A STUDY OF COVERT QUEUEING CHANNELS IN SHARED SCHEDULERS

BY

AMIREMAD GHASSAMI

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2016

Urbana, Illinois

Adviser:

Associate Professor Negar Kiyavash

ABSTRACT

We study covert queueing channels (CQCs), which are a kind of covert timing channel that may be exploited in shared queues across supposedly isolated users. In our system model, a user modulates messages to another user via his pattern of access to the shared resource. One example of such a channel is the cross-virtual network covert channel in data center networks resulting from the queueing effects of the shared resource.

First, we study a system comprising a transmitter and a receiver that share a deterministic and work-conserving first-come-first-served scheduler, and we compute the maximum reliable data transmission rate, i.e., the capacity, of this channel. Next, we extend the model to include a third user who also uses the shared resource and study the effect of the presence of this user on the information transmission rate. The solution approach presented in this extension may be applied to calculate the capacity of the covert queueing channel among any number of users.

We also study a queueing covert channel between two users sharing a round robin scheduler. Such a covert channel can arise when users share a resource such as a computer processor or a router arbitrated by a round robin policy. We present an information-theoretic framework to model and derive the capacity of this channel for both noiseless and noisy scenarios. Our results show that seemingly isolated users can communicate at a high rate over the covert channel. Furthermore, we propose a practical finite-length code construction, which achieves the capacity limit.

To my parents and my sister, for their love and support.

ACKNOWLEDGMENTS

I would like to thank my adviser, Professor Negar Kiyavash, for her guidance while I carried out this work. I would also like to thank Professor Yihong Wu, who formed my perspective towards information theory. Many of the ideas in this thesis originated in Professor Wu's amazing information theory class.

I would like to thank my friends and my colleagues at the Coordinated Science Laboratory for the excellent research environment. Especially, I would like to thank Xun Gong for all his help in the early stage of the project, Ali Yekkehkhany for his help and contributions in Chapter 4 of this thesis, and Daniel Cullina for his guidance and several illuminating discussions that we had. Finally, I would like to thank my parents and my sister for their love and support.

TABLE OF CONTENTS

LIST OF FIGURES

PUBLICATIONS

- A. Ghassami, X. Gong and N. Kiyavash, "Capacity limit of queueing timing channel in shared FCFS schedulers," in *2015 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2015, pp. 789-793.
- R. Tahir, M. T. Khan, X. Gong, A. Ahmed, A. Ghassami, H. Kazmi, M. Caesar, F. Zaffar, and N. Kiyavash, "Sneak-Peek: High Speed Covert Channels in Data Center Networks," in *IEEE International Conference on Computer Communications (INFOCOM)*. 2016.
- A. Ghassami and N. Kiyavash "A Covert Queueing Channel in FCFS Schedulers," submitted.
- A. Ghassami, A. Yekkehkhany, N. Kiyavash and Y. Lu, "A Covert Queueing Channel in Round Robin Schedulers," submitted.

CHAPTER 1

INTRODUCTION

The existence of side and covert channels due to the fragility of isolation mechanisms is an important privacy and security threat in computer networks. Such channels may be created across users which were supposed to be isolated, resulting in information leakage. By definition, a covert channel is a hidden communication channel which is not intended to exist in the system and is created furtively by users [1]. Covert channels may be exploited by a trusted user, or possibly a malware inside a system with access to secret information to leak it to a distrusted user. On the other hand, in a side channel a malicious user attempts to learn private information by observing information not intended for him. In this scenario, there is no collaboration between the source of information and the recipient [2].

A special case of covert and side channels is a timing channel in which information is conveyed through timing of occurrence of events (e.g., inter-arrival times of packets). For instance, queueing covert/side timing channels may arise between users who share a packet scheduler in a network.

Packet schedulers serve packets from multiple streams which are queued in a single queue. This causes dependencies between delays observed by users. Particularly, the delay that one user experiences depends on the amount of traffic generated by other streams, as well as his own traffic. Hence, a user can gain information about other users' traffic by observing delays of his own stream. This dependency between the streams can breach private information as well as create hidden communication channels between the users.

One example of a covert/side queueing channel is the cross-virtual network covert channel in data center networks and cloud environments. In recent years, migrating to commercial clouds and data centers is becoming increasingly popular among companies that deal with data. The multi-tenant nature of cloud and sharing infrastructure between several users has made

data protection and avoiding information leakage a serious challenge in such environments [3]. In data center networks, software-defined-networks are frequently used for load balancing [4]. This generates logically isolated virtual networks and prevents direct data exchange. However, since packet flows belonging to different VNs inevitably share underlying network infrastructure (such as a router or a physical link), it is possible to transfer data across VNs through timing channels resulting from the queueing effects of the shared resource(s).

In this thesis, we focus on the covert queueing channels (CQCs) that could be created in shared schedulers. Different schedulers such as first-come-first-served (FCFS), time division multiple access (TDMA), round robin, etc., can be used for resource sharing. Clearly the optimal scheme for message transmission and the rate of communication between the users in a CQC depend on the scheduling policy of the shared resource.

The utility of a scheduler is predominantly evaluated by its throughput. As long as the rates at which users request the shared resource is within the scheduler's capacity region, an effective scheduler should be able to respond to the users' requests in a stable fashion. Such a scheduler is called a throughput optimal scheduler.

It has been shown that, from the security viewpoint, TDMA is the most secure scheduling policy [5]. In this type of scheduler, since the serving times of the users are decoupled in time, users' delays are independent of each other, and hence no information can be conveyed through the scheduler. However, the decoupling can cause significant delays in service given to users. For instance, a user has to wait despite the other user having no jobs to be served. Hence, TDMA is not throughput optimal.

In this thesis, we study CQCs in a shared deterministic and work-conserving scheduler. We present an information-theoretic framework to describe and model the data transmission in this channel and calculate its capacity.

First, in Chapter 2, we consider the setting with two users sharing an FCFS scheduler. We will show that although this scheduler does not waste any resource and hence is throughput optimal, it allows users to communicate with an information rate as high as 0.8114 bits per time slot. We also study the effect of the presence of a third user on the information transmission rate, presented in Chapter 3. The approach for analyzing the effect of the presence of the third user may be extended to calculate the capacity of the

queueing covert channel serving any number of users.

In Chapter 4, we focus on a round robin scheduler, which is another throughput optimal policy commonly used in computer processors and communication networks. We show that users can communicate with an information rate of 0.6942 bits per time slot through the covert channel created between them in this system in the absence of noise. Additionally, we study the noisy version of this covert channel in which packets are dropped with a certain probability, and compute the capacity as a function of packet drop probability.

1.1 Related Works

The existing literature on covert/side timing channels has mainly concentrated on timing channels in which the receiver/adversary has direct access to the timing sequence produced by the transmitter/victim or a noisy version of it. However, in a covert/side queueing channel, the receiver/adversary does the inference based on the timing of his own packets which has been influenced by the original stream.

In a queuing side channel, where a malicious user, called an attacker, attempts to learn another user's private information, the main approach used by the attacker is traffic analysis. That is, the attacker tries to infer private information from the victim's traffic pattern. The attacker can have an estimation of the features of the other user's stream such as packet size and timing by emitting frequent packets in his own sequence. Previous work shows that through traffic analysis, the attacker can obtain various private information including visited web sites [6], sent keystrokes [7], and even inferring spoken phrases in a voice-over-IP connection [8].

In [9], Gong et al. proposed an attack where a remote attacker learns about a legitimate user's browser activity by sampling the queue sizes in the downstream buffer of the user's DSL link. The information leakage of a queueing side channel in an FCFS scheduler is analyzed in [10]. The analysis of more general work-conserving policies has been done in [11] and [5]. The authors in [5] present an analytical framework for modeling information leakage in queuing side channels and quantify the leakage for several common scheduling policies. Kadloor and Kiyavash [12] showed that when

dealing with queueing side channels, round robin scheduling is privacy optimal within the class of work-conserving policies. In Chapter 4 of this thesis, we focus on covert channels created in this scheduler.

Most of the work in covert timing channels is devoted to the case in which two users communicate by modulating the timings, and the receiver sees a noisy version of the transmitter's inputs [13–17]. Also, there are many works devoted to the detection of such channels [14, 18, 19]. The setup of CQC is new in the field of covert communication and as far as we are aware, there are very few works on this setup [20, 21].

CHAPTER 2

FCFS SCHEDULER

In this chapter, we consider the setting with two users sharing a first-come-first-served (FCFS) scheduler. We will show that the supposedly isolated users of an FCFS scheduler are capable to communicate with an information rate as high as 0.8114 bits per time slot.

2.1 System Description

Consider the architecture depicted in Figure 2.1. In this model, a scheduler serves packets from 2 users: U_e and U_d. Each user, U_i, $i \in \{e, d\}$, is modeled by a transmitter and a receiver node, denoted by U_i^T and U_i^R, respectively. U_i^R is the node which receives U_i^T's packet stream. Note that U_i^T and U_i^R could correspond to the uplink and downlink of the same entity. U_e intends to send a message to U_d, but there is no direct channel between them. However, since U_e^T and U_d^T's packets share the same queue, U_e^T can encode messages in the arrival times of its packets, which are passed onto U_d via queueing delays. Therefore, a timing channel is created between users via the delays experienced through the coupling of their traffic due to the shared scheduler.

To receive the messages from U_e, user U_d sends a packet stream from the node U_d^T. He then uses the delays he experiences by receiving the packet stream at U_d^R to decode the message. Therefore, effectively, the nodes U_e^T and U_e^R are on the encoder side and the nodes U_d^T and U_d^R are on the decoder side of the channel of our interest. In the sequel, we call U_d's sent stream the *probe stream*.

We consider an FCFS scheduler, which is commonly used in DSL routers. We assume this scheduler is deterministic and work conserving. Time is discretized into slots, and the scheduler is capable of processing at most one packet per time slot. At each time slot, each user either issues one packet or

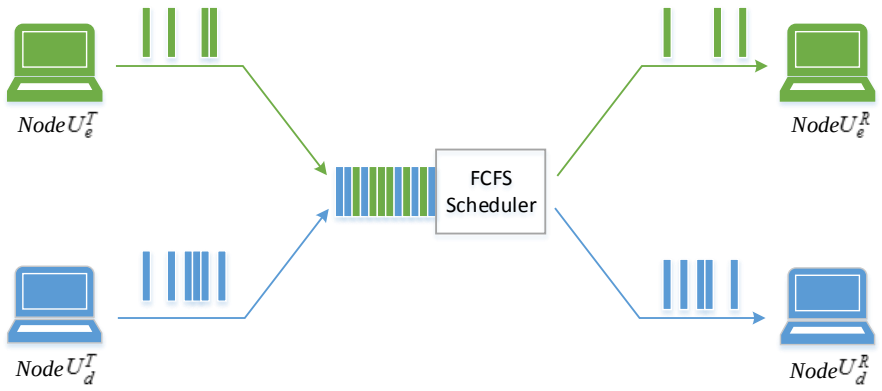

Figure 2.1: Covert queueing channel in a system with 2 users.

remains idle. Furthermore, we assume that all packets are the same size.

Figure 2.2 shows an example of the input and output streams of the system depicted in Figure 2.1 with an FCFS scheduler. In this figure, the first stream is the arrival stream, i.e., arrivals from both U_e^T and U_d^T, depicted by red and blue, respectively. The second part is the output stream of user U_e (received by U_e^R), and the third part is the output stream of user U_d (received by U_d^R). In this example, we assume that one packet is buffered in the queue at time A_i, where a packet arrives from both U_d^T and U_e^T. If user U_e had not sent the two packets (depicted in red), the second packet of user U_d which arrives at time A_{i+1} could have departed one time slot earlier. Therefore, U_d knows that U_e has issued two packets.

We assume that the priorities of the users are known. Particularly, without loss of generality, we assume that U_d has the highest priority; i.e., in the case of simultaneous arrivals, U_d's packet will be served first.

As mentioned earlier, at each time slot, each user is allowed to either send one packet or none; hence, the input and output packet sequences of each user could be viewed as a binary bitstream, where '1' and '0' indicates whether a packet was sent or not in the corresponding time slot.

Assume message W drawn uniformly from the message set $\{1, 2, ..., M\}$ is transmitted by U_e^T, and $\hat{W}$ is U_d's estimate of the sent message. Our performance metric is the average error probability, defined as follows:

$$P_e \triangleq Pr(W \neq \hat{W}) = \sum_{m=1}^{M} \frac{1}{M} Pr(\hat{W} \neq m | W = m).$$

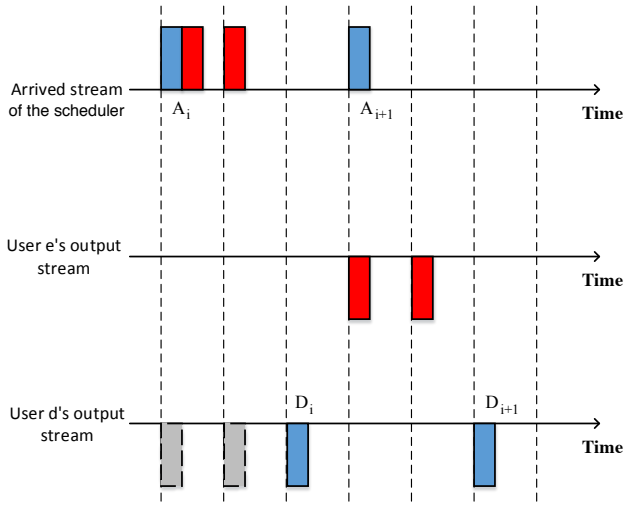

Figure 2.2: An example of the input and output streams of the FCFS scheduler serving two users. Red packets belong to U_e and blue packets belong to U_d. We assume that one packet is buffered in the queue at time A_i.

U_e encodes each message into a binary sequence of length n, Δ^n, to create the codebook, which is known at the decoder, U_d.

In order to send a message, U_e^T emits a packet in the i^{th} time slot if $\Delta_i = 1$ and remains idle otherwise, i.e.,

$$\Delta_i = \begin{cases} 1 & \Rightarrow U_e^T \text{issues a} \\ & \text{packet in time slot } i. \\ 0 & \Rightarrow U_e^T \text{remains} \\ & \text{idle in time slot } i. \end{cases}$$

To decode this message, U_d^T sends a binary length n stream (the probe stream) to the scheduler during the same length n time period. User U_d will use this stream and the response stream received at node U_d^R to decode the sent message.

We define the code, rate of the code, and the channel capacity similar to the definitions in [13], [22] and [23], as follows:

Definition 1 *An (n, M, ϵ)-code consists of a codebook of M equiprobable binary codewords, where messages take on average n time slots to be received. The error probability satisfies $P_e \leq \epsilon$.*

Definition 2 *The information transmission rate of a code, R, is the amount of conveyed information (logarithm of the codebook size) normalized by the average number of used time slots for the message to be received, i.e.,*

$$R = \frac{\log M}{n}.$$

Rate may be interpreted as the average amount of information conveyed in a single time slot.

