bounded delay into a pair of arbitrary point processes[5].

*2) Multi-hop Flows:* Now consider extending BGM to a chaff-inserting algorithm for multi-hop information flow under the bounded delay constraint. The extended algorithm is called "Multi-Bounded-Delay-Relay" (MBDR). Implementation of the algorithm is presented in Table II. The complexity of MBDR is[6] $O(n^2|S_1|)$.

It can be shown that MBDR inserts the minimum number of chaff packets for any processes.

### *B. Inserting Chaff for Bounded Memory Flows*

*1) Two-hop Flows:* Consider scheduling the transmissions for a flow of information-carrying packets over two hops under the bounded memory constraint. We propose an algorithm called

[5]The original proof in [1] is for independent binomial processes, but it holds for arbitrary processes.

[6]The dominating step is the recursive computation of $C_{i,\,j}$'s. Suppose the maximum rate of $S_1, \ldots, S_{n+1}$ is $\lambda$, and thus there are at most $(i-1)\lambda\Delta$ points in $C_{i,\,j}$ on the average; the selection of these points takes $(2i-3)\lambda\Delta$ steps. The total complexity can be calculated by $|S_1| \sum_{i=2}^{n+1} (2i-3)\lambda\Delta = \lambda\Delta n^2|S_1|$.

TABLE II

MULTI-BOUNDED-DELAY-RELAY (MBDR).

Multi-Bounded-Delay-Relay($S_1, \ldots, S_{n+1}, \Delta$):

```
(p_i)_{i=1}^{n+1} = (0, ..., 0);
for j = 1 : |S_1|
  C_{1,j} = {s_j^{(1)}};
  for i = 1 : n
    for all s ∈ C_{i,j} in increasing order
      for all t ∈ T_{S_{i+1}} ∩ [s, s + Δ], t > p_{i+1}, and t ∉ C_{i+1,j}
        t.predecessor = s;
        add t to C_{i+1,j};
      end
    end
  end
  if |C_{n+1,j}| ≠ 0
    t_{n+1} = min(C_{n+1,j});
    for i = n : −1 : 1
      t_i = t_{i+1}.predecessor;
    end
    (t_i)_{i=1}^{n+1} is the relay sequence for s_j^{(1)};
    (p_i)_{i=1}^{n+1} = (t_i)_{i=1}^{n+1};
  end
end
for all s ∈ ⋃_{i=1}^{n+1} S_i and s ∉ any selected relay sequences
  s = chaff;
end
```

"Bounded-Memory-Relay" (BMR) for this purpose.

Algorithm BMR assigns a packet received or transmitted at a certain epoch to be chaff if and only if the acceptance of this packet will cause a memory underflow (memory size $< 0$) or overflow (memory size $> M$).

A pseudo code implementation of BMR is given in Table III.

Note that once BMR marks out the chaff packets, the order in which information-carrying packets are transmitted is irrelevant as far as the memory constraint is concerned. The complexity

TABLE III

BOUNDED-MEMORY-RELAY (BMR).

```
Bounded-Memory-Relay(S_1, S_2, M):
  S = merge(S_1, S_2);
  d = 0;
  for w = 1 : |S|
    if (d = M and s_w ∈ T_{S_1}) or (d = 0 and s_w ∈ T_{S_2})
      s_w = chaff;
    else if s_w ∈ T_{S_1}
          send a packet to the node at s_w;
          d = d + 1;
      else
          relay a packet from the node at s_w;
          d = d − 1;
      end
    end
  end
```

of BMR is only $O(|S_1| + |S_2|)$.

It can be shown that BMR is optimal in the sense that given any $(S_1,\ S_2)$, it inserts the minimum number of chaff packets in scheduling an information flow with bounded memory.

*2) Multi-hop Flows:* If we want to send information flows over multiple hops, we can generalize BMR to an algorithm called "Multi-Bounded-Memory-Relay" (MBMR) for this purpose. Implementation of MBMR is given in Table IV.

Algorithm MBMR has complexity $O(\sum_{i=1}^{n+1} |S_i|)$. It uses $M_i$ to record the number of packets stored in the $i$th relay node. The algorithm keeps track of $M_i$ and guarantees that $M_i$ is always between $0$ and $M$, which implies that the scheduling found by MBMR satisfies the bounded memory constraint. Algorithm MBMR is optimal in the sense that the number of chaff packets is minimized.

TABLE IV

MULTI-BOUNDED-MEMORY-RELAY (MBMR).

