Proceedings of the IASTED International Conference
Communication, Network, and Information Security
October 9–11, 2006, MIT, Cambridge, MA, USA
ISBN CD: 0-88986-636-8

COVERT TIMING CHANNEL CAPACITY OF RATE MONOTONIC REAL-TIME SCHEDULING ALGORITHM IN MLS SYSTEMS

Joon Son and Jim Alves-Foss
Center for Secure and Dependable Systems
University of Idaho
POBOX 441008 Moscow, ID 83844-1008
email: [son2320,jimaf]@uidaho.edu

ABSTRACT
Real-time systems must satisfy timing constraints. In our previous work, we showed that a covert timing channel cannot be completely closed in some system configurations due to the timing constraints imposed by the Rate-Monotonic (RM) real-time scheduling algorithm. In this paper, we construct a probabilistic model to measure two quantities of a covert timing channel in RM based systems: channel capacity and quantity of specific information. We show how these two metrics can be calculated from our probabilistic model and why they are useful metrics in evaluation of a covert (timing) channel.

KEY WORDS
Covert timing channel capacity, quantity of specific information, Rate-Monotonic scheduling.

1 Introduction

Many researchers formally define (real-time) information flow via a communication path called a covert (timing) channel in multi-level secure systems. *A covert timing channel* is an illicit communication path in which one entity ($High$) signals information to another entity (Low) in violation of the security policy by modulating its use of system resources in such a way that this manipulation affects the response time observed by Low. One central idea of the proposed definitions of information flow is the following: there is no information flow via a covert channel if Low cannot deduce anything about the activities of $High$.

Numerous papers [4, 8, 10, 11, 13] have presented mathematical frameworks for measuring the amount of possible information leakage through a covert timing channel for various systems. We take as our area of interest a real-time system. A real-time operating system employs a scheduling algorithm to schedule multiple tasks so that each task can meet its real-time constraints. Most real-time scheduling algorithms are priority based. In priority-based scheduling, control of the CPU is always given to the highest priority task ready to run. How a scheduling priority is assigned to a task and when the highest priority task runs on the CPU, however, are determined by the type of scheduling algorithms used.

In previous work [12], we analyzed how a covert timing channel is created and exploited by a $High$ task ($\mathcal{T}_H$) and a Low task ($\mathcal{T}_L$) while a third party task ($\mathcal{T}_N$) is concurrently running with them under Rate Monotonic scheduling [9]. The RM scheduling algorithm is one of the most widely used scheduling strategies due to its rich theoretical background and simplicity of implementation. Our previous work [12] is based upon the possibilistic approach, which assumes that the probability of a timed action of a task is not known. Using the possibilistic approach, we formally proved that in some system configurations it will not be possible to completely close the covert timing channel due to the timing constraint of the RM scheduling algorithm. In this paper, we incorporate a probabilistic model into our previous work and consider a probabilistic covert channel under RM scheduling. Using a probabilistic model, our goal is to measure two quantities: the capacity of a covert channel via Shannon's information theory [5] and a quantity called *specific* information [6]. The channel capacity represents the *maximum* average amount of information per symbol (bit/symbol) that can be transmitted through a given noisy channel. The quantity of specific information signifies the amount information carried by a specific output symbol regarding the range of input symbols transmitted. Although channel capacity is a useful metric in assessing the overall severity of a covert channel, the quantity of specific information is an effective metric when one wants to know the amount of information gained from a specific observation, rather than the average.

One important characteristic of a real-time operating system is that it performs operations at fixed, predetermined times (called preemption points) or within predetermined intervals [14]. For the purpose of this paper we define a unit of time as the interval between preemption points, giving us a well-defined discrete time domain in which to analyze covert timing channels.

2 Rate-Monotonic Scheduling Algorithm

2.1 Background, Notations, and Assumptions

Liu and Layland's Rate-Monotonic (RM) scheduling algorithm [9] has become one of the most widely used schedul-

C_i:	Worst case computation (execution) time required by an invocation of task $\mathcal{T}_i$.
T_i:	Lower bound between successive arrivals of a task $\mathcal{T}_i$. This is the period of a periodic task $\mathcal{T}_i$.
D_i:	The deadline for each invocation of task $\mathcal{T}_i$, measured from its arrival time. Usually $D_i \leq T_i$.
R_i:	Worst case response time for an invocation of task $\mathcal{T}_i$, measured from its arrival time to its termination time. A schedulable task must have $R_i \leq D_i$.