Definition 3 ***(Channel Capacity)*** *The Shannon capacity, C, for a channel is the maximum achievable rate at which one can communicate through the channel when the average probability of error goes to zero. In other words, C is the supremum of rates, R, which satisfies the following property [23]:*

$$\forall \delta > 0, \ \exists (n, M, \epsilon_n)\text{-}code \quad s.t. \quad \begin{cases} \frac{\log M}{n} > R - \delta \\ \epsilon_n \to 0 \end{cases} \quad as\ n \to \infty.$$

The following notations will be used in Chapters 2 and 3:

- r_i: U_i^T's packet rate.
- A_i: Arrival time of the i^{th} packet in the probe stream.
- D_i: Departure time of the i^{th} packet of the probe stream.
  We assume m packets are sent by U_d during n time slots and we have: $r_d = \lim_{n\to\infty} \frac{m}{n}$.
- X_i: Number of U_e's packets sent in the interval $[A_i, A_{i+1})$. Note that $X_i = \sum_{j=A_i}^{A_{i+1}-1} \Delta_j$.
- $T_i = A_{i+1} - A_i$: inter-arrival time between i^{th} and $(i+1)^{th}$ packet of the probe stream. We denote a realization of T by τ.
- $Y_i = D_{i+1} - D_i - 1$.
- $\hat{X}_i$: estimate of X_i by decoder.
- $\hat{W}$: decoded message.

In an FCFS scheduler, U_d can have an estimation of the number of the packets of other users between any of his own consecutive packets. The estimation of the number of packets in the interval $[A_i, A_{i+1})$ is accurate if the scheduler is deterministic and work-conserving and a sufficient number of packets is buffered in the queue at time A_i[1]. In that case, the number of other users' packets arriving in the interval $[A_i, A_{i+1})$ could be simply calculated by $D_{i+1} - D_i - 1$. Note that U_d cannot pinpoint the location of the sent packets; that is, if the inter-arrival time is τ, U_d can distinguish between $\tau + 1$ different sets of bit streams sent during this time. Therefore, we look at any probe stream sent during n time slots as a combination of different inter-arrival times.

If the sum of the packet rates of the users during sending a message of length n is on average larger than 1, then the message will arrive on average during more than n time slots. Also, this will destabilize the input queue of the scheduler. For example, for a system with two users U_d and U_e, if U_d^T sends packets in every time slot, then sending a packet by U_e^T in any time slot would cause a delay in the serving of the next packet of U_d^T and hence could be detected. Therefore, in each time slot, U_e^T could simply idle to signal a bit '0' or send a packet to signal a bit '1', resulting in the information rate of $\frac{1}{1.5}$ bit per time slot in the case that bits are equiprobable. But, this scheme is not feasible in practice as it would destabilize the queue and result in severe packet drops.

In order to have queue stability, it suffices that the total packet arrival rate does not exceed the service rate, which for a deterministic and work-conserving scheduler is equal to 1 (see Appendix A.1 for the proof of stability which is based on a Lyapunov stability argument for the general case that the serving rate is assumed to be $0 \leq \rho \leq 1$ and arbitrary number of users is

[1]If the service rate of the scheduler is equal to 1, there should be at least $A_{i+1} - A_i - 1$ packets buffered in the queue at time A_i. Therefore, user U_d needs to know the queue length. This is feasible using the following formula:

$$q(A_i) = D_i - A_i - 1,$$

where $q(A_i)$ denotes the queue length at the time that the i^{th} packet in the probe stream arrives at the queue. The extra 1 in the formula is the time needed for the i^{th} packet of the probe stream to be served. Therefore, user U_d should always be aware of the queue length and keep it sufficiently large by sending extra packets when needed.

considered). Specifically, for the case of two users we need:

$$r_e + r_d < 1. \tag{2.1}$$

On the other hand, if the sum of the packet rates of the users used during sending a message of length n is on average less than 1, the length of the input queue may go to zero, and consequently U_d may not be able to count the number of packets of other users correctly. Note that increasing r_d increases the resolution available for user U_d and hence this user can have a better estimation of the number of the other user's sent packets; therefore, in the case of two users, in order to achieve the highest information rate, the operation point should tend to the line $r_e + r_d = 1$.

Therefore, we focus on the coding schemes where the sum of the rates is held at 1, with considering a preamble stage in our communication to guarantee sufficient queue length.

In the following section, the capacity of the introduced system will be calculated for a system with a deterministic and work-conserving FCFS scheduler serving packets from 2 users.

2.2 Channel Coding Theorem

In this section, using achievability and converse arguments, the capacity of the introduced system is calculated for the basic case that the scheduler has serving rate 1 and serves packets from two users.

As depicted in figure 2.1, user U_e is attempting to send a message to U_d through the covert queueing channel between them. Note that since we have considered service rate of 1 for the FCFS scheduler and users can agree on the packet stream sent by U_d^T ahead of time, the feedback U_e^R is already available at the encoder. Therefore, the following Markov chain holds:

$$W \to X^m \to Y^m \to \hat{W}. \tag{2.2}$$

Note that as mentioned earlier, if there is a sufficient number of packets buffered in the shared queue, $\hat{X}_i$ could be accurately estimated as Y_i.

The main result of this section is the following theorem, the proof of which is developed in the rest of the section.

Theorem 1 *The capacity of the timing channel in a shared FCFS scheduler with service rate 1 depicted in Figure 2.1 is equal to* 0.8114 *bits per time slot, which can be obtained by solving the following optimization problem:*

$$\begin{aligned} C = & \sup_{\alpha,\gamma_1,\gamma_2} \alpha\tilde{H}(\gamma_1, 1) + (1-\alpha)\tilde{H}(\gamma_2, \frac{1}{2}) \\ s.t. & \\ & \alpha(\gamma_1 + 1) + (1-\alpha)(\gamma_2 + \frac{1}{2}) = 1, \end{aligned} \tag{2.3}$$

where $0 \leq \alpha \leq 1$ *and* $0 \leq \gamma_1, \gamma_2 \leq \frac{1}{2}$ *and the function* $\tilde{H} : [0,1] \times \{\frac{1}{k} : k \in \mathbb{N}\} \mapsto [0,1]$ *is defined as:*

$$\tilde{H}(\gamma, \frac{1}{k}) = \frac{1}{k} \sup_{\substack{X \in \{0,1,\ldots,k\} \\ \mathbb{E}[X] = k\gamma}} H(X), \quad k \in \mathbb{N}, 0 \leq \gamma \leq 1. \tag{2.4}$$

We first investigate some of the properties of the function $\tilde{H}$.

Lemma 1 *Let* $U_k \sim unif(\{0,1,...,k\})$*. The distribution which achieves the optimum value in (2.4) is the tilted version of* $unif(\{0,1,...,k\})$ *with parameter* λ*, where* $\lambda = (\psi'_{U_k})^{-1}(k\gamma)$*, where the function* $\psi'_{U_k}(\cdot)$ *is the derivative of the log-moment generating function of* U_k*.*

See Appendix A.2 for the proof of Lemma 1.

Lemma 2 *The function* $(\gamma, \frac{1}{k}) \mapsto \tilde{H}(\gamma, \frac{1}{k})$ *could be computed using the following expression:*

$$\tilde{H}(\gamma, \frac{1}{k}) = \frac{1}{k}[\log_2(k+1) - \psi^*_{U_k}(k\gamma)\log_2 e], \tag{2.5}$$

where $U_k \sim unif(\{0,1,...,k\})$*, and the function* $\psi^*_{U_k}(\cdot)$ *is the rate function given by the Legendre-Fenchel transform of the log-moment generating function,* $\psi_{U_k}(\cdot)$*:*

$$\psi^*_{U_k}(\gamma) = \sup_{\lambda \in \mathbb{R}} \{\lambda\gamma - \psi_{U_k}(\lambda)\}. \tag{2.6}$$

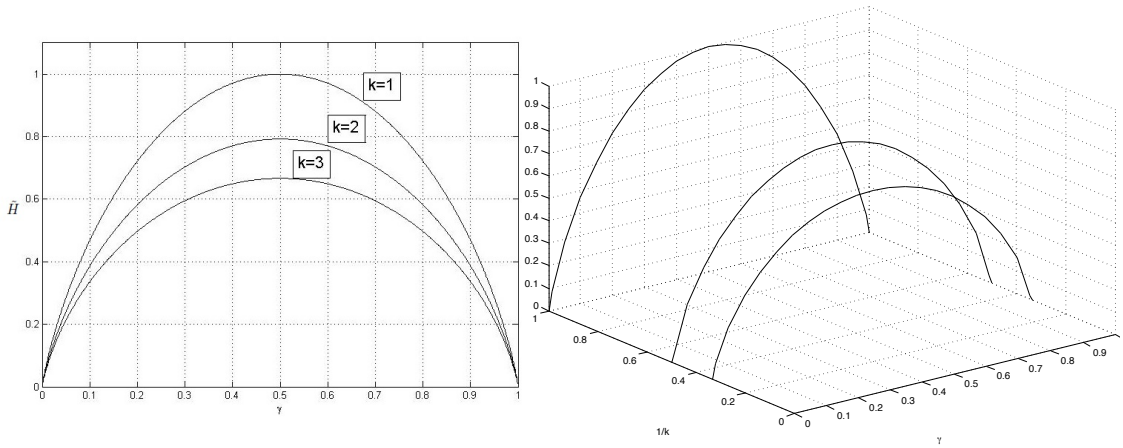

Figure 2.3: $\tilde{H}(\gamma, \frac{1}{k})$ for different values of γ and $k \in \{1, 2, 3\}$.

In order to prove this lemma, first we note that for any random variable X defined over the set $\{0, 1, ..., k\}$,

$$\begin{aligned} H(X) &= \sum_{i=0}^{k} P_X(i) \log \frac{1}{P_X(i)} \\ &= \sum_{i=0}^{k} P_X(i) \log (k+1) - \sum_{i=0}^{k} P_X(i) \log \frac{P_X(i)}{\frac{1}{k+1}} \\ &= \log (k+1) - D(P_X || U_k), \end{aligned}$$

where $D(P_X||U_k)$ denotes the KL-divergence between P_X and U_k. Therefore, in order to maximize $H(X)$, we need to minimize $D(P_X||U_k)$. Using the following well-known fact concludes the lemma [24]:

$$\min_{\mathbb{E}[X]=k\gamma} D(P_X||U_k) = \psi^*_{U_k}(k\gamma) \log_2 e. \tag{2.7}$$

Figure 2.3 shows the function $\tilde{H}(\gamma, \frac{1}{k})$ for different values of γ and $k \in \{1, 2, 3\}$.

Substituting (2.5) in (2.3) and solving it, the capacity of the timing channel in the shared FCFS scheduler with service rate 1 depicted in Figure 2.1 is equal to 0.8114 bits per time slot, achieved by $\alpha = 0.177$, $\gamma_1 = 0.43$ and $\gamma_2 = 0.407$.

Lemma 3 *The function $\tilde{H}(\cdot, \cdot)$ is concave in pair $(\gamma, \frac{1}{k})$ in the sense that for integers k_1, k_2, k_3, and for values $0 \leq \gamma_1, \gamma_2, \gamma_3 \leq 1$, and for $\alpha \in [0, 1]$*

such that $\alpha(\gamma_1, \frac{1}{k_1}) + (1-\alpha)(\gamma_3, \frac{1}{k_3}) = (\gamma_2, \frac{1}{k_2})$*, we have:*

$$\alpha\tilde{H}(\gamma_1, \frac{1}{k_1}) + (1-\alpha)\tilde{H}(\gamma_3, \frac{1}{k_3}) \leq \tilde{H}(\gamma_2, \frac{1}{k_2}). \tag{2.8}$$

See Appendix A.3 for the proof of Lemma 3.

Lemma 4 $\tilde{H}(\gamma, \frac{1}{k}) = \tilde{H}(1-\gamma, \frac{1}{k})$.

See Appendix A.4 for the proof of Lemma 4.

In the following, the proof of Theorem 1 is given. The proof is based on converse and achievability arguments.

2.2.1 Converse side

In the converse side, the ultimate goal is to prove that

$$C \leq \sup_{\alpha,\gamma_1,\gamma_2} \alpha\tilde{H}(\gamma_1, 1) + (1-\alpha)\tilde{H}(\gamma_2, \frac{1}{2})$$

s.t.

$$\alpha(\gamma_1 + 1) + (1-\alpha)(\gamma_2 + \frac{1}{2}) = 1,$$

where $0 \leq \alpha \leq 1$ and $0 \leq \gamma_1, \gamma_2 \leq \frac{1}{2}$. We break the proof in two lemmas.