Multi-Bounded-Memory-Relay($S_1, \ldots, S_{n+1}, M$):

$S =$ merge$(S_1, \ldots, S_{n+1})$;
$(M_1, \ldots, M_n) = (0, \ldots, 0)$;
for $w = 1 : |S|$
  $i$ is such that $s_w \in \mathcal{T}_{S_i}$;
  if $i = 1$
    if $M_1 < M$
      $s_w =$ a packet sent to node 1;
      $M_1 = M_1 + 1$;
    else
      $s_w =$ chaff;
    end
  else if $i = n + 1$
      if $M_n > 0$
        $s_w =$ a packet departing from node $n$;
        $M_n = M_n - 1$;
      else
        $s_w =$ chaff;
      end
    else if $M_{i-1} > 0$ & $M_i < M$
        $s_w =$ a packet relayed from node $(i - 1)$ to node $i$;
        $M_{i-1} = M_{i-1} - 1$;
        $M_i = M_i + 1$;
      else
        $s_w =$ chaff;
      end
    end
  end
end

## IV. Detection Algorithms

In this section, we present detection algorithms to solve the hypothesis testing problem proposed in Section II. The algorithms are based on the optimal chaff-inserting algorithms. Each detection algorithm makes detection if the minimum CTR in the measurements is bounded by a predetermined threshold. In the following, we will present the algorithms and explain why the CTR's calculated by these algorithms are minimal.

### *A. Detecting Bounded Delay Flows*

*1) Pairwise Detection:* Algorithm "Detect-Bounded-Delay" (DBD) is derived to detect 2-hop information flows with bounded delay. It does detection with the help of the optimal chaff-inserting algorithm BGM. Implementation of DBD is presented in Table V. The complexity of DBD is $O(N)$, where $N$ is the joint sample size, *i.e.,* the total number of examined packets in $S_1 \bigoplus S_2$.

Suppose $\mathcal{H}_1$ is true. Then the actual number of chaff packets in $S_1 \bigoplus S_2$ has to be no smaller than $C$ because BGM is optimal, and chaff packets in $[0,\ \Delta)$ in $S_2$ have been ignored (because they may be the relay packets of packets arriving before time $0$). It means that the actual CTR has to be more than $1/(1+\lambda'\Delta)$ to evade DBD. Therefore, DBD is robust for CTR$\leq 1/(1+\lambda'\Delta)$.

*2) Joint Detection:* Based on similar idea, the detection can be extended over multiple hops by utilizing the multi-hop chaff-inserting algorithm MBDR. The algorithm, called "Detect-Multi-Bounded-Delay" (DMBD), is presented in Table VI. The complexity of DMBD is $O(nN)$.

Since MBDR inserts the minimum number of chaff packets, and chaff packets in $S_1 \cap [0,\ (i-1)\Delta)$, which may be the relay packets of information-carrying packets sent before the detector starts, are ignored, $C$ is always a lower bound on the actual number of chaff packets in $S_1 \bigoplus \cdots \bigoplus S_{n+1}$, which means that the actual CTR has to be larger than $\gamma_n$ to evade DMBD. Therefore, DMBD is robust for CTR$\leq \gamma_n$.

### *B. Detecting Bounded Memory Flows*

To detect information flows with bounded memory, we develop algorithms based on the optimal chaff-inserting algorithms BMR and MBMR.

TABLE V

DETECT-BOUNDED-DELAY (DBD).

Detect-Bounded-Delay($S_1$, $S_2$, $\Delta$, $N$, $\lambda'$):

$i = j = 1$;
$C = 0$;
while $i + j \leq N$
  if $s_j^{(2)} - s_i^{(1)} < 0$
    if $s_j^{(2)} \geq \Delta$
      $C = C + 1$;
    end
    $j = j + 1$;
  else if $s_j^{(2)} - s_i^{(1)} > \Delta$
      $C = C + 1$; $i = i + 1$;
    else
      $i = i + 1$; $j = j + 1$;
    end
  end
end
return $\begin{cases} \mathcal{H}_1 & \text{if } \frac{C}{N} \leq \frac{1}{1+\lambda'\Delta}, \\ \mathcal{H}_0 & \text{o.w.}; \end{cases}$

*1) Pairwise Detection:* Algorithm "Detect-Bounded-Memory" (DBM) detects 2-hop information flows based on the chaff-inserting algorithm BMR. Implementation of DBM is given in Table VII. Algorithm DBM has complexity $O(N)$.

It is shown in [2] that the actual number of chaff packets in $S_1 \bigoplus S_2$ is lower bounded by $C$. It implies that DBM has no miss detection for information flows with CTR up to $1/(1 + M')$, and is therefore robust for CTR$\leq 1/(1 + M')$.

*2) Joint Detection:* Now we extend DBM to a joint detection algorithms called "Detect-Multi-Bounded-Memory" (DMBM), which is presented in Table VIII. Algorithm DMBM has complexity $O(N)$.

The value of $C$ is actually the number of times that memory overflow or underflow would have occurred if chaff packets had not been inserted. For an information flow with bounded

TABLE VI

DETECT-MULTI-BOUNDED-DELAY (DMBD).