Figure 1. Notations used to characterize a task

ing algorithms in real time systems. It provides:

- Tasks with shorter periods (higher request rates) will have higher priorities.
- A priority assigned to a task is fixed.
- A currently executing task is preempted by a newly arrived task with a higher priority (shorter period).

The mathematical symbols used are defined in Figure 1. In order to simplify our analysis, the following assumptions are applied to a periodic task:

A1. The periodic tasks running on a single processor are *independent* (no shared resources among tasks other than the processor).

A2. The time between successive arrivals of a task is fixed as T_i (a task is activated at a constant rate T_i).

A3. The deadline for each invocation of a task is equal to the period ($D_i = T_i$).

A4. A task is released as soon as it arrives.

A5. The initial release time of all tasks is zero.

From the above assumptions, a periodic task $\mathcal{T}_i$ can be completely characterized as $\mathcal{T}_i(T_i, C_i)$. Thus, we can denote a set of periodic tasks running under the RM scheduling algorithm as:

$$\Gamma_{RM} = \{\mathcal{T}_i(T_i, C_i),\ \ i = 1 \ldots n\}$$

2.2 Worst Case Response Time Analysis

Let $\Gamma_{RM} = \{\mathcal{T}_i(T_i, C_i),\ \ i = 1 \ldots n\}$. Let $\pi(\mathcal{T}_i)$ represent a scheduling priority assigned to a task $\mathcal{T}_i$. Assume that $\pi(\mathcal{T}_1) > \pi(\mathcal{T}_2) \cdots > \pi(\mathcal{T}_n)$. Joseph and Pandya [7] showed that a task set Γ_{RM} will meet all its deadlines if:

$$\begin{aligned} &\forall \mathcal{T}_i \in \Gamma_{RM} \quad R_i \leq D_i, \\ &\qquad where\ R_i = C_i + I_i \\ &\qquad and\ \ I_i = \sum_{j=1}^{i-1} \left\lceil \frac{R_i}{T_j} \right\rceil C_j \end{aligned} \tag{1}$$

The definition of R_i in Eq. (1) is recursive. If a set of tasks is not schedulable, one cannot find a solution for R_i of the lowest priority task which satisfies $R_i \leq D_i$ [1].

3 Covert Timing Channel Analysis

In this section, we introduce a model which describes a covert timing channel between a high-level task $\mathcal{T}_H$ and a low-level task $\mathcal{T}_L$, while a third party (system) task $\mathcal{T}_N$ (noise) is running concurrently with them under RM scheduling. We make the following assumptions in building our model:

- **Periodic tasks:** Three tasks are running under RM scheduling, i.e., $\Gamma_{RM} = \{\mathcal{T}_i(T_i, C_i),\ \ i = N, H, L\}$.
- **Schedulability:** Γ_{RM} is schedulable.
- **Scheduling priority:** We assume $T_N < T_H < T_L$, i.e., $\pi(\mathcal{T}_N) > \pi(\mathcal{T}_H) > \pi(\mathcal{T}_L)$. The outcome of analysis will be very similar when $T_H < T_N < T_L$, i.e., $\pi(\mathcal{T}_H) > \pi(\mathcal{T}_N) > \pi(\mathcal{T}_L)$. What is important here is that a low level task $\mathcal{T}_L$ is assumed to be a task with the longest period. There will be no covert timing channel if $\mathcal{T}_L$ has the shortest period; however, this may not be a viable solution for some real-time applications.
- **Ability of *High***: $High$ is given as much opportunity as possible for creating a covert timing channel. At every release time, the computation of a task $\mathcal{T}_H$ may vary from one unit to C_H units of time.
- **Ability of *Low***: Low cannot measure the time between any two occurrences of context switch. However, for each period, Low is able to assess the response time of its own task (the time between the submission of its task to a scheduler and the notification that it is completed). We assume there is no overhead associated with task submission and notification.
- **Timing behavior of $\mathcal{T}_N$:** At every release time, the computation time of task $\mathcal{T}_N$ may vary from one unit to C_N units of time. However, its timing behavior is nondeterministic in a sense that neither Low nor $High$ can reliably predict the computation time performed by $\mathcal{T}_N$ at each release.
- **Periods of $\mathcal{T}_N$, $\mathcal{T}_H$, and $\mathcal{T}_L$:** To simplify our analysis, we assume that T_N divides T_L and T_H divides T_L. With this assumption, Low can obtain a single output sample (the response time of its own task) by monitoring at most T_L units of time. In addition, the number