Lemma 5 *For any* (n, M, ϵ)*-code we have*

$$\frac{1}{n}\log M \leq \sum_{\tau=1}^{n} [\pi_\tau \tilde{H}(\mu_\tau, \frac{1}{\tau})] + \epsilon_n \tag{2.9}$$

such that

$$\sum_{\tau=1}^{n} \pi_\tau(\mu_\tau + \frac{1}{\tau}) = 1 \quad \textit{and} \quad 0 \leq \mu_\tau \leq \frac{1}{2}, \quad \forall\tau, \tag{2.10}$$

where $\epsilon_n = \frac{1}{n}(H(P_e) + P_e \log_2(M-1))$*,* π_τ *is the portion of time that user* U_d *sends packets with inter-arrival time equal to* τ *in the probe stream, and* μ_τ *is* U_e^T*'s average packet rate when the inter-arrival time is equal to* τ*.*

Proof. We first we note that for any (n, M, ϵ)-code we have:

$$\begin{aligned}
\frac{1}{n}\log M &\overset{(a)}{=} \frac{1}{n}H(W) \\
&\overset{(b)}{=} \frac{1}{n}H(W|\tau^m) \\
&= \frac{1}{n}I(W;\hat{W}|\tau^m) + \frac{1}{n}H(W|\hat{W},\tau^m) \\
&\overset{(c)}{\leq} \frac{1}{n}I(W;\hat{W}|\tau^m) + \epsilon_n \\
&\overset{(d)}{\leq} \frac{1}{n}I(X^m;Y^m|\tau^m) + \epsilon_n,
\end{aligned}$$

where (a) holds because W is a uniform random variable over the set of messages $\{1, ..., M\}$, (b) follows from the fact that the chosen message is independent of the inter-arrival time of decoder's packets, (c) follows from Fano's inequality with $\epsilon_n = \frac{1}{n}(H(P_e) + P_e \log_2(M-1))$, and (d) follows from data processing inequality in Markov chain in (2.2). Therefore:

$$\begin{aligned}
\frac{1}{n}\log M &\leq \frac{1}{n}[H(X^m|\tau^m) - H(X^m|Y^m,\tau^m)] + \epsilon_n \\
&\leq \frac{1}{n}H(X^m|\tau^m) + \epsilon_n \\
&\leq \frac{1}{n}\sum_{j=1}^{m} H(X_j|\tau^m) + \epsilon_n \\
&\leq \frac{1}{n}\sum_{j=1}^{m} \max_{P_{X_j|\tau^m}} H(X_j|\tau^m) + \epsilon_n,
\end{aligned}$$

where, in the maximization above, the mean of the distribution $P_{X_j|\tau^m}$ is $\mathbb{E}[X_j|\tau^m]$. As mentioned earlier, in order to find the maximum information rate while having stability, we are interested in the asymptotic regime in which the operating point is converging to the line $r_e + r_d = 1$. Therefore, the information rate is upper bounded by having the set of means, $\{\mathbb{E}[X_1|\tau^m], \mathbb{E}[X_2|\tau^m], ..., \mathbb{E}[X_m|\tau^m]\}$, satisfying the constraint $\frac{1}{n}\sum_{j=1}^{m}\mathbb{E}[X_j|\tau^m] + r_d = 1$. Let $\xi_j = \frac{\mathbb{E}[X_j|\tau^m]}{\tau_j}$. Using (2.4), we have:

$$\max_{\substack{P_{X_j|\tau^m} \\ \mathbb{E}[X_j|\tau^m]=\tau_j\xi_j}} H(X_j|\tau^m) = \tau_j \tilde{H}(\xi_j, \frac{1}{\tau_j}), \tag{2.11}$$

where as mentioned in Lemma 1, the distribution for each X_j which achieves the maximum value in (2.11) is the tilted distribution of U_{τ_j} with parameter λ, such that $\lambda = (\psi'_{U_{\tau_j}})^{-1}(\tau_j \xi_j)$.

Therefore, we will have:

$$\frac{1}{n} \log M \leq \frac{1}{n} \sum_{j=1}^{m} \tau_j \tilde{H}(\xi_j, \frac{1}{\tau_j}) + \epsilon_n,$$

such that the set $\{\xi_1, \xi_2, ..., \xi_m\}$ satisfies the constraint $\frac{1}{n} \sum_{j=1}^{m} \tau_j \xi_j + r_d = 1$.

The inter-arrival times take values in the set $\{1, 2, ..., n\}$. Therefore, in the summation above we can fix the value of inter-arrival time on the value τ and count the number of times that τ_j has that value. Defining m_τ as the number of times that the inter-arrival time is equal to τ (note that $n = \sum_{\tau=1}^{n} \tau \cdot m_\tau$), we can break the summation above as follows:

$$\begin{aligned}
\frac{1}{n} \log M &\leq \frac{1}{n} \sum_{\tau=1}^{n} [\sum_{k=1}^{m_\tau} \tau \tilde{H}(\mu_{\tau,k}, \frac{1}{\tau})] + \epsilon_n \\
&= \frac{1}{n} \sum_{\tau=1}^{n} [\tau \sum_{k=1}^{m_\tau} \tilde{H}(\mu_{\tau,k}, \frac{1}{\tau})] + \epsilon_n \\
&= \frac{1}{n} \sum_{\tau=1}^{n} [\tau \cdot m_\tau \sum_{k=1}^{m_\tau} \frac{1}{m_\tau} \tilde{H}(\mu_{\tau,k}, \frac{1}{\tau})] + \epsilon_n,
\end{aligned}$$

where $\mu_{\tau,k}$ is equal to the k^{th} ξ_j which has $\tau_j = \tau$.

By Lemma 3, the function $\tilde{H}(\cdot, \cdot)$ is a concave function of its first argument. Therefore, by Jensen's inequality, we will have:

$$\sum_{k=1}^{m_\tau} \frac{1}{m_\tau} \tilde{H}(\mu_{\tau,k}, \frac{1}{\tau}) \leq \tilde{H}(\mu_\tau, \frac{1}{\tau}), \tag{2.12}$$

where $\mu_\tau = \frac{1}{m_\tau}\sum_{k=1}^{m_\tau}\mu_{\tau,k}$. Using (2.12) and the equation $n = \sum_{\tau=1}^{n}\tau \cdot m_\tau$, we have:

$$\begin{aligned}\frac{1}{n}\log M &\leq \sum_{\tau=1}^{n}[\frac{\tau \cdot m_\tau}{\sum_{\tau=1}^{n}\tau \cdot m_\tau}\tilde{H}(\mu_\tau, \frac{1}{\tau})] + \epsilon_n \\ &= \sum_{\tau=1}^{n}[\pi_\tau \tilde{H}(\mu_\tau, \frac{1}{\tau})] + \epsilon_n,\end{aligned} \tag{2.13}$$

where $\pi_\tau = \frac{\tau \cdot m_\tau}{\sum_{\tau=1}^{n}\tau \cdot m_\tau}$.
The packet rates of the users could be written as follows:

$$\begin{aligned}r_e &= \frac{1}{n}\sum_{j=1}^{m}\tau_j\xi_j = \frac{1}{n}\sum_{\tau=1}^{n}\sum_{k=1}^{m_\tau}\tau\mu_{\tau,k} \\ &= \frac{1}{n}\sum_{\tau=1}^{n}\tau m_\tau\frac{1}{m_\tau}\sum_{k=1}^{m_\tau}\mu_{\tau,k} = \sum_{\tau=1}^{n}\frac{1}{n}\tau m_\tau\mu_\tau \\ &= \sum_{\tau=1}^{n}\frac{\tau \cdot m_\tau}{\sum_{\tau=1}^{n}\tau \cdot m_\tau}\mu_\tau = \sum_{\tau=1}^{n}\pi_\tau\mu_\tau,\end{aligned}$$

and

$$r_d = \sum_{\tau=1}^{n}\pi_\tau\frac{1}{\tau}.$$

Therefore, the constraint could be written as follows:

$$\sum_{\tau=1}^{n}\pi_\tau(\mu_\tau + \frac{1}{\tau}) = 1. \tag{2.14}$$

Suppose the set of pairs $\{(\mu_\tau, \frac{1}{\tau})\}_{\tau=1}^{n}$ satisfies (2.14) and maximizes the right hand side of (2.13). By Lemma 4, there exists another set of pairs $\{(\hat{\mu}_\tau, \frac{1}{\tau})\}_{\tau=1}^{n}$ with $\hat{\mu}_\tau$ defined as:

$$\hat{\mu}_\tau = \begin{cases} \mu_\tau & \text{if } 0 \leq \mu_\tau \leq \frac{1}{2}, \\ 1-\mu_\tau & \text{if } \frac{1}{2} \leq \mu_\tau \leq 1, \end{cases}$$

that gives the same value for the right-hand side of (2.13), but it has $\sum_{\tau=1}^{n}\pi_\tau(\hat{\mu}_\tau + \frac{1}{\tau}) \leq \sum_{\tau=1}^{n}\pi_\tau(\mu_\tau + \frac{1}{\tau})$. Therefore, U_d can increase his packet rate and increase the information rate. Therefore, in the maximizing set, for all τ, we have

$0 \le \mu_\tau \le \frac{1}{2}$.
Therefore, the optimal operating point will be on the line $r_e + r_d = 1$, with $0 \le r_e \le \frac{1}{2}$ and $\frac{1}{2} \le r_d \le 1$.

□

Applying Lemma 3, we can replace all pairs of form $(\mu_\tau, \frac{1}{\tau})$, $\tau \ge 2$, with a single pair of form $(\mu, \frac{1}{2})$:

Lemma 6 *Suppose the set of pairs* $\mathcal{S} = \{(\mu_\tau, \frac{1}{\tau}), \tau \in [n]\}$ *where for all* τ, $0 \le \mu_\tau \le \frac{1}{2}$, *with weights* $\{\pi_\tau, \tau \in [n]\}$ *gives rate* $\sum_{\tau=1}^{n} \pi_\tau \tilde{H}(\mu_\tau, \frac{1}{\tau})$ *and has its operating point on the line* $0 \le r_e \le \frac{1}{2}$ *and* $\frac{1}{2} \le r_d \le 1$. *Then, there exists* $0 \le \alpha \le 1$ *and* $0 \le \gamma_1, \gamma_2 \le \frac{1}{2}$ *such that:*

$$\alpha(\gamma_1 + 1) + (1 - \alpha)(\gamma_2 + \frac{1}{2}) = 1,$$

and

$$\sum_{\tau=1}^{n} [\pi_\tau \tilde{H}(\mu_\tau, \frac{1}{\tau})] \le \alpha \tilde{H}(\gamma_1, 1) + (1 - \alpha)\tilde{H}(\gamma_2, \frac{1}{2}).$$

Proof. Suppose

$$\beta_\tau(\mu_1, 1) + (1 - \beta_\tau)(\mu_\tau, \frac{1}{\tau}) = (\mu_2^\tau, \frac{1}{2}) \qquad \forall \tau \in \{3, ..., n\},$$

for some $\beta_\tau \in [0, 1]$. Clearly, the set $\{(\mu_1, 1), (\mu_2, \frac{1}{2}), (\mu_2^3, \frac{1}{2}), ..., (\mu_2^n, \frac{1}{2})\}$ can also give the same operating point as $\mathcal{S}$ does. By Lemma 3,

$$\beta_\tau \tilde{H}(\mu_1, 1) + (1 - \beta_\tau)\tilde{H}(\mu_\tau, \frac{1}{\tau}) \le \tilde{H}(\mu_2^\tau, \frac{1}{2}) \qquad \forall \tau \in \{3, ..., n\}.$$

Therefore,

$$\begin{aligned}
\sum_{\tau=1}^{n} \pi_\tau \tilde{H}(\mu_\tau, \frac{1}{\tau}) &= \zeta_1 \tilde{H}(\mu_1, 1) + \zeta_2 \tilde{H}(\mu_2, \frac{1}{2}) + \sum_{\tau=3}^{n} \zeta_\tau (\beta_\tau \tilde{H}(\mu_1, 1) + (1 - \beta_\tau)\tilde{H}(\mu_\tau, \frac{1}{\tau})) \\
&\le \zeta_1 \tilde{H}(\mu_1, 1) + \zeta_2 \tilde{H}(\mu_2, \frac{1}{2}) + \sum_{\tau=3}^{n} \zeta_\tau \tilde{H}(\mu_2^\tau, \frac{1}{2}) \\
&\le \zeta_1 \tilde{H}(\mu_1, 1) + (1 - \zeta_1)\tilde{H}(\frac{\zeta_2 \mu_2 + \sum_{\tau=3}^{n} \zeta_\tau \mu_2^\tau}{1 - \zeta_1}, \frac{1}{2}),
\end{aligned} \tag{2.15}$$

where $\pi_1 = \zeta_1 + \sum_{\tau=3}^{n} \zeta_\tau \beta_\tau$, $\pi_2 = \zeta_2$ and $\pi_\tau = \zeta_\tau(1 - \beta_\tau)$ for $3 \le \tau \le n$ and we have used Lemma 3 again in the last inequality. □

From Lemmas 5 and 6, we have:

$$\begin{aligned}\frac{1}{n}\log M &\leq \alpha\tilde{H}(\gamma_1, 1) + (1-\alpha)\tilde{H}(\gamma_2, \frac{1}{2}) + \epsilon_n \\ &\leq \sup_{\alpha,\gamma_1,\gamma_2} \alpha\tilde{H}(\gamma_1, 1) + (1-\alpha)\tilde{H}(\gamma_2, \frac{1}{2}) + \epsilon_n.\end{aligned}$$

Letting $n \to \infty$, ϵ_n goes to zero and we have

$$\begin{aligned}C \leq \sup_{\alpha,\gamma_1,\gamma_2} \alpha\tilde{H}(\gamma_1, 1) + (1-\alpha)\tilde{H}(\gamma_2, \frac{1}{2}) \\ s.t. \\ \alpha(\gamma_1 + 1) + (1-\alpha)(\gamma_2 + \frac{1}{2}) = 1,\end{aligned}$$

where $0 \leq \alpha \leq 1$ and $0 \leq \gamma_1, \gamma_2 \leq \frac{1}{2}$.

This completes the proof of the converse part.

2.2.2 Achievability side

The sequence of steps in our achievability scheme is as follows:

- Set $\alpha = 0.177 - \delta$, for a small and positive value of δ.
- Fix a binary distribution P_1 such that $P_1(1) = 0.43$ and $P_1(0) = 0.57$. Generate a binary codebook $\mathcal{C}_1$ containing $2^{\alpha n R_1}$ sequences of length αn of i.i.d. entries according to P_1.
  Fix a ternary distribution P_2 over set of symbols $\{a_0, a_1, a_2\}$ such that $P_2(a_0) = 0.43$, $P_2(a_1) = 0.325$ and $P_2(a_2) = 0.245$. Generate a ternary codebook $\mathcal{C}_2$ containing $2^{(1-\alpha)nR_2}$ sequences of length $\frac{1}{2}(1-\alpha)n$ of i.i.d. entries according to P_2. Substitute a_0 with 00, a_1 with 10 and a_2 with 11, so we will have $2^{(1-\alpha)nR_2}$ binary sequences of length $(1-\alpha)n$.
  Combine $\mathcal{C}_1$ and $\mathcal{C}_2$ to get $\mathcal{C}$, such that $\mathcal{C}$ has $2^{n(\alpha R_1 + (1-\alpha)R_2)}$ binary sequences of length n where we concatenate i^{th} row of $\mathcal{C}_1$ with j^{th} row of $\mathcal{C}_2$ to make the $((i-1)(2^{(1-\alpha)nR_2}) + j)^{th}$ row of $\mathcal{C}$ (note that $2^{(1-\alpha)nR_2}$ is the number of rows in $\mathcal{C}_2$). Rows of $\mathcal{C}$ are our codewords. In above, n should be chosen such that $\alpha n R_1$, αn, $(1-\alpha)nR_2$ and $\frac{1}{2}(1-\alpha)n$ are all integers.

- **Encoding:** U_d^T sends the stream of all ones (one packet in each time slot) in the first αn time slots and sends bit stream of concatenated 10's for the rest of $(1-\alpha)n$ time slots.

  To send message m, U_e^T sends the corresponding row of $\mathcal{C}$, that is, it sends the corresponding part of m from $\mathcal{C}_1$ in the first αn time slots and the corresponding part of m from $\mathcal{C}_2$ in the rest of $(1-\alpha)n$ time slots.

- **Decoding:** Assuming the queue is not empty,[2] since there is no noise in the system, the decoder can always learn the exact sequence sent by U_e.