Detect-Multi-Bounded-Delay$(S_1, \ldots, S_{n+1}, \Delta, N, \gamma_n)$:

$C = 0$;
$(J_1, \ldots, J_{n+1}) = (I_1, \ldots, I_{n+1}) = (0, \ldots, 0)$;
$K_1 = 0$;
for $i = 2 : n+1$
  $K_i = \sup\{k : s_k^{(i)} \leq (i-1)\Delta\}$;
end
$j = 1$;
while $\sum_{i=1}^{n+1} J_i < N$ & $j \leq |S_1|$
  $C_{1,j} = \{s_j^{(1)}\}$;
  for $i = 1 : n$
    for all $s \in C_{i,j}$ in increasing order
      for all $t \in \mathcal{T}_{i+1} \cap [s, s+\Delta]$, $t > s_{J_{i+1}}^{(i+1)}$, and $t \notin C_{i+1,j}$
        $t$.predecessor $= s$;
        add $t$ to $C_{i+1,j}$;
      end
    end
  end
  if $|C_{n+1,j}| \neq 0$
    $I_{n+1} = \min\{k : s_k^{(n+1)} \in C_{n+1,j}\}$;
    for $i = n : -1 : 1$
      $I_i$ is such that $s_{I_i}^{(i)} = s_{I_{i+1}}^{(i+1)}$.predecessor;
    end
    $C = C + \sum_{i=1}^{n+1} (I_i - \max(J_i, K_i) - 1)$;
    $(J_1, \ldots, J_{n+1}) = (I_1, \ldots, I_{n+1})$;
  end
  $j = j + 1$;
end
$C = C + max(N - \sum_{i=1}^{n+1} \max(J_i, K_i), 0)$;
$\tilde{N} = max(\sum_{i=1}^{n+1} J_i, N)$;
return $\begin{cases} \mathcal{H}_1 & \text{if } \frac{C}{\tilde{N}} \leq \gamma_n \\ \mathcal{H}_0 & \text{o.w.;} \end{cases}$

TABLE VII

DETECT-BOUNDED-MEMORY (DBM).

Detect-Bounded-Memory($S_1,\ S_2,\ M,\ N,\ M'$):

$S = \text{merge}(S_1,\ S_2)$;

$d = d_{\max} = d_{\min} = 0$;

$C = 0$;

for $w = 1 : N$

if ($s_w \in \mathcal{T}_{S_1}$, $d - d_{\min} = M$) or ($s_w \in \mathcal{T}_{S_2}$, $d_{\max} - d = M$)

$C = C + 1$;

else

$$d = \begin{cases} d+1 & \text{if } s_w \in \mathcal{T}_{S_1}, \\ d-1 & \text{if } s_w \in \mathcal{T}_{S_2}; \end{cases}$$

$d_{\max} = \max(d_{\max},\ d)$;

$d_{\min} = \min(d_{\min},\ d)$;

end

end

$$\text{return} \begin{cases} \mathcal{H}_1 & \text{if } \frac{C}{N} \leq \frac{1}{1+M'}, \\ \mathcal{H}_0 & \text{o.w.}; \end{cases}$$

memory $M$, the actual number of chaff packets is at least $C$, and the CTR has to be larger than $\tau_n$ to evade DMBM. Therefore, DMBM is robust for CTR$\leq \tau_n$.

## REFERENCES

[1] A. Blum, D. Song, and S. Venkataraman, "Detection of Interactive Stepping Stones: Algorithms and Confidence Bounds," in *Conference of Recent Advance in Intrusion Detection (RAID)*, (Sophia Antipolis, French Riviera, France), September 2004.

[2] T. He and L. Tong, "Detecting Encrypted Stepping-stone Connections." accepted to IEEE Trans. on Signal Processing, 2006.

TABLE VIII

DETECT-MULTI-BOUNDED-MEMORY (DMBM).

Detect-Multi-Bounded-Memory$(S_1, \ldots, S_{n+1}, M, N, \tau_n)$:

$S =$ merge$(S_1, \ldots, S_{n+1})$;
$(M_1, \ldots, M_n) = (U_1, \ldots, U_n) = (V_1, \ldots, V_n) = (0, \ldots, 0)$;
$C = 0$;
for $w = 1 : N$
  $i$ is such that $s_w \in \mathcal{T}_{S_i}$;
  if $i = 1$
    if $M_1 - V_1 < M$
      $M_1 = M_1 + 1$;
      $U_1 = max(U_1, M_1)$;
    else
      $C = C + 1$;
    end
  else if $i = n + 1$
      if $U_n - M_n < M$
        $M_n = M_n - 1$;
        $V_n = min(V_n, M_n)$;
      else
        $C = C + 1$;
      end
    else if $U_{i-1} - M_{i-1} < M$ & $M_i - V_i < M$
        $M_{i-1} = M_{i-1} - 1$;
        $M_i = M_i + 1$;
        $V_{i-1} = min(V_{i-1}, M_{i-1})$;
        $U_i = max(U_i, M_i)$;
      else
        $C = C + 1$;
      end
    end
  end
end
return $\begin{cases} \mathcal{H}_1 & \text{if } \frac{C}{N} \leq \tau_n, \\ \mathcal{H}_0 & \text{o.w.;} \end{cases}$