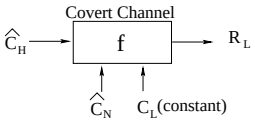

Figure 2. Covert channel model

of periods of $\mathcal{T}_N$ and $\mathcal{T}_H$ which affect the response time of Low can be well defined. This assumption is not necessary, but without the assumption, Low has to wait for the maximum of $l.c.m.$ (T_N, T_H, T_L) [1] units of time in each period to sample the output.

- **Sampling factor of *Low***: In order to detect (sample) a response time of it own task, Low submits a task with known computation time to a real-time scheduler. This fixed computation time is called the sampling factor of $\mathcal{T}_L$. We use C_L to denote the sampling factor.
- **Pre-agreement:** $High$ has some pre-agreement with Low that it begins to transmit a new symbol every T_L units of time (Low receives (samples) an output every T_L units of time).

3.1 Extending Response Time Analysis

We extend Eq. (1) to calculate all possible response times (not just the worst response time) of a task $\mathcal{T}_L$. Assume that $\Gamma_{RM} = \{\mathcal{T}_i(T_i, C_i),\ i = N, H, L\}$ passes the schedulability test of Eq. (1).

Let $\hat{C}_H[k]$ denote the computation time of a task $\mathcal{T}_H$ at the k^{th} release (during the k^{th} period), $1 \leq \hat{C}_H[k] \leq C_H$. We call $\hat{C}_H[k]$ a *timed action (computation time)* of a task $\mathcal{T}_H$ at the k^{th} release. Let $\hat{C}_H$ be a vector (tuple) which represents a sequence of timed actions of a task $\mathcal{T}_H$ from the first up to the $\lceil \frac{T_L}{T_H} \rceil^{th}$ release:

$$\hat{C}_H = (\hat{C}_H[1], \hat{C}_H[2], \ldots, \hat{C}_H[k], \ldots, \hat{C}_H[\left\lceil \frac{T_L}{T_H} \right\rceil])$$

Let V_h be a set with all possible combinations of $\hat{C}_H$:

$$V_h = \{\hat{C}_H \mid \hat{C}_H[k] \in \{1, \ldots, C_H\},\ k \in \{1, \ldots, \left\lceil \frac{T_L}{T_H} \right\rceil\}\}$$

For instance, if $1 \leq \hat{C}_H[k] \leq C_H$, then

$$V_h = \{(1, 1, \ldots, 1), (1, 1, \ldots, 2), \ldots, (C_H, C_H, \ldots, C_H)\}$$

Similarly, assuming $1 \leq \hat{C}_N[l] \leq C_N$, we denote a sequence of timed actions of a task $\mathcal{T}_N$ from the first up to the $\lceil \frac{T_L}{T_N} \rceil^{th}$ release as:

$$\hat{C}_N = (\hat{C}_N[1], \hat{C}_N[2], \ldots, \hat{C}_N[l], \ldots, \hat{C}_N[\left\lceil \frac{T_L}{T_N} \right\rceil])$$

[1] $l.c.m.$ (T_N, T_H, T_L) represents the least common multiple of the periods of tasks specified in the argument. It is often called the hyper-period.