Consequently, we will have:

$$\begin{aligned} C &\geq \frac{\log_2 2^{n(\alpha R_1 + (1-\alpha)R_2)}}{n} \\ &= \alpha R_1 + (1-\alpha)R_2. \end{aligned}$$

In infinite block-length regime, where $n \to \infty$, we can choose $R_1 = H(P_1)$, $R_2 = \frac{1}{2}H(P_2)$ and find codebooks $\mathcal{C}_1$ and $\mathcal{C}_2$ such that this scheme satisfies the rate constraint. Therefore,

$$C \geq \alpha H(P_1) + (1-\alpha)\frac{1}{2}H(P_2).$$

Substituting the values in the expression above, and letting δ go to zero, we see that the rate 0.8114 bits per time slot is achievable.

[2]Since in our achievable scheme, U_d^T's packets are spaced by either one or two time slots, it is enough to have one packet buffered in the queue, where since we are working in the heavy traffic regime, it will not be a problem.

CHAPTER 3

FCFS SCHEDULER WITH 3 USERS

As an extension to the basic problem, in this section we consider the case that a third user is also using the shared scheduler. We add user U_p to our basic system model. This user has nodes U_p^T and U_p^R as his transmitter and receiver nodes, respectively (see Figure 3.1). We assume that the node U_p^T sends packets according to a Bernoulli process with rate r_p to the shared scheduler. The shared scheduler is again assumed to be FCFS with service rate 1 and we analyze the capacity for coding schemes satisfying queueing stability condition, in the asymptotic regime where the operating point is converging to the line $r_e + r_p + r_d = 1$. Also, in this section we consider the extra assumption that the inter-arrival time of the packets in the probe stream is upper bounded by the value τ_{max}. Assuming that a sufficient number of packets are buffered in the shared queue, user U_d can still count the number of packets sent by the other two users between any of his own consecutive packets, yet he cannot distinguish between packets sent by user U_e and the packets sent by user U_p. Hence, user U_d has uncertainty in estimating the values of X. We model this uncertainty as a noise in receiving X.

3.1 Channel Coding Theorem

Suppose U_d^T sends two packets with $\tau_i = 2$. Each of the other users can possibly send at most 2 packets in the interval $[A_i, A_{i+1})$ and hence, $Y \in \{0, 1, 2, 3, 4\}$. Therefore, we have the channel shown in Figure 3.2 for this instance. In the general case, for the inter-arrival time τ, given $X = x$, we have $Y \in \{x + 0, ..., x + \tau\}$ such that

$$Pr(Y = i + x | X = x) = \binom{\tau}{i} (r_p)^i (1 - r_p)^{\tau - i} \qquad i \in \{0, ..., \tau\}, \qquad (3.1)$$

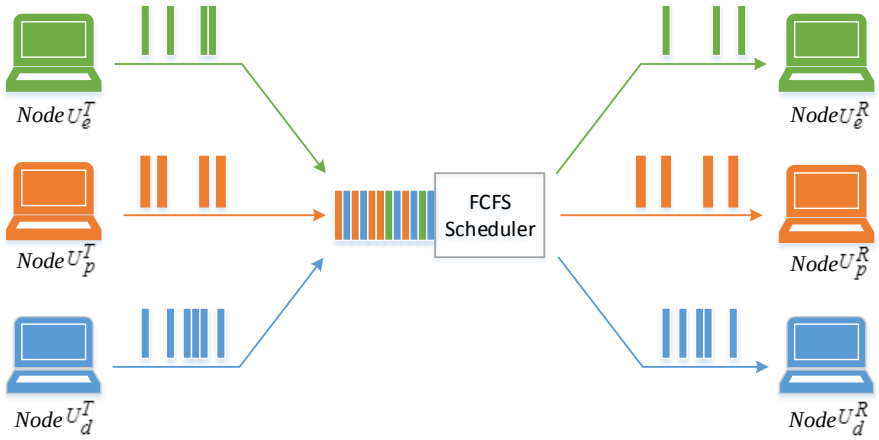

Figure 3.1: Covert queueing channel in a system with 3 users.

which is a binomial distribution $Bin(\tau, r_p)$. Therefore, the support of the random variable Y is $\{0, 1, ..., 2\tau\}$. For the mean of Y, we have:

$$\mathbb{E}[Y|\tau] = \mathbb{E}[\mathbb{E}[Y|X, \tau]|\tau] = \mathbb{E}[X + \tau r_p|\tau] = \tau(r_e + r_p). \tag{3.2}$$

Because of user U_p's stream, the encoder is not aware of the stream received at node U_e^R beforehand and this output can provide information to the encoder about U_p^T's stream. The more packets node U_e^T sends to the scheduler, the more information this stream contains about U_p^T's stream. Using this information, the encoder can have an estimation of the output of the channel at the decoder's side and hence it could be considered as a noisy feedback to the encoder. Figure 3.3 shows the graphical model for random variables in our system.

The main result of this section is evaluation of the capacity of the introduced channel, presented in the following theorem:

Theorem 2 *If the rate of U_p is r_p, the capacity of the timing channel in a shared FCFS scheduler with service rate 1 depicted in Figure 3.1 is given by:*

$$\begin{aligned} C(r_p) = & \sup_{\alpha,\gamma_1,\gamma_2,\tau} \alpha \tilde{I}_{r_p}(\gamma_1, \frac{1}{\tau}) + (1-\alpha)\tilde{I}_{r_p}(\gamma_2, \frac{1}{\tau+1}) \\ s.t. & \\ & \alpha(\gamma_1 + \frac{1}{\tau}) + (1-\alpha)(\gamma_2 + \frac{1}{\tau+1}) = 1 - r_p, \end{aligned} \tag{3.3}$$

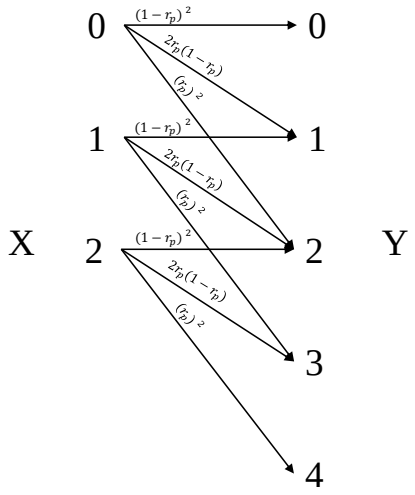

Figure 3.2: Channel between the encoder and the decoder of the system for the case that the inter-arrival time of two packets of the probe stream is 2.

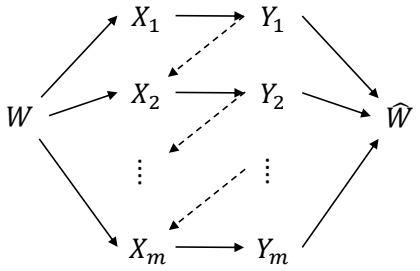

Figure 3.3: The graphical model representing the statistical relation between W, X^m, Y^m and $\hat{W}$.

where $0 \leq \alpha \leq 1$ *and* $0 \leq \gamma_1, \gamma_2 \leq 1$ *and* $1 \leq \tau \leq \tau_{max} - 1$. *The function* $\tilde{I}_{r_p} : [0,1] \times \{\frac{1}{k} : k \in \mathbb{N}\} \mapsto [0,1]$ *is defined as:*

$$\tilde{I}_{r_p}(\gamma, \frac{1}{k}) = \frac{1}{k} \sup_{\substack{X \in \{0,1,\ldots,k\} \\ \mathbb{E}[X] = k\gamma}} I_{r_p}(X;Y), \quad k \in \mathbb{N}, 0 \leq \gamma \leq 1, \tag{3.4}$$

where the subscript r_p *denotes that the mutual information between* X *and* Y *is calculated when the rate of* U_p *is* r_p.

The proof is based on converse and achievability arguments. Before giving the proof, we first investigate some of the properties of the function $\tilde{I}$.

Lemma 7 *The function* $\tilde{I}_{r_p}(\gamma, \frac{1}{k})$ *could be computed using the following expression:*

$$\tilde{I}_{r_p}(\gamma, \frac{1}{k}) = \frac{1}{k} \check{H}_{r_p}(\gamma, \frac{1}{k}) - \frac{1}{k} H(Bin(k, r_p)), \tag{3.5}$$

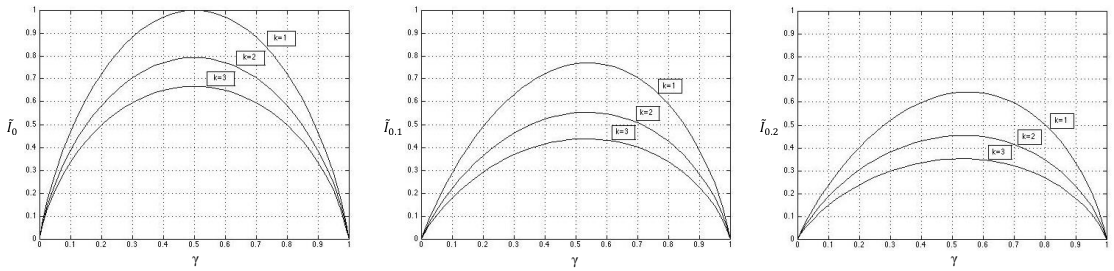

Figure 3.4: $\tilde{I}_{r_p}(\gamma, \frac{1}{k})$ for different values of γ and $k \in \{1, 2, 3\}$ and $r_p \in \{0, 0.1, 0.2\}$.

where $\check{H}_{r_p}(\gamma, \frac{1}{k}) = \sup_{\substack{X \in \{0,1,\ldots,k\} \\ \mathbb{E}[X]=k\gamma}} H_{r_p}(Y)$ *and the second term is the entropy of the binomial distribution with parameters* k *and* r_p.

See Appendix B.1 for the proof of Lemma 7.

In order to calculate $\check{H}_{r_p}(\gamma, \frac{1}{k})$, the following optimization problem should be solved:

$$\begin{aligned} &\max_{P_X \geq 0} \log_2 e \sum_{i=0}^{2k} P_Y(i) \ln(\frac{1}{P_Y(i)}) \\ &\qquad s.t. \quad \begin{cases} \sum_{i=0}^{k} i P_X(i) = \mathbb{E}[X] = k\gamma \\ \sum_{i=0}^{k} P_X(i) = 1 \end{cases} \end{aligned} \tag{3.6}$$

where $P_Y = P_X * P_{Bin(k,r_p)}$, i.e.,

$$P_Y(i) = \sum_{j=0}^{k} P_X(j) P_{Bin(k,r_p)}(i-j) \qquad i \in \{0, 1, ..., 2k\}. \tag{3.7}$$

Figure 3.4 shows the functions $\tilde{I}_0(\gamma, \frac{1}{k})$, $\tilde{I}_{0.1}(\gamma, \frac{1}{k})$ and $\tilde{I}_{0.2}(\gamma, \frac{1}{k})$ for different values of γ and $k \in \{1, 2, 3\}$.

Lemma 8 *For all* $0 \leq r_p \leq 1$, *integers* $1 \leq k_1, k_2, k_3 \leq \tau_{max}$, *values* $0 \leq \gamma_1, \gamma_2, \gamma_3 \leq 1$, *and* $\alpha \in [0, 1]$ *such that* $\alpha(\gamma_1, \frac{1}{k_1}) + (1-\alpha)(\gamma_3, \frac{1}{k_3}) = (\gamma_2, \frac{1}{k_2})$, *we have:*

$$\alpha \tilde{I}(\gamma_1, \frac{1}{k_1}) + (1-\alpha)\tilde{I}(\gamma_3, \frac{1}{k_3}) \leq \tilde{I}(\gamma_2, \frac{1}{k_2}). \tag{3.8}$$

See Appendix B.2 for the proof of Lemma 8.

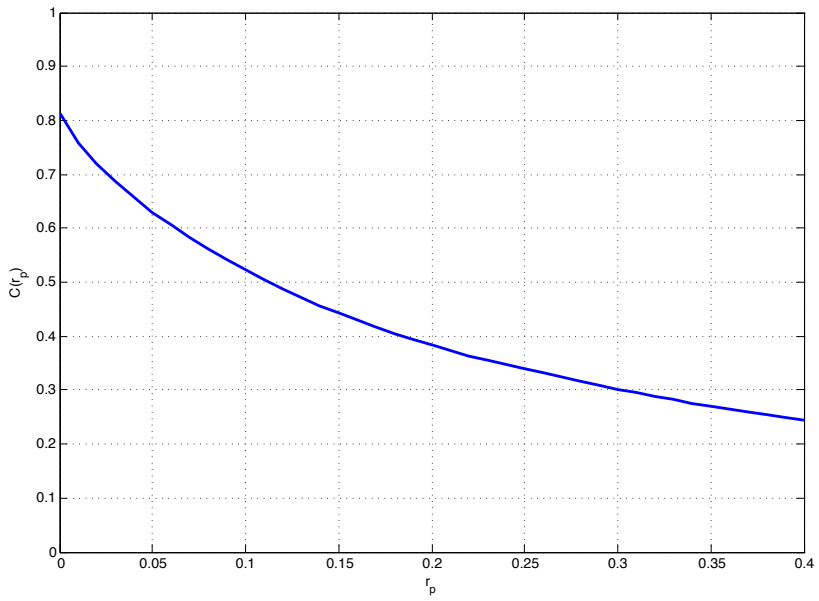

Figure 3.5: Capacity of the timing channel in the shared FCFS scheduler of Figure 3.1 for different values of r_p.

Using the mentioned properties, the capacity of the timing channel in the shared FCFS scheduler of Figure 3.1 for different values of r_p can be calculated. Figure 3.5 shows the value of the capacity with respect to r_p.

The following proof of Theorem 2 is based on converse and achievability arguments.