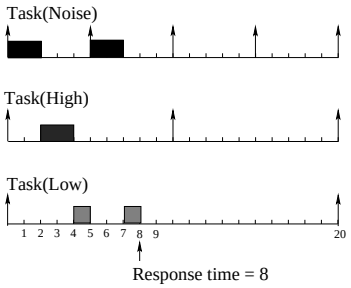

Figure 3. Timing diagram - Example 1

Let V_n be a set with all possible combinations of $\hat{C}_N$:

$$V_n = \{\hat{C}_N \mid \hat{C}_N[l] \in \{1, \ldots, C_N\}, l \in \{1, \ldots, \left\lceil \frac{T_L}{T_N} \right\rceil\}\}$$

Given that $\hat{C}_H \in V_h$, $\hat{C}_N \in V_n$, the response time of a task $\mathcal{T}_L$ is computed by adding up the sampling factor C_L, interference time caused by $\mathcal{T}_H$ and interference time caused by $\mathcal{T}_N$:

$$R_L = C_L + \sum_{k=1}^{\lceil \frac{R_L}{T_H} \rceil} \hat{C}_H[k] + \sum_{l=1}^{\lceil \frac{R_L}{T_N} \rceil} \hat{C}_N[l] \qquad (2)$$

We solve Eq. (2) for the response time R_L using a recurrent relationship [12].

As Eq. (2) indicates, not all the timed actions in $\hat{C}_N$ and $\hat{C}_H$ affect the response time R_L except when $R_L = T_L$. Let $*[i]$ or $*$ be a timed action (computation time) of $\mathcal{T}_H$ or $\mathcal{T}_N$ occurring after R_L, where i indicates the i^{th} period. Then, we let $\hat{C}_N(R_L)$ and $\hat{C}_H(R_L)$ be the notations which incorporate the notation $*[i]$ or $*$ into $\hat{C}_H$ and $\hat{C}_N$.

$$\hat{C}_H(R_L) = (\hat{C}_H[1], \hat{C}_H[2], \ldots, \hat{C}_H[\left\lceil \frac{R_L}{T_H} \right\rceil], *[k], \ldots, *[\left\lceil \frac{T_L}{T_H} \right\rceil]) \qquad (3)$$

$$\hat{C}_N(R_L) = (\hat{C}_N[1], \hat{C}_N[2], \ldots, \hat{C}_N[\left\lceil \frac{R_L}{T_N} \right\rceil], *[l], \ldots, *[\left\lceil \frac{T_L}{T_N} \right\rceil]) \qquad (4)$$

Finally, we can express the response time of a task $\mathcal{T}_L$ as a function f of $\hat{C}_H(R_L)$ $\hat{C}_N(R_L)$ and C_L (Figure 2):

$$R_L = f(\hat{C}_H(R_L), \hat{C}_N(R_L), C_L),\ \hat{C}_H(R_L) \in V_h, \hat{C}_N(R_L) \in V_n \qquad (5)$$

Example 1. Let $\Gamma_{RM} = \{\mathcal{T}_N(5, 2), \mathcal{T}_H(10, 2), \mathcal{T}_L(20, 2)\}$ and $C_L = 2$. If $\hat{C}_H = (2, 2)$ and $\hat{C}_N$=$(2, 2, 2, 2)$, then $R_L = 8$ (Eq. (2)). This can be represented via Eq. (5) as $R_L = 8 = f((2, *), (2, 2, *, *), 2)$. The timing diagram (also known as Gantt diagram) of this result is shown in Figure 3.

4 Communication Channel Model

Mathematically, one can view a channel as a probabilistic function that transforms a sequence of input symbols, $x \in$

X=$\{x_1, \ldots, x_k, \ldots, x_K, \}$, into a sequence of channel output symbols, $y \in Y$=$\{y_1, \ldots, y_j, \ldots, y_J\}$. We assume that the number of inputs and outputs of a channel are finite and the current output depends on only the current input. Such a channel is called a discrete memoryless channel (DMC).

Because of noise in a communication system, this transformation is typically not a one-to-one mapping from the set of input symbols X to the set of output symbols Y. Instead, any particular input symbol $x_k \in X$ may have some probability $P(y_j \mid x_k)$ of being transformed to the output symbol $y_j \in Y$. $P(y_j \mid x_k)$ is called a (forward) transition probability. Given a DMC, the probability distribution of the output set Y, denoted by Q_Y, can be calculated in matrix form as:

$$Q_Y = \begin{pmatrix} P(y_1) \\ P(y_2) \\ \vdots \\ P(y_J) \end{pmatrix} = \begin{pmatrix} P(y_1 \mid x_1) & \ldots & P(y_1 \mid x_K) \\ P(y_2 \mid x_1) & \ldots & \\ \vdots & \vdots & \vdots \\ P(y_J \mid x_1) & \ldots & P(y_J \mid x_K) \end{pmatrix} \begin{pmatrix} P(x_1) \\ P(x_2) \\ \vdots \\ P(x_K) \end{pmatrix} \quad (6)$$