3.1.1 Converse side

Suppose the rate of U_p is r_p. Similar to the proof of Lemma 5, for any (n, M, ϵ)-code, we have:

$$\begin{aligned}\frac{1}{n}\log M &\leq \frac{1}{n} I_{r_p}(W; \hat{W}|\tau^m) + \epsilon_n \\ &\overset{(a)}{\leq} \frac{1}{n} I_{r_p}(W; Y^m|\tau^m) + \epsilon_n,\end{aligned}$$

where $\epsilon_n = \frac{1}{n}(H(P_e) + P_e \log_2(M-1))$ and (a) follows from data processing inequality in the model in Figure 3.3. Therefore:

$$\begin{aligned}
\frac{1}{n}\log M &\leq \frac{1}{n}\sum_{j=1}^{m} I_{r_p}(W; Y_j | Y^{j-1}\tau^m) + \epsilon_n \\
&\leq \frac{1}{n}\sum_{j=1}^{m} I_{r_p}(W, Y^{j-1}; Y_j | \tau^m) + \epsilon_n \\
&\overset{(a)}{\leq} \frac{1}{n}\sum_{j=1}^{m} I_{r_p}(X_j; Y_j | \tau^m) + \epsilon_n \\
&\leq \frac{1}{n}\sum_{j=1}^{m} \max_{P_{X_j|\tau^m}} I_{r_p}(X_j; Y_j | \tau^m) + \epsilon_n,
\end{aligned}$$

where (a) again follows from data processing inequality in the model in Figure 3.3. In the maximization above, the mean of the distribution $P_{X_j|\tau^m}$ is $\mathbb{E}[X_j|\tau^m]$ and in order to find the maximum information rate, the set of means, $\{\mathbb{E}[X_1|\tau^m], \mathbb{E}[X_2|\tau^m], ..., \mathbb{E}[X_m|\tau^m]\}$, should satisfy the constraint $r_e + r_p + r_d = 1$, that is $\frac{1}{n}\sum_{j=1}^{m}\mathbb{E}[X_j|\tau^m] + r_p + r_d = 1$. Let $\xi_j = \frac{\mathbb{E}[X_j|\tau^m]}{\tau_j}$. Using (3.4), we have:

$$\max_{\substack{P_{X_j|\tau^m} \\ \mathbb{E}[X_j|\tau^m]=\tau_j\xi_j}} I_{r_p}(X_j; Y_j | \tau^m) = \tau_j \tilde{I}_{r_p}(\xi_j, \frac{1}{\tau_j}). \tag{3.9}$$

Therefore, we will have:

$$\frac{1}{n}\log M \leq \frac{1}{n}\sum_{j=1}^{m} \tau_j \tilde{I}(\xi_j, \frac{1}{\tau_j}) + \epsilon_n.$$

Next, using Lemma 8 and similar to the proof of Lemma 5, by breaking the summation, using Jensen's inequality and the equation $n = \sum_{\tau=1}^{\tau_{max}} \tau \cdot m_\tau$, we will have:

$$\frac{1}{n}\log M \leq \sum_{\tau=1}^{\tau_{max}} \pi_\tau \tilde{I}_{r_p}(\mu_\tau, \frac{1}{\tau}) + \epsilon_n, \tag{3.10}$$

where μ_τ is the average of ξ_j's which have $\tau_j = \tau$ and $\pi_\tau = \frac{\tau \cdot m_\tau}{\sum_{\tau=1}^{\tau_{max}} \tau \cdot m_\tau}$. In this expression, π_τ could be interpreted as the portion of time that user U_d sends packets with inter-arrival time equal to τ.

Also, using the same approach as the one used in the proof of Lemma 5, the constraint of the problem could be written as follows:

$$\sum_{\tau=1}^{\tau_{max}} \pi_\tau(\mu_\tau + \frac{1}{\tau}) = 1 - r_p. \tag{3.11}$$

Suppose the set of pairs $\mathcal{S} = \{(\mu_\tau, \frac{1}{\tau}), \tau \in [\tau_{max}]\}$ with weights $\{\pi_\tau, \tau \in [\tau_{max}]\}$ gives rate $\sum_{\tau=1}^{\tau_{max}} \pi_\tau \tilde{I}_{r_p}(\mu_\tau, \frac{1}{\tau})$ and has its operating point on the line $r_e + r_d = 1 - r_p$, and we have

$$\frac{1}{\tau^*} \leq \sum_{\tau=1}^{\tau_{max}} \pi_\tau \frac{1}{\tau} \leq \frac{1}{\tau^*+1},$$

for some $1 \leq \tau^* \leq \tau_{max} - 1$. Suppose

$$\begin{cases} \beta_\tau(\mu_{\tau^*+1}\frac{1}{\tau^*+1}) + (1-\beta_\tau)(\mu_\tau, \frac{1}{\tau}) = (\mu^\tau_{\tau^*}, \frac{1}{\tau^*}) & \tau \leq \tau^* - 1, \\ \beta_\tau(\mu_{\tau^*}\frac{1}{\tau^*}) + (1-\beta_\tau)(\mu_\tau, \frac{1}{\tau}) = (\mu^\tau_{\tau^*+1}, \frac{1}{\tau^*+1}) & \tau \geq \tau^* + 2, \end{cases}$$

for some $\beta_\tau \in [0,1]$. Clearly, the set $\{(\mu^1_{\tau^*}, \frac{1}{\tau^*}), \cdots, (\mu^{\tau^*-1}_{\tau^*}, \frac{1}{\tau^*}), (\mu_{\tau^*}, \frac{1}{\tau^*}), (\mu_{\tau^*+1}, \frac{1}{\tau^*+1}), (\mu^{\tau^*+2}_{\tau^*+1}, \frac{1}{\tau^*+1}), \cdots, (\mu^{\tau_{max}}_{\tau^*+1}, \frac{1}{\tau^*+1})\}$ can give the same operating point as $\mathcal{S}$ does. Therefore, by using the technique used in (2.15) and twice use of Lemma 8 we have

$$\begin{aligned} \frac{1}{n}\log M &\leq \alpha\tilde{I}_{r_p}(\gamma_1, \frac{1}{\tau^*}) + (1-\alpha)\tilde{I}_{r_p}(\gamma_2, \frac{1}{\tau^*+1}) + \epsilon_n \\ &\leq \sup_{\alpha,\gamma_1,\gamma_2,\tau} \alpha\tilde{I}_{r_p}(\gamma_1, \frac{1}{\tau}) + (1-\alpha)\tilde{I}_{r_p}(\gamma_2, \frac{1}{\tau+1}) + \epsilon_n. \end{aligned}$$

Letting $n \to \infty$, ϵ_n goes to zero and we get the desired result.

3.1.2 Achievability side

Achieving the proposed upper bound could be done by a method exactly like the one used in Subsection 2.2.2. We need to solve the optimization problem (3.3) to find parameters α, γ_1, γ_2 and τ. Because of Lemma 8, in order to find the optimal τ, we can start with $\tau = 1$ and optimize other parameters, and then calculate $\alpha\tilde{I}_{r_p}(\gamma_1, \frac{1}{\tau}) + (1-\alpha)\tilde{I}_{r_p}(\gamma_2, \frac{1}{\tau+1})$ and, in each step, increase the value of τ by 1, stopping whenever the obtained value is decreased com-

pared to the previous step. For instance, for $r_p \leq 0.1$, the optimal τ is 1, and hence the procedure stops after checking two steps. After calculating parameters γ_1, γ_2 and τ, the optimal input distribution could be obtained using optimization problem (3.6).

CHAPTER 4

ROUND ROBIN SCHEDULER

In this chapter we focus on the round robin scheduler, which is another throughput optimal policy commonly used in computer processors and communication networks.

Our main contributions are the following:

- We characterize the optimum signaling scheme for the covert queuing channel with round robin scheduler (Section 4.2).
- We calculate the capacity of this covert queuing channel and show that it is approximately equal to 0.6942 bits per time slot (Subsection 4.3.1).
- We propose practical optimal finite block length coding schemes for both fixed and variable length codewords. Our proof along with the simulation results shows that the rates of the proposed optimal coding schemes approach the capacity as the number of messages goes to infinity (Subsection 4.3.2).
- Finally, we extend the model to a more realistic noisy scenario in which users' packets may drop, and calculate the capacity of the covert channel in this case as a function of probability of packet drop (Section 4.4).

For round robin scheduler, we only consider a system with two users; hence, for ease of representation, in the sequel, we will refer to users U_e and U_d as Alice and Bob respectively. Therefore, Alice will be the transmitter of the message and Bob will be the receiver. The system used in this chapter is depicted in Figure 4.1.

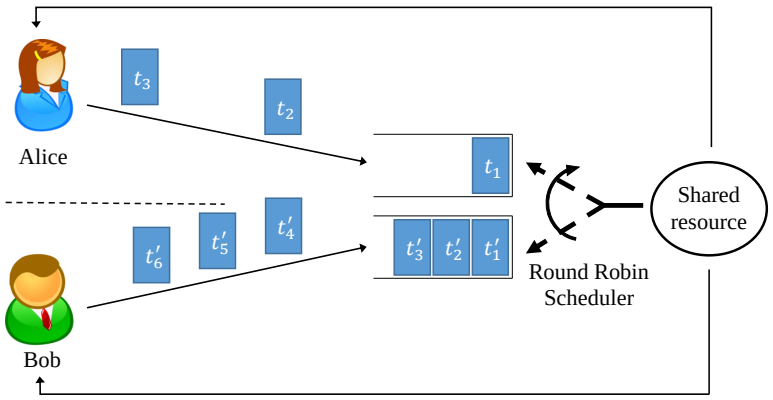

Figure 4.1: System Setup: Alice and Bob share a resource arbitrated in round robin fashion. Users get acknowledgments when their packets are served.

4.1 System Description

We consider a system in which a shared resource services jobs from two users, Alice and Bob, using round robin policy. Time is assumed to be discretized into slots, and each packet generated by users is served in one time slot. The scheduler can serve one packet in each time slot. We follow the common convention that the packets arrive at the beginning of time slots and the departures occur at the end of time slots. Each user's packets are buffered in a separate queue, and the round robin scheduler picks packets from the two queues as follows. In each time slot,

- If both users' arrival queues are empty, the system remains idle and resumes scheduling in the next time slot.
- If only one user's queue has a packet, the current slot is given to that user, and the scheduler continues scheduling in the next time slot.
- If both users have waiting packets, the scheduler always gives priority to a fixed user. That is, the current time slot is allocated to serve a packet from the user with priority, and the next time slot will be allocated to the other user. The system continues scheduling after both users are served. Without loss of generality, we assume that the priority is always given to Bob in the sequel.

We assume both Alice and Bob send at most one packet per time slot. Thus, their packet stream can be modeled as a binary bit stream, where bit '1'

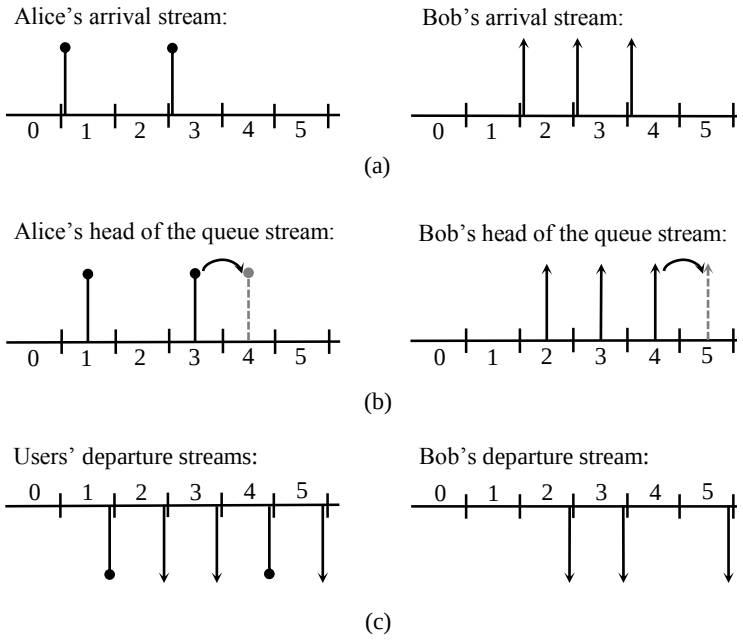

Figure 4.2: (a) Arrival stream for Alice and Bob; (b) users' head-of-the-queue streams (the gray symbol is in fact the same packet in the previous time slot which has not yet been served, and has remained in the head of the queue for one time slot); (c) the users' departure streams.

indicates a packet was sent, and bit '0' indicates no packet was transmitted. Since the scheduler can serve at most one packet per time slot, the sum of users' packet rates should be less than one for stability. That is, $\lambda_1 + \lambda_2 < 1$, where λ_1 and λ_2 denote Alice and Bob's packet rates, respectively (see Appendix C.1 for the proof of stability). Figure 4.1 depicts the system setup. In this depiction, each packet is marked by its arrival time. As shown in this figure, there is a feedback line from the shared resource to the users, which notifies them when their packet is served. Clearly, this allows the users to infer the status of the head of their queue.

Figure 4.2 depicts an example of the system scheduling. In this and other such figures, Alice's packets are shown by circled tip arrows and Bob's packets by regular arrows. For each user, the arrival stream, the head-of-the-queue stream and the departure stream are shown. By arrival stream, we mean the actual packet stream sent by the user, and by head-of-the-queue stream, we mean the packets ready to be served at the head of the corresponding user's queue. Therefore, at any given time slot, the head of the queue can be '1' even though no packet arrived in that slot. In part (b) of Figure 4.2, the

packet denoted by the gray dashed line indicates that it has been the same as its previous packet, which has been made to wait in the queue for one time slot to receive service in the next time slot (we emphasize that the gray dashed symbols are not packet arrival). The downward streams (Figure 4.2 (c)) indicate the departure time of users' packets.

Suppose Alice aims to send message W uniformly drawn from the set $\{1, 2, \ldots, M\}$. To this end, Alice encodes this message to a bit stream X^m which is sent out as a packet stream with arrival times A_A^n. Based on scheduling policy and both Alice's and Bob's packet arrivals, Bob receives a stream of acknowledgments from the system which is denoted by D_B^n. Finally, Bob transforms this stream to a bit stream Y^m which will be decoded to message $\hat{W}$. As a result, we have the following Markov chain:

$$W \to X^m \to A_A^n \to D_B^n \to Y^m \to \hat{W}. \tag{4.1}$$

The noise in the system is modeled as follows. The packets generated by either Alice or Bob may be dropped in the link between the users and the shared resource with probability δ. Note that this noise can affect the transmissions in $X^m \to A_A^n$ and $A_A^n \to D_B^n$ in Markov chain (4.1). We will later show that with the optimum signaling scheme between Alice and Bob, noise will not affect the process $A_A^n \to D_B^n$.

4.2 Optimum Signaling Scheme

The main idea behind the signaling scheme from Alice to Bob is to utilize the delays occurred in Bob's departure stream caused by Alice's packets. First, we investigate the optimum stream sent by Bob:

Lemma 9 *In order that Bob maximizes his inference from Alice's signaling (the number of slots Bob receives a bit from Alice), he should have a ready-to-be-served packet at the head of his queue in all time slots.*

Proof. We state the proof by contradiction. Suppose Bob does not have any packets to be served at time slot n. Then the round robin policy will restart scheduling in time slot $n+1$ regardless of whether Alice had a packet in time slot n or not. Therefore, Alice cannot affect the departure time of Bob as

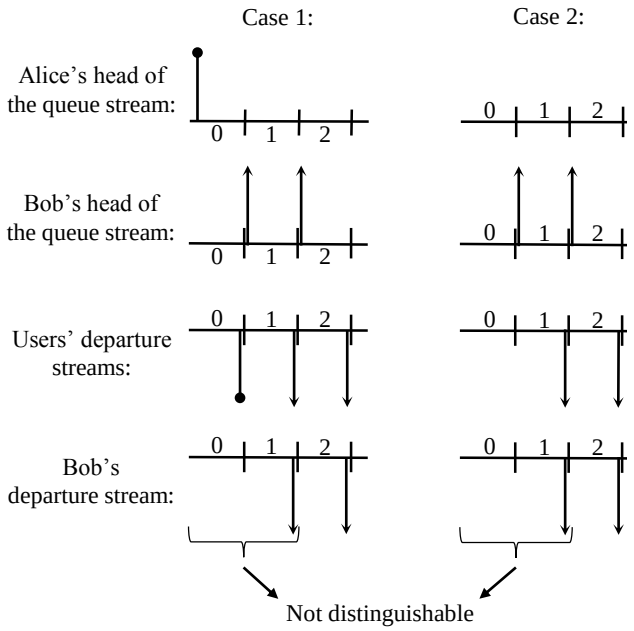

Figure 4.3: The head of the queue and the departure stream of the users when Alice sends bits '0' or '1' while Bob does not have a ready-to-be-served packet at the head of his queue. As shown in the last row, the services given to Bob in both cases are the same, and hence Bob is not capable of distinguishing between the bits '0' and '1' sent by Alice.

shown in Figure 4.3. This means that Alice cannot communicate with Bob in time slot n.

□

The requirement that Bob should have a ready-to-be-served packet at all time slots does not mean that he has to send a packet in all time slots. It suffices for him to fix his queue length at some nonzero length, and whenever one of his packets is served, he generates a packet to ensure his queue length remains nonzero. This strategy allows him to keep the sum rate of arrivals from Alice and Bob less than 1 and keep the system stable.

In the following we illustrate the effect of Alice's bits '0' and '1' on the acknowledgments given to Bob.

- **Signaling bit '1':** To signal bit '1' in time slot n, Alice must have a head-of-the-queue packet at the beginning of the time slot. Recall that Bob has a ready-to-be-served packet in all time slots. Thus, round robin policy will serve Bob and Alice at time slots n and $n+1$, respectively.

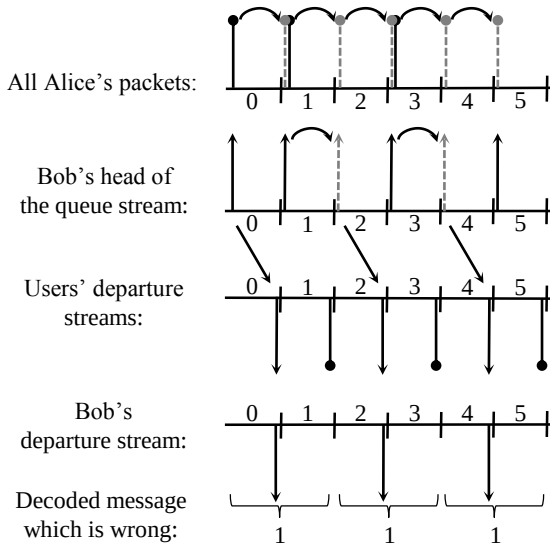

Figure 4.4: Visualization of the case when Alice intends to send message 1101 to Bob, but she does not idle for one time slot after she sends her packets. As shown, bit '0' disappears and Bob decodes 111.

Therefore, when Bob receives service in a time slot but does not receive service in the next time slot, he decodes bit '1'.

- **Signaling bit '0':** To signal bit '0' in time slot n, Alice must not have a head-of-the-queue packet at the beginning of the time slot. Because Bob has a packet which is ready to be serviced in the head of the queue in this time slot, he receives service at time slot n, and the scheduler resets for time slot $n+1$. As a result, at time slot $n+1$, Bob is served again. Therefore, if Bob receives service in two consecutive time slots, he decodes it bit '0'.

Remark 1 *Note that Alice cannot send two packets in two consecutive time slots. This is because if Alice sends two (or more) packets in consecutive time slots, her next (or more) '0'(s) would disappear as her packets are accumulated in the queue. Therefore, bit '1' of message $\mathcal{W}$ effectively requires two time slots for transmission.*

We clarify Remark 1 in more detail in the following example.

Example 1 *Assume that Alice's queue is empty and she wants to transmit* 1101 *to Bob. If Alice sends bits of '1' in her message immediately in each time slot (as depicted in Figure 4.4), Bob would erroneously decode message*

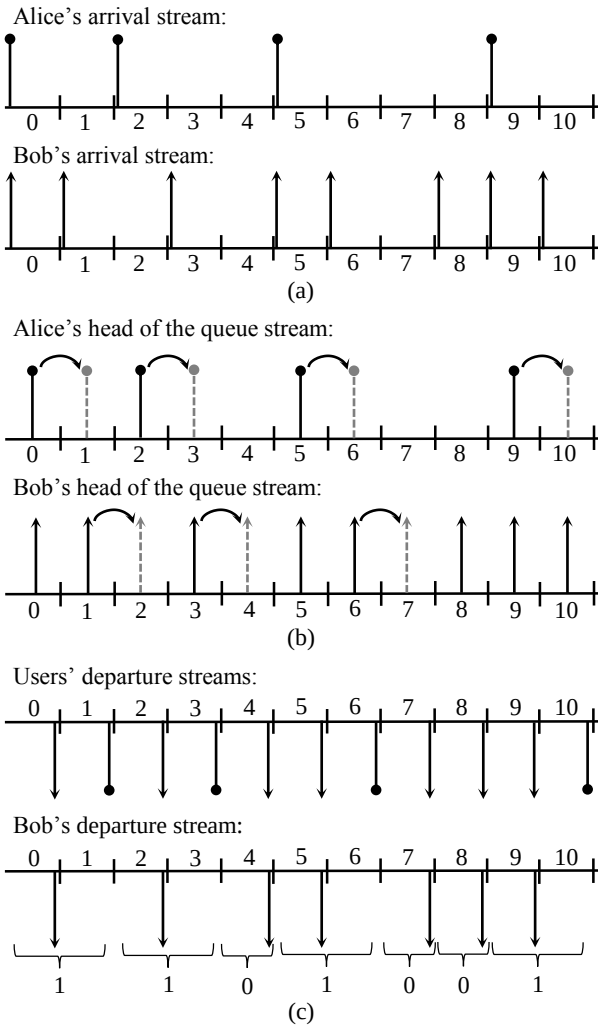

Figure 4.5: Example of correct signaling between Alice and Bob.

111. *This is caused by the accumulation of packets in Alice's queue, stemming from existence of a packet at the head of her queue before clearing the previous '1'.*

As mentioned earlier, to solve this problem, Alice must not send two packets in consecutive time slots; i.e., she must idle for one time slot whenever she sends a packet. Figure 4.5 shows an example of an effective communication, in which Alice is sending the bit stream 1101001 to Bob.

In the next section, we find the maximum achievable rate at which Alice can communicate reliably with Bob through the timing covert channel.

4.3 Noiseless Covert Channel

In this section we calculate the capacity of the introduced covert channel and investigate the optimum coding schemes in finite-length codeword regime.

4.3.1 Coding Theorem

We define the used performance metric, code, information transmission rate of a code, achievable rate, and channel capacity similar to the definitions in Chapter 2. The definitions are repeated in the following for convenience.

The performance metric used is the average error probability defined as follows:

$$P_e \triangleq \mathbb{P}(W \neq \hat{W}) = \sum_{m=1}^{M} \frac{1}{M} \mathbb{P}(\hat{W} \neq m | W = m). \tag{4.2}$$

Definition 4 *An (n, M, ϵ)-code consists of a codebook of size M with equiprobable binary codewords of average length n satisfying $P_e \leq \epsilon$.*

Definition 5 *The information transmission rate of a code is*

$$R = \frac{logM}{n},$$

which is the amount of conveyed information normalized by the average number of used time slots.

Definition 6 *A rate R is said to be achievable if there exists a sequence of (n, M, ϵ_n)-codes such that $\epsilon_n \to 0$ as $n \to \infty$.*

Definition 7 *The channel capacity is the supremum achievable rate at which Alice can communicate through the covert channel with Bob.*

The fundamental limit of the introduced channel in information transmission is presented in the following theorem:

Theorem 3 *The capacity of the introduced covert channel between Alice and Bob created by the shared resource arbitrated by a round robin scheduler is*

$$C = \sup_{p \in [0,1]} \frac{h(p)}{1+p}, \tag{4.3}$$

where p is the probability of sending message bit '1' by Alice and $h(\cdot)$ is the binary entropy function. The maximum of (4.3) *is approximately* 0.6942 *achieved at* $p = \frac{3-\sqrt{5}}{2}$.

The proof of Theorem 3 appears in Appendix C.2. The intuition behind the proof is as follows: Alice can send bits '0' or '1' to Bob in 1 and 2 time slots, respectively. Therefore, it takes on average $2 \times p + (1-p)$ time slots for Alice to send a bit. As a result, Alice can transmit bits '0' and '1' to Bob at rate $\frac{h(p)}{2\times p+(1-p)}$. The capacity can be calculated by taking the supremum on this rate with respect to $p \in [0, 1]$ while noting that the channel is memoryless.

4.3.2 Finite-length Codeword Regime

As mentioned earlier, Alice encodes each message to a binary sequence and creates a codebook $\mathcal{C}$, known to both Alice and Bob. The codewords in the codebook could be all of the same or different lengths. In the following two subsections, we will consider both these scenarios and find the optimum codebook for the setting.

Variable-length Codewords

In this subsection, for any fixed number of messages, we propose an algorithm which generates the optimum variable-length codebook, i.e., the list of codewords that results in maximum communication rate between the users. By Definition 5, the rate at which Alice can communicate with Bob is as follows:

$$R = \frac{log(M)}{\frac{1}{M}\sum_{m=1}^{m=M} T_m}, \tag{4.4}$$

where T_m is the transmission time of the m-th codeword. As we discussed in Section 4.1, transmission of bit '1' takes two time slots, while bit '0' requires one time slot. Denote the number of bits '0' and '1' in the codebook by n_0 and n_1, respectively. We have $\sum_{m=1}^{m=M} T_m = 2n_1 + n_0$, and (4.4) could be rewritten as

$$R = \frac{Mlog(M)}{2n_1 + n_0}. \tag{4.5}$$

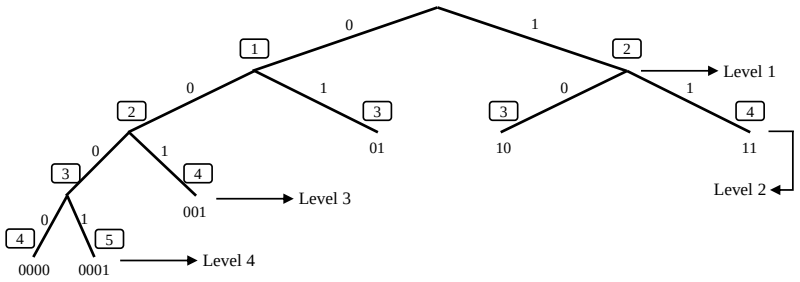

Figure 4.6: A tree representation of the codewords. Codewords are the leaves of the tree. The cost of a codeword, defined to be its transmission time, is written in the boxes.

Given that M is the fixed given parameter, maximizing the rate is equivalent to searching for a codebook which achieves the minimum of the denominator in (4.5).

Our technique for finding the optimum codebook is as follows. Represent each codeword in the codebook by a leaf in a tree, as depicted in Figure 4.6. The numbers in boxes in Figure 4.6 denote the cost of each codeword, defined as the number of time slots required for transmission of that codeword. Call the resulting graph the *codeword tree*. In this representation, for each node, the branch to the left (right) side appends a 0 (1) to the codeword corresponding to that node. For example, if a node represents 00101, its left and right children will represent codewords 00101**0** and 00101**1**, respectively. The reason we use the leaves of a tree for representing the codewords is to guarantee that the codewords are uniquely decodable [22]. Algorithm 1 describes how the M optimal codewords are selected from the tree.

After initializing the codebook to $\{0, 1\}$, in each iteration of Algorithm 1, one of the current codewords is replaced with its two children. This procedure is repeated until all M codewords are obtained. The codeword replaced by its children at each iteration is one with the minimum cost. As an example, the result of Algorithm 1 for $M = 6$ is depicted in Figure 4.6.

Algorithm 1 Finding the optimum codebook with M codewords

1: *Initialize the codebook to be* $\{0, 1\}$
2: **while** *number of codewords is less than* M **do**
3: *Choose a codeword with the minimum cost in the current codebook and replace it by its two children (by adding* 0 *and* 1 *to the right side of the code) to build a new codebook.*
4: **end while**

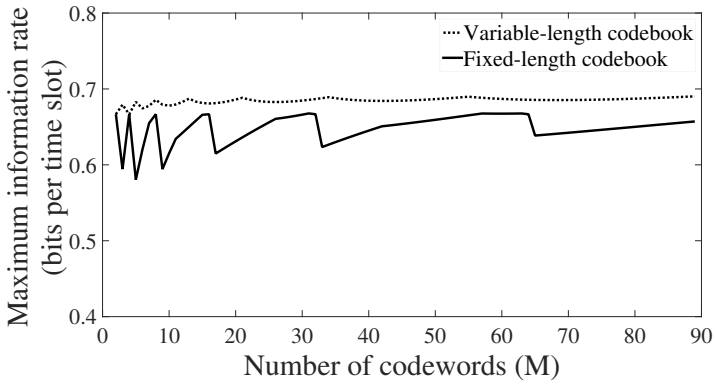

Figure 4.7: The maximum rates at which Alice can communicate with Bob versus the number of codewords for two cases of variable and fixed-length codebooks.

Theorem 4 *For a fixed given number of equiprobable messages, Algorithm 1 is optimal in the sense that it provides a codebook which maximizes the communication rate between the users.*

The proof of Theorem 4 appears in Appendix C.3.

The maximum rate at which Alice can communicate with Bob versus the number of codewords, M, is depicted in Figure 4.7. The overall trend of the maximum communication rate increases as the number of codewords increases, and converges to the capacity. The following theorem formalizes this claim.

Theorem 5 *The information transmission rate of a codebook created by Algorithm 1 converges to the capacity of the covert channel as the number of messages goes to infinity.*

The proof of Theorem 5 appears in Appendix C.4.