Let $Q_{Y|X}$ be a matrix which has the transition probabilities of a noisy channel as its entities and Q_X represent the probability distribution of the input set X. Then, Eq. (6) is abbreviated as:

$$Q_Y = Q_{Y|X} Q_X$$

For notational convenience, $Q_Y(j)$ and $Q_X(k)$ represent the j^{th} and k^{th} entries of the column vectors Q_Y and Q_X. $Q_{Y|X}(j,k)$ or $Q_{j|k}$ represents the entry that lies in the j^{th} row and the k^{th} column of the matrix $Q_{Y|X}$.

According to Shannon's information theory, the entropy $H(X)$ is a measure of the information per symbol in a channel input set X and is defined as: $H(X) = \sum_{k=1}^{K} Q_X(k) \log_2 (1/Q_X(k))$. The average amount of the information transmitted overt a channel is defined in information theory as the mutual information $I(X;Y)$:

$$I(X:Y) = \sum_{k=1}^{K} \sum_{j=1}^{J} Q_X(k) Q_{j|k} \, log \frac{Q_{j|k}}{\sum_{i=1}^{K} Q_X(i) Q_{j|k}}$$

For a fixed transition probability matrix $Q_{Y|X}$, the mutual information $I(X;Y)$ is a function of the probability distribution Q_X of the set of input symbols X. The maximum mutual information achieved for a given transition probability matrix is the channel capacity C:

$$C = \max_{Q_X} I(X;Y) \quad (7)$$

Note that channel capacity (bits/symbol) is found by maximizing $I(X;Y)$ with respect to Q_X for a given transition probability matrix. We denote the Q_X which maximizes $I(X;Y)$ as Q_X^{max}. Generally, it is non-trivial to find Q_X^{max}. However, the Arimoto-Blahut algorithm [3] can be used to efficiently calculate Q_X^{max} and the channel capacity of a noisy channel if a transition probability matrix $Q_{Y|X}$ of the channel is provided to the algorithm. Thus, finding $Q_{Y|X}$ of a noisy channel is the most important step in evaluating the channel capacity.

In order to build a model for a covert timing path between $High$ and Low in RM based real-time systems, it is crucial to identify all possible input symbols available to $High$ and corresponding output symbols observed by Low. After this step, one must find a transition probability matrix of a covert timing channel to evaluate channel capacity. The following definition identifies the input and output symbols for our RM-based covert timing channel:

Definition 1. *A set $channel_{\Gamma_{RM}}$ is defined to be a binary relation between an input symbol being a sequence of timed actions $\hat{C}_H(R_L)$ of $\mathcal{T}_H$ and an output symbol being a corresponding response time R_L. Formally,*

$$channel_{\Gamma_{RM}} = \{(\hat{C}_H(R_L), R_L) \mid R_L = f(\hat{C}_H(R_L), \hat{C}_N(R_L), C_L), \hat{C}_H(R_L) \in V_h, \ \hat{C}_N(R_L) \in V_n\}$$

Two assumptions are needed to determine a transition probability of $channel_{\Gamma_{RM}}$: a statistical relationship between two real-time tasks $\mathcal{T}_H$ and $\mathcal{T}_N$, and a probability distribution of timed actions of $\mathcal{T}_N$. First, we assume that two tasks $\mathcal{T}_H$ and $\mathcal{T}_N$ are statistically *independent*, e.g., the timed actions of $\mathcal{T}_N$ are not known to $High$ nor under the control of $High$ (the timed actions of $\mathcal{T}_N$ are nondeterministic). This view is not the worst case assumption: our main interest is in the vulnerability of RM scheduling itself, not the correlation between $\mathcal{T}_H$ and $\mathcal{T}_L$.