Fixed-length Codeword

In many applications, using variable-length codewords is not desirable from the designer's point of view. For example, in a noisy system, a variable-length scheme may lead to loss of synchronization between encoder and decoder. To obtain fixed-length codewords, all M codewords must be selected from the

same level of the tree. Such a constraint on choosing codewords can lead to reduction in information rate for a fixed number of messages, but as we shall see, these codes can still achieve the capacity as when the length of the codewords goes to infinity. Algorithm 2 shows how to select the optimal fixed-length codebook for a given number of messages. Before stating the algorithm, we need the following definition. Denote the cost of a codebook and the cost of a codeword with $\eta(\mathcal{C})$ and $\eta(\mathcal{W})$, respectively. Note that $\eta(\mathcal{C}) = n_0(\mathcal{C}) + 2n_1(\mathcal{C})$.

Algorithm 2 Finding the optimum fixed-length codebook with M codewords

1: *Set* $\hat{l} = \lceil log(M) \rceil$.
2: **for** $l = \hat{l}$ *to* $2\hat{l}$ **do**
3: $\quad \mathcal{C}_l$ *= Set of* M *codewords with the least number of bits* 0 *in the* l*-th level of the codeword tree.*
4: $\quad \eta(\mathcal{C}_l) = n_0(\mathcal{C}_l) + 2n_1(\mathcal{C}_l)$.
5: **end for**
6: *Output* $\mathcal{C}_{l^*}$ *such that* $l^* = \arg\min_l \eta(\mathcal{C}_l)$.

The optimal codebooks with the least number of bits $'1'$ should be chosen in each of the levels $\hat{l} = \lceil \log(M) \rceil$ to $2\hat{l}$. Then, the optimal codebook is the one with minimum cost among these created codebooks.

Theorem 6 *For a fixed given number of equiprobable messages, Algorithm 2 outputs the optimal fixed-length codebook.*

The proof of Theorem 6 appears in Appendix C.5.

As shown in Figure 4.7 using Algorithm 2 the overall trend of the maximum communication rate increases as the number of codewords increases, and converges to the capacity. The following theorem formalizes this claim.

Theorem 7 *The information transmission rate of a codebook created by Algorithm 2 converges to the capacity of the covert channel as the number of messages goes to infinity.*

For the proof of Theorem 7, see Appendix C.4.

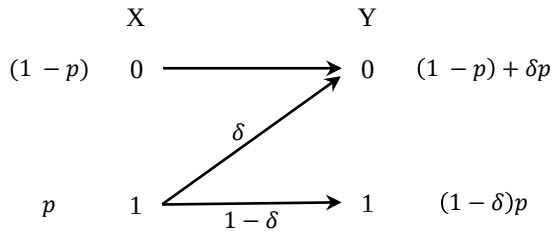

Figure 4.8: Z-channel model of the covet channel with packet drops.

4.4 Noisy Covert Channel

In this section we consider the case where the channel between the users is noisy. The noise model is as follows. We assume that a packet generated by a user is dropped with probability δ. In the following lemma, we investigate the effect of the noise on the covert channel between the users.

Lemma 10 *In the optimum signaling scheme proposed in Section 4.2, packet drops converts the channel between Alice and Bob to a Z-channel. That is, '0' is always transmitted error free, but '1' is flipped with probability δ.*

Figure 4.8 shows the resulting Z-channel. Note that the channel is depicted between X and Y in Markov chain (4.1), but the noise occurs between X and A_A.

Proof. As discussed in Section 4.2, Bob should keep his queue length positive at all time slots. If he keeps his queue length large enough, even if his packets are dropped in multiple time slots, he still has remaining ready-to-be-served packets in his queue. Thus, by letting the probability of his queue length becoming zero be arbitrary small, Bob can avoid dropped packets impacting the scheme.

Noise also does not affect data transmission when Alice sends '0' as she does not send any packets in this case. On the other hand, when Alice sends a packet to communicate bit '1', this '1' changes to '0' if the packet is dropped which happens with probability δ. As discussed in Section 4.2, normally Alice should wait for one time slot after she sends a packet; however, when packet drops occur she does not need to wait out for a time slot. Alice can always tell that a drop has occurred because she knows her queue length at the end of each time slot. Thus the aggregate effect of noise may be modeled as a Z-channel. □

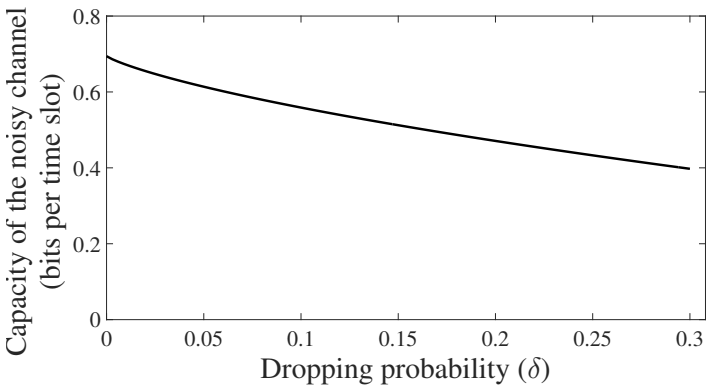

Figure 4.9: Capacity of the noisy covert channel versus the drop probability δ.

In the following theorem, capacity of the covert channel with noise is given.

Theorem 8 *The capacity of the noisy covert channel between Alice and Bob with drop probability δ resulting from Round Robin scheduler is*

$$C = \sup_{p\in[0,1]} \frac{h((1-\delta)p) - ph(\delta)}{(1-p) + \delta p + 2(1-\delta)p}, \tag{4.6}$$

where p is the probability of sending bit '1' by Alice and $h(\cdot)$ is the binary entropy function.

See Appendix C.2 for the proof.

Figure 4.9 depicts capacity C versus the drop probability δ.

In the noisy setting, as mentioned earlier, synchronization between the encoder and decoder sides of the system may be lost. To prevent this from happening, users should utilize the fixed-length codebook design presented in Subsection 4.3.2.

CHAPTER 5

CONCLUSION

We studied convert queueing channels (CQCs) that can occur through delays experienced by users who are sharing a scheduler. As the scheduling policy plays a crucial role in the possible information transmission rate in this type of channel, we focused on work-conserving scheduling policies and studied two commonly used policies of this type.

First we considered first-come-first-served (FCFS) scheduling policy. An information-theoretic framework was proposed to derive the capacity of the CQC under this scheduling policy. We obtained the maximum information transmission rate in this CQC and showed that an information leakage rate as high as 0.8114 bits per time slot is possible. We also considered the effect of the presence of other users on the information transmission rate of this channel.

Next we studied a CQC between two users sharing a round robin scheduler. An information-theoretic framework was again utilized to derive the capacity of this channel in both noisy and noiseless cases. We showed that in the noiseless case an information rate as high as 0.6942 bits per time slot is achievable in this channel. For the noisy case, where users' packets may drop, we again analyzed the highest achievable information rate and obtained the capacity for different levels of noise. Furthermore, we proposed a practical finite-length code construction, which is of more interest from a practical point of view. We designed and analyzed the optimum coding scheme which achieves the capacity limit.

The achievable information transmission rates obtained from this study demonstrate the possibility of significant information leakage and great privacy threats brought by CQCs in FCFS and round robin schedulers. Based on this result, special attention must be paid to CQCs in high security systems.

Finding the capacity of CQCs under other scheduling policies, especially

non-deterministic policies, remains to be done in the research area of covert communications and is considered as the main direction for future work. Furthermore, a comprehensive study is required to design suitable scheduling policies that can simultaneously guarantee adequate levels of both security and throughput. Another important direction for future work is to consider the more practical scenario in which the sizes of packets sent by users vary, and to investigate the role of packet size in the information rate of CQCs.

APPENDIX A

PROOFS FOR CHAPTER 2

A.1 Proof of Stability

For the system model with M users and service rate ρ, the arrival process has a Poisson binomial distribution with probability mass function

$$Pr(K = k) = \sum_{A \in F_k} \prod_{i \in A} r_i \prod_{j \in A^c} (1 - r_j),$$

with support $k \in \{0, 1, ..., M\}$, where F_k is the set of all subsets of k integers that can be selected from $\{1, 2, 3, ..., M\}$. The mean of this distribution is $\mu = \sum_{i=1}^{M} r_i$.

We denote arrival, service and queue length at time k, with $a(k)$, $s(k)$ and $q(k)$, respectively, and we have

$$q(k+1) = (q(k) + a(k) - s(k))^+.$$

Using Foster-Lyapunov theorem with Lyapunov function $V(q(k)) = (q(k))^2$ and calculating the drift, we have

$$\begin{aligned} \mathbb{E}[q^2(k+1) - q^2(k) | q(k) = q] &\leq \mathbb{E}[(q + a - s)^2 - q^2] \\ &= \mathbb{E}[2q(a - s)] + \mathbb{E}[(a - s)^2], \end{aligned}$$

where $\mathbb{E}[(a - s)^2]$ is a constant and we denote it by K. Therefore, for some $\epsilon > 0$, if $\mu < \rho$, for large enough value of q, we have

$$\mathbb{E}[q^2(k+1) - q^2(k) | q(k) = q] \leq 2q(\mu - \rho) + K \leq -\epsilon,$$

which implies the stability.

A.2 Proof of Lemma 1

In order to find the optimum distribution, P_X, the optimization problem could be written as follows:

$$\begin{aligned} &\max_{P_X \geq 0} \log_2 e \sum_{i=0}^{k} P_X(i) \ln(\frac{1}{P_X(i)}) \\ &\qquad s.t. \quad \begin{cases} \sum_{i=0}^{k} i P_X(i) = \mathbb{E}[X] = k\gamma \\ \sum_{i=0}^{k} P_X(i) = 1 \end{cases} \end{aligned} \tag{A.1}$$

which could be solved using the Lagrange multipliers method. The Lagrangian function would be as follows:

$$\sum_{i=0}^{k} P_X(i) \ln(\frac{1}{P_X(i)}) + \lambda(\sum_{i=0}^{k} i P_X(i) - k\gamma) + \rho(\sum_{i=0}^{k} P_X(i) - 1).$$

Setting the derivative with respect to $P_X(i)$ equal to zero, we get $\ln(\frac{1}{P_X(i)}) - 1 + i\lambda + \rho = 0$, which implies that

$$P_X(i) = e^{\rho - 1} \cdot e^{i\lambda}. \tag{A.2}$$

Also, from the second constraint we have

$$\sum_{i=0}^{k} e^{\rho-1} \cdot e^{i\lambda} = 1 \Rightarrow e^{\rho-1} = \frac{1}{\sum_{i=0}^{k} e^{i\lambda}}. \tag{A.3}$$

Combining (A.2) and (A.3), we have:

$$P_X(i) = \frac{e^{i\lambda}}{\sum_{i=0}^{k} e^{i\lambda}},$$

which is the tilted distribution of U_k with parameter λ.

In order to calculate λ, from the first constraint:

$$k\gamma = \sum_{i=0}^{k} iP_X(i) = \sum_{i=0}^{k} i\frac{e^{i\lambda}}{\sum_{i=0}^{k} e^{i\lambda}} = \frac{\sum_{i=0}^{k} ie^{i\lambda}}{\sum_{i=0}^{k} e^{i\lambda}} = \frac{\mathbb{E}[U_k e^{U_k\lambda}]}{\mathbb{E}[e^{U_k\lambda}]} = \frac{d}{d\lambda}(\ln \mathbb{E}[e^{U_k\lambda}])$$

$$= \psi'_{U_k}(\lambda).$$

A.3 Proof of Lemma 3

First we note that

$$\begin{aligned}\tilde{H}(\gamma, \frac{1}{k}) &= \frac{1}{k}[\log_2(k+1) - \psi^*_{U_k}(k\gamma)\log_2 e] \\ &= [\frac{1}{k}\log_2(k+1) - \frac{1}{k}\sup_{\lambda}\{k\gamma\lambda - \log(\frac{\sum_{i=0}^{k} e^{i\lambda}}{k+1})\}\log_2 e] \\ &= -\sup_{\lambda}\{\gamma\lambda\log_2 e - \frac{1}{k}\log_2(\sum_{i=0}^{k} e^{i\lambda})\}.\end{aligned}$$

Therefore, if we can show that for any given λ the function $h(\gamma, \frac{1}{k}) = \gamma\lambda\log_2 e - \frac{1}{k}\log_2(\sum_{i=0}^{k} e^{i\lambda})$ is convex, then since the supremum of convex functions is convex, we can conclude the desired concavity of the function $\tilde{H}(\cdot,\cdot)$. Noting that $\frac{1}{k}\log_2(\sum_{i=0}^{k} e^{i\lambda}) = \frac{1}{k}\log_2(\frac{1-e^{(k+1)\lambda}}{1-e^{\lambda}})$, to prove the convexity of $h(\cdot,\cdot)$, it suffices to prove that the function $g(x) = x\log(\frac{1-e^{(\frac{1}{x}+1)\lambda}}{1-e^{\lambda}})$, $0 < x \leq 1$, is concave. This is true from the concavity of the function $\hat{g}(x) = \log(\frac{1-e^{(x+1)\lambda}}{1-e^{\lambda}})$, and the fact that for any function f, $xf(\frac{1}{x})$ is concave if $f(x)$ is concave. The concavity of the function $\hat{g}(\cdot)$ can be easily seen by taking its second derivative.

A.4 Proof of Lemma 4

For a given γ and support set $\{0, 1, ..., k\}$, suppose the distribution P_X^* is defined over $\{0, 1, ..., k\}$ and has mean $\mathbb{E}_{P^*}[X] = k\gamma$ and

$$\sup_{\substack{X \in \{0,1,...,k\} \\ \mathbb{E}[X]=k\gamma}} \frac{1}{k} H(X) = \frac{1}{k} H(P_X^*).$$

Define distribution Q_X as follows:

$$Q_X(i) = P_X^*(k - i) \qquad 0 \leq i \leq k.$$

Therefore the entropy of Q_X will be the same as the entropy of P_X^* and we have

$$\begin{aligned}\mathbb{E}_Q[X] &= \sum_{i=0}^{k} i Q_X(i) = \sum_{i=0}^{k} i P_X^*(k-i) = -\sum_{i=0}^{k} (-k + (k-i)) P_X^*(k-i) \\ &= k - \sum_{i=0}^{k} (k-i) P_X^*(k-i) = k - k\gamma = k(1-\gamma).\end{aligned}$$

Hence, we have

$$\begin{aligned}\tilde{H}(\gamma, \frac{1}{k}) &= \sup_{\substack{X \in \{0,1,...,k\} \\ \mathbb{E}[X]=k\gamma}} \frac{1}{k} H(X) = \frac{1}{k} H(P_X^*) = \frac{1}{k} H(Q_X) \leq \sup_{\substack{X \in \{0,1,...,k\} \\ \mathbb{E}[X]=k(1-\gamma)}} \frac{1}{k} H(X) \\ &= \tilde{H}(1-\gamma, \frac{1}{k}).\end{aligned} \tag{A.4}$$

Similarly, suppose for the distribution Q_X^*, defined over $\{0, 1, ..., k\}$ and with mean $\mathbb{E}_{Q^*}[X] = k(1-\gamma)$

$$\sup_{\substack{X \in \{0,1,...,k\} \\ \mathbb{E}[X]=k(1-\gamma)}} \frac{1}{k} H(X) = \frac{1}{k} H(Q_X^*).$$

Define distribution P_X as follows:

$$P_X(i) = Q_X^*(k - i) \qquad 0 \leq i \leq k.$$

Therefore the entropy of P_X will be the same as the entropy of Q_X^* and we

have

$$\mathbb{E}_P[X] = \sum_{i=0}^{k} iP_X(i) = \sum_{i=0}^{k} iQ_X^*(k-i) = -\sum_{i=0}^{k}(-k+(k-i))Q_X^*(k-i)$$
$$= k - \sum_{i=0}^{k}(k-i)Q_X^*(k-i) = k - k(1-\gamma) = k\gamma.$$

Hence, we have

$$\tilde{H}(1-\gamma, \frac{1}{k}) = \sup_{\substack{X\in\{0,1,\ldots,k\} \\ \mathbb{E}[X]=k(1-\gamma)}} \frac{1}{k}H(X) = \frac{1}{k}H(Q_X^*) = \frac{1}{k}H(P_X) \leq \sup_{\substack{X\in\{0,1,\ldots,k\} \\ \mathbb{E}[X]=k\gamma}} \frac{1}{k}H(X)$$
$$= \tilde{H}(\gamma, \frac{1}{k}). \tag{A.5}$$

Comparing (A.4) and (A.5) gives the desired result.