Let R and $\vec{X}$ be the random variables representing the response time R_L and a sequence of timed actions $\hat{C}_H(R_L)$ of $\mathcal{T}_H$, respectively. Let $\vec{N}$ be a random variable representing a sequence of timed actions $\hat{C}_N(R_L)$ of $\mathcal{T}_N$. Provided that two tasks $\mathcal{T}_H$ and $\mathcal{T}_N$ are independent, we can compute a transition probability of $channel_{\Gamma_{RM}}$ as:

$$P(R = R_L \mid \vec{X} = \hat{C}_H(R_L)) = P(\vec{N} = \hat{C}_N(R_L) \mid R_L = f(\hat{C}_H(R_L), \hat{C}_N(R_L), C_L)) \quad (8)$$

A channel capacity is a (sensitive) function of a transition probability of a noisy channel (Eq. (7)) and the transition probability of $channel_{\Gamma_{RM}}$ depends on a probabilistic distribution of $\hat{C}_N(R_L)$ (Eq. (8)). Therefore, a choice of a probabilistic distribution for a timed action performed by $\mathcal{T}_N$ could (significantly) affect the capacity of $channel_{\Gamma_{RM}}$. In this paper, we assume that a timed action performed by $\mathcal{T}_N$ during each period follows the discrete uniform distribution and any two timed actions of $\mathcal{T}_N$ are statistically independent. This view may be most appropriate to our assumption that the timing behaviors of $\mathcal{T}_N$ are nondeterministic. Let N be a random variable indicating the computation time performed by $\mathcal{T}_N$ during each period. Assuming that $1 \leq \hat{C}_N[l] \leq C_N^{max}$ and the random variable N follows a discrete uniform distribution, we have the following probabilistic equation:

$$P(N = 1) = P(N = 2) \ldots = P(N = C_N) = \frac{1}{C_N} \quad (9)$$

Under our discrete uniform distribution and independence assumptions, $P(\vec{N}$=$(n_1, \ n_2, \cdots, n_k, *, *, \cdots, *))$

can be evaluated, assuming that $1 \leq n_k \leq C_N$:

$$\begin{aligned} & P(\vec{N} = (n_1, n_2, \cdots, n_k, *, *, \cdots, *)) \qquad (10) \\ &= P(N = n_1)P(N = n_2) \cdots P(N = n_k)P(N = *) \\ & \quad P(N = *) \cdots P(N = *) \\ &= \left(\frac{1}{C_N}\right)^k \end{aligned}$$

Note that $P(N = *) = 1$ since $P(N = *) = P(N = 1) + P(N = 2) + \cdots P(N = C_N)$. After the transition probability matrix $Q_{Y|X}$ of $channel_{\Gamma_{RM}}$ is constructed via Eq. (8), (9), and (10), the channel capacity can be found by using Eq. (7) or the Arimoto-Blahut algorithm.

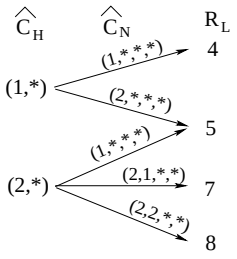

Figure 4. Covert communication channel

Example 2. Given that $\Gamma_{RM} = \{ \mathcal{T}_N(5,2), \mathcal{T}_H(10,2), \mathcal{T}_L(20,2) \}$ and $C_L = 2$, entering all possible combinations of $\hat{C}_H \in V_h$ and $\hat{C}_N \in V_n$ into Eq. (2), one can construct $channel_{\Gamma_{RM}} = \{((1,*),4),((1,*),5),((2,*),5),((2,*),7),((2,*),8)\}$ as shown in Figure 4. Using Eq. (8), (9), and (10), the transition probability of the channel can be calculated. For example, $P(R_L = 7 \mid \vec{X} = (2,*)) = P(\vec{N} = (2,1,*,*)) = \left(\frac{1}{2}\right)^2$. The transition probability matrix $Q_{Y|X}$ of the channel shown in Figure 4 is evaluated as:

$$Q_{Y|X} = \begin{pmatrix} 1/2 & 0 \\ 1/2 & 1/2 \\ 0 & 1/4 \\ 0 & 1/4 \end{pmatrix}$$

For the given $Q_{Y|X}$, the capacity of $channel_{\Gamma_{RM}}$ is 0.5 (bit/symbol) and $Q_X^{max} = (1/2 \;\; 1/2)^T$. Thus, $channel_{\Gamma_{RM}}$ achieves the maximum transmission rate when $P(\vec{X} = (1,*)) = P(\vec{X} = (2,*)) = \frac{1}{2}$.

4.1 Quantity of Specific Information

The channel capacity represents the maximum amount of information per symbol (bits/symbol) that can be transmitted through the given noisy channel on average. Security researchers commonly use the channel capacity or mutual information (if the probabilistic behavior of an input symbol is known and fixed) as a security metric to access the severities of a covert channel. These two quantities indicate the average amount information one obtains about the input symbols transmitted from observing output symbols. Thus, the channel capacity and mutual information are good indicators of overall performance of a communication channel.

However, in some cases, it is useful to know the amount of information gained from observing a specific output symbol received, rather than the average. For instance, an observer (Low) may deduce more about the state of a sender or the range of input symbols transmitted by observing some output symbols than others, i.e., some output symbols may be more informative about the range of input symbols transmitted than others.

Let us denote a set of input symbols as $X = \{x_1, \ldots, x_k, \ldots, x_K,\}$. Let y_j be an output symbol observed by a receiver. We write $I(X; y_j)$ to denote the amount of information carried by a particular output symbol y_j about the range of input symbols transmitted. We call $I(X; y_j)$ a quantity of specific information or a degree of deducibility associated with y_j. The formula[2] for the quantity of specific information [6] is:

$$I(X; y_j) = -\sum_{i=1}^{K} P(x_i) \log p(x_i) + \sum_{i=1}^{K} P(x_i \mid y_j) \log p(x_i \mid y_j) \qquad (11)$$

If a transition probability matrix $Q_{Y|X}$ and a probability distribution[3] of input symbols Q_X are known, $I(X; y_j)$ can be calculated since $P(x_i \mid y_j) = \frac{P(x_i)P(y_j|x_i)}{P(y_j)}$ and $P(y_j)$ is found via Eq (6).

The metric $I(X; y_j)$ may be very useful in identifying a system component which causes an output symbol with a higher degree of deducibility. Then one can effectively reduce a channel capacity by first working on a component which generates an output with the higher value of $I(X; y_j)$ rather than introducing arbitrary noise to a system.

Example 3. Assume that we have $channel_{\Gamma_{RM}}$ as shown in Figure 4. Let X be $\{x_1, x_2\} = \{(1,*),(2,*)\}$, and let y_1 and y_2 represent the response times with values of 4 and 5, respectively. Also assume that an input probabilistic distribution is chosen to achieve the maximum transmission rate, which means $Q_X = (1/2 \;\; 1/2)^T = Q_X^{max}$. Then, we can compare $I(X; y_1)$ and $I(X; y_2)$ to determine which output symbol carries more information about the range of input symbols. Using Eq. (11), $I(X; y_1 = 4) = 1$ and $I(X; y_2 = 5) = \frac{1}{2}$. This result indicates that the output symbol y_1 carries 1 bit of information about the range of the input symbols transmitted and y_2 carries 0 bits. Thus, upon receiving y_1, a receiver can deduce with a higher degree of certainty the range of input symbols transmitted than when receiving y_2.

Let us explain the previous example from an information-theoretical point of view. The value of $I(X; y_1)$ indicates that the amount of information generated at a sender side is transferred to a receiver side without

[2]There are also other formulas [2] for $I(X; y_j)$ but only Eq. (11) has an additive property [6]: $I(X; \{y_j, z_k\}) = I(X; y_j) + I(X; z_k \mid y_j)$, where y_j and z_k are two observations.

[3]When a probability of distribution Q_X of input symbols is not known, it is reasonable to assume $Q_X = Q_X^{max}$.

any loss since the quantity of specific information associated with y_j is equal to the entropy of the input symbol set X, i.e., $I(X; y_1) = H(X) = 1$. On the other hand, $I(X; y_2) = 0$ indicates the amount of information generated at a sender side is all lost during the transmission. The output symbol like y_1 is most informative and the output symbol like y_2 is least informative to a receiver. We can generalize our observation:

Proposition 1. *If $I(X; y_j) = H(X)$, upon receiving y_j, an observer (Low) can deduce exactly which input symbol has been transmitted by a sender ($High$). On the other hand, if $I(X; y_j) = 0$, upon receiving y_j, an observer cannot deduce anything about the range of input symbols transmitted.*

Proof. Let $X = \{x_1, \cdots, x_K\}$ be a set of input symbols. $I(X; y_j) = H(X)$ implies that $\sum_{i=1}^{K} P(x_i \mid y_j) \log p(x_i \mid y_j) = 0$. Let $E(y_j) = \sum_{i=1}^{K} P(x_i \mid y_j) \log p(x_i \mid y_j)$ ($E(y_j)$ stands for the equivocation associated with y_j). In order for $E(y_j)$ to be 0, there must exist a one-to-one mapping from an input symbol to the output symbol y_j. Because of the existence of the one-to-one mapping, Low can pin-point the input symbol transmitted which results in y_j. If $I(X; y_j) = 0$, then $E(y_j) = H(X)$. This indicates the statistical independence between the input symbols and the output symbol y_j since $P(x_i \mid y_j) = P(x_i)$ for all input symbols $x_i \in X$. Thus, an observer cannot deduce anything about the range of input symbols transmitted.

5 Conclusion

Under the assumption that $\mathcal{T}_L$ is a task with the longest period, we provide a mathematical framework for computing the capacity of a covert timing channel between $\mathcal{T}_L$ and $\mathcal{T}_H$ while a third party task is running concurrently with them under RM scheduling. Another metric called the quantity of specific information is used to quantify the amount of information carried by a specific output symbol. The two quantities are both useful for different reasons: channel capacity can serve as a metric to measure the overall severity of a covert channel. Meanwhile, the quantity of specific information can be used to find out which output symbol is more informative to Low.

Acknowledgements

This material is based on research sponsored by AFRL and DARPA under agreement number F30602-02-1-0178. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expresses or implied, of AFLR and DARPA or the U.S. Government.

References

[1] N.C. Audsly, A. Burns, M.F. Richardson, K. Tindell, and A.J. Wellings. Applying New Scheduling Theory to Static Priority Pre-emptive Scheduling. *Software Engineering Journal*, 8(5):284–292, 1993.

[2] M. Bezzi. Quantifying the information transmitted in a single stimulus. *arXiv:q-bio.NC/0601038 v1*, Jan 2006.

[3] R. E. Blahut. Computation of channel capacity and rate-distortion functions. *IEEE Trans. on Inform. Theory*, IT-18:460–473, July 1972.

[4] S. Cabuk, C. Brodley, and C. Shields. IP Covert Timing Channels: Design and Detection. In *Proc. ACM conference on Computer and Communications Security*, 2004.

[5] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley-Interscience, 1991.

[6] M. R. DeWeese and M. Meister. How to measure the information gained from one symbol. *Network: Comput. Neural Syst*, 10:325–340, 1999.

[7] M. Joseph and P. Pandya. Finding Response Time in a Real-Time System. *The Computer Journal*, 29(5):390–395, 1986.

[8] M. H. Kang and I. Moskowitz. A pump for rapid, reliable, secure communication. In *Proc. the first ACM Conference on Computer and Communication Security*, 1993.

[9] C. Liu and J. Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of the ACM*, 20(1), 1973.

[10] J. K. Millen. Covert Channel Capacity. In *Proc. IEEE Symposium on Security and Privacy*, pages 60–66, 1987.

[11] I. Moskowitz and M. H. Kang. Covert Channels - Here to Stay. In *Proc. COMPASS 94*, pages 235–243, 1994.

[12] J. Son and J. Alves-Foss. Covert Timing Channel Analysis of Rate Monotonic Real-Time Scheduling Algorithm in MLS Systems. In *Proc. IEEE Workshop on Information Assurance*, pages 361–368, 2006.

[13] S. Son, R. Mukkamala, and R. David. Integrating security and real-time requirements using covert channel capacity. *IEEE Trans. Knowledge and Data Eng.*, 12(6):865–879, 2000.

[14] W. Stallings. *Operating Systems: Internals and Design Principles*. Prentice Hall, fifth edition, July 2004.