APPENDIX B

PROOFS FOR CHAPTER 3

B.1 Proof of Lemma 7

$$\begin{aligned}
\tilde{I}_{r_p}(\gamma, \frac{1}{k}) &= \sup_{\substack{X\in\{0,1,\ldots,k\}\\ \mathbb{E}[X]=k\gamma}} \frac{1}{k} I_{r_p}(X;Y) \\
&= \sup_{\substack{X\in\{0,1,\ldots,k\}\\ \mathbb{E}[X]=k\gamma}} \frac{1}{k}[H_{r_p}(Y) - H_{r_p}(Y|X)] \\
&= \sup_{\substack{X\in\{0,1,\ldots,k\}\\ \mathbb{E}[X]=k\gamma}} \frac{1}{k}[H_{r_p}(Y) - \sum_{x=0}^{k} P_X(x) H_{r_p}(Y|X=x)] \\
&\overset{(a)}{=} \sup_{\substack{X\in\{0,1,\ldots,k\}\\ \mathbb{E}[X]=k\gamma}} \frac{1}{k}[H_{r_p}(Y) - \sum_{x=0}^{k} P_X(x) H(Bin(k, r_p))] \\
&= \sup_{\substack{X\in\{0,1,\ldots,k\}\\ \mathbb{E}[X]=k\gamma}} \frac{1}{k}[H_{r_p}(Y) - H(Bin(k, r_p))] \\
&= \sup_{\substack{X\in\{0,1,\ldots,k\}\\ \mathbb{E}[X]=k\gamma}} \frac{1}{k} H_{r_p}(Y) - \frac{1}{k} H(Bin(k, r_p)) \\
&= \frac{1}{k}\check{H}_{r_p}(\gamma, \frac{1}{k}) - \frac{1}{k} H(Bin(k, r_p)),
\end{aligned}$$

where (a) follows from (3.1).

B.2 Proof of Lemma 8

We first prove that the function $\tilde{I}(\cdot,\cdot)$ is concave in its first argument: Let $P^*_{X_1}$ and $P^*_{X_3}$ be the optimum distributions resulted from optimization problem (3.6) for parameters $(\gamma_1, \frac{1}{k})$ and $(\gamma_3, \frac{1}{k})$, respectively. Therefore for any $0 \leq$

$\alpha \leq 1$,

$$
\begin{aligned}
&\alpha \tilde{I}_{r_p}(\gamma_1, \frac{1}{k}) + (1-\alpha)\tilde{I}_{r_p}(\gamma_3, \frac{1}{k}) \\
&\overset{(a)}{=} \frac{1}{k}\alpha H(P^*_{X_1} * Bin(k, r_p)) + \frac{1}{k}(1-\alpha)H(P^*_{X_3} * Bin(k, r_p)) - \frac{1}{k}H(Bin(k, r_p)) \\
&\overset{(b)}{\leq} \frac{1}{k}H(\alpha(P^*_{X_1} * Bin(k, r_p)) + (1-\alpha)(P^*_{X_3} * Bin(k, r_p))) - \frac{1}{k}H(Bin(k, r_p)) \\
&\leq \frac{1}{k} \sup_{\substack{X \in \{0,1,\ldots,k\} \\ \mathbb{E}[X] = k(\alpha\gamma_1 + (1-\alpha)\gamma_3)}} H(P_X * Bin(k, r_p)) - \frac{1}{k}H(Bin(k, r_p)) \\
&= \tilde{I}_{r_p}(\alpha\gamma_1 + (1-\alpha)\gamma_3, \frac{1}{k}),
\end{aligned}
$$

where (a) follows from Lemma 7 and (b) follows from the concavity of the entropy function.

Because of the complexity and lack of symmetry and structure in the function $\tilde{I}$, there is no straightforward analytic method for proving its concavity. But, we notice that it suffices to show that for all $2 \leq k \leq \tau_{max} - 1$, and α such that

$$\alpha \frac{1}{k-1} + (1-\alpha)\frac{1}{k+1} = \frac{1}{k}, \tag{B.1}$$

we have

$$\alpha \tilde{I}_{r_p}(\gamma_1, \frac{1}{k-1}) + (1-\alpha)\tilde{I}_{r_p}(\gamma_3, \frac{1}{k+1}) \leq \tilde{I}_{r_p}(\alpha\gamma_1 + (1-\alpha)\gamma_3, \frac{1}{k}). \tag{B.2}$$

From (B.1) we have $\alpha = \dfrac{k-1}{2k}$, hence using Lemma 7, (B.2) reduces to

$$2\check{H}(\gamma_2, \frac{1}{k}) - \check{H}(\gamma_1, \frac{1}{k-1}) - \check{H}(\gamma_3, \frac{1}{k+1}) + f(k, r_p) \geq 0, \tag{B.3}$$

where $f(k, r_p) = H(Bin(k-1, r_p)) + H(Bin(k+1, r_p)) - 2H(Bin(k, r_p))$. Noting that the left-hand side is a Lipschitz continuous function of γ_1, γ_3, and r_p and the fact that k takes finitely many values, the validation of inequality (B.3) can be done numerically.

APPENDIX C

PROOFS FOR CHAPTER 4

C.1 Proof of Stability

As mentioned in Section 4.1, each user has a separate queue. Denote the queue length and the number of packet arrivals at each queue at time slot n by $q_i(n)$ and $a_i(n)$, respectively. Let $\mathbb{E}[a_i(n)] = \lambda_i$ and $\mathbb{E}[a_i^2(n)] < \infty$, where $i \in \{A, B\}$ signifies Alice or Bob. We assume the arrival processes of Alice and Bob are independent of each other and the system state. The system is stable if neither user's queue length grows to infinity in the steady state of the system, as long as the arrivals are in the capacity region of the scheduler. Thus, it suffices to prove that the sum of the queue lengths is finite with probability one, which implies the stability of both queues. We use the Foster-Lyapunov theorem to prove this statement (for more applications of Foster-Lyapunov theorem refer to [25] and the references inside). Denote the sum of the queue lengths with $\hat{q}(n) = q_A(n) + q_B(n)$, and the sum of packet arrivals for Alice and Bob with $\hat{a}(n) = a_A(n) + a_B(n)$. Note that $\mathbb{E}[\hat{a}(n)] = \lambda_A + \lambda_B$ and the second moment of $\hat{a}(n)$ is finite. As long as a task is available in one of the two queues, the round robin scheduler serves a task; that is, the service rate is one packet per time slot. Thus $\hat{q}$ evolves as:

$$\hat{q}(n+1) = (\hat{q}(n) + \hat{a}(n) - 1)_+, \tag{C.1}$$

where $(x)_+ \triangleq \max\{x, 0\}$.

Choose the Lyapunov function $V(\hat{q}(n)) = \frac{\hat{q}^2(n)}{2}$. Note that this choice of Lyapunov function satisfies the requirements of nonnegativity, being equal to zero only at $\hat{q} = 0$, and going to infinity as $\hat{q}$ goes to infinity. We show that the drift of this Lyapunov function is negative outside of a bounded region of the state space, and is positive and finite inside this bounded region, which implies that the system state is positive recurrent.

$$
\begin{aligned}
&\mathbb{E}[V(\hat{q}(n+1)) - V(\hat{q}(n))\big|\hat{q}(n) = q] \\
&\quad = \mathbb{E}[\frac{(q + \hat{a}(n) - 1)_+^2 - q^2}{2}\big|q] \\
&\quad \leq \mathbb{E}[\frac{\hat{a}^2(n) + 1 + 2\hat{a}(n)q - 2q - 2\hat{a}(n)}{2}] \\
&\quad \leq \mathbb{E}[\frac{\hat{a}^2(n) + 1 - 2\hat{a}(n)}{2} - 2q + 2\hat{a}(n)q] \\
&\quad = c - 2q(1 - \lambda_A - \lambda_B),
\end{aligned} \tag{C.2}
$$

where in the last equality, c is a constant because $\hat{a}$ has bounded first and second moments. For $\lambda_A + \lambda_B < 1$, the drift of the Lyapunov function is bounded by the constant c in the bounded set $B = \{\hat{q} \mid \hat{q} \leq \frac{c}{2(1-\lambda_A-\lambda_B)}\}$, and is negative in the complement set, B^c. Therefore, with our queueing structure and round robin scheduler, the system is stable as long as $\lambda_A + \lambda_B < 1$.

C.2 Proof of Theorems 3 and 8

Since Theorem 3 is a special case of Theorem 8 ($\delta = 0$), we will only prove Theorem 8 here. The proof consists of achievability and converse arguments.
Converse: For any (n, M, ϵ)-code we have

$$
\begin{aligned}
\frac{1}{n}\log M &\overset{(a)}{=} \frac{1}{n}H(W) \\
&= \frac{1}{n}I(W;\hat{W}) + \frac{1}{n}H(W|\hat{W}) \\
&\overset{(b)}{\leq} \frac{1}{n}I(W;\hat{W}) + \epsilon_n \\
&\overset{(c)}{\leq} \frac{1}{n}I(X^m;Y^m) + \epsilon_n,
\end{aligned}
$$

where (a) holds because W is a uniform random variable over the message set $\{1, ..., M\}$, (b) follows from Fano's inequality with $\epsilon_n = \frac{1}{n}(H(P_e) + P_e \log_2(M-1))$ and (c) follows from application of data processing inequality to the Markov chain in (4.1).

Since the channel model is memoryless,

$$
I(X^m;Y^m) \leq \sum_{i=1}^{m} I(X_i;Y_i).
$$

Therefore,

$$\begin{aligned}\frac{1}{n}\log M &\leq \sum_{i=1}^{m}\frac{1}{n}I(X_i;Y_i)+\epsilon_n \\ &\leq \sup_{P_X}\frac{m}{n}I(X;Y)+\epsilon_n.\end{aligned} \tag{C.3}$$

Note that

$$\begin{aligned}n &= (1-\delta)pm\times 2+(\delta pm+(1-p)m)\times 1 \\ &= ((1-p)+\delta p+2(1-\delta)p)m,\end{aligned} \tag{C.4}$$

and

$$\begin{aligned}I(X;Y) =& h(Y)-h(Y|X) \\ =& h((1-\delta)p)-ph(\delta).\end{aligned} \tag{C.5}$$

Substituting (C.4) and (C.5) in (C.3), we have

$$\frac{1}{n}\log M \leq \sup_{p\in[0,1]}\frac{h((1-\delta)p)-ph(\delta)}{(1-p)+\delta p+2(1-\delta)p}+\epsilon_n.$$

As $n\to\infty$, $\epsilon_n\to 0$ and we have

$$C\leq \sup_{p\in[0,1]}\frac{h((1-\delta)p)-ph(\delta)}{(1-p)+\delta p+2(1-\delta)p}.$$

Achievability: Fix a Bernoulli distribution P with parameter p^*, where

$$p^* = \arg\sup_{p\in[0,1]}\frac{h((1-\delta)p)-ph(\delta)}{(1-p)+\delta p+2(1-\delta)p},$$

and generate a binary codebook $\mathcal{C}$ containing 2^{mR} length m i.i.d. sequences drawn according to P, where $m=\frac{n}{(1-p)+\delta p+2(1-\delta)p}$.

In order to send a bit '1', Alice sends a packet and then idles for one time slot. To send a bit '0', she just idles for one time slot. Thus, each message on average takes $m\times((1-p)+\delta p+2(1-\delta)p)=n$ time slots to be transmitted. At the same time, Bob keeps his head of the queue always full.

Since this is a discrete memoryless channel, by the standard typicality decoding arguments [22], the error can be kept arbitrary close to zero as long as

$$R\leq \max_{p\in[0,1]} I(X;Y).$$

Consequently,

$$\begin{aligned} C &\geq \frac{\log 2^{m \times \max_{p\in[0,1]} I(X;Y)}}{n} \\ &\geq \max_{p\in[0,1]} \frac{h((1-\delta)p) - ph(\delta)}{(1-p) + \delta p + 2(1-\delta)p}. \end{aligned} \tag{C.6}$$

The achievability and converse complete the proof of the coding theorem. Therefore,

$$C = \sup_{p\in[0,1]} \frac{h(p)}{1+p}.$$

Since the function $h(.)$ is differentiable, to find the optimum point, it suffices to take the derivative with respect to p and set it to zero:

$$\frac{d}{dp}(\frac{h(p)}{1+p}) = \frac{2log(1-p) - log(p)}{(1+p)^2} = 0.$$

Therefore,

$$p = \frac{3-\sqrt{5}}{2}.$$

C.3 Proof of Theorem 4

We show that Algorithm 1 minimizes the sum of costs of codewords which is $2n_1 + n_0$. Note that replacing codeword $\mathcal{W}$ with cost $\eta(\mathcal{W})$ results in two codewords $\overline{\mathcal{W}0}$ and $\overline{\mathcal{W}1}$ with costs $\eta(\mathcal{W})+1$ and $\eta(\mathcal{W})+2$, respectively. As a result, replacing codeword $\mathcal{W}$ with its two children causes additional cost of $\eta(\mathcal{W})+3$, and an additional codeword to the codebook. Therefore, since the added cost is increasing in $\eta(\mathcal{W})$, to obtain the optimal codebook, it suffices to replace the minimum cost codeword by its children.

Suppose Algorithm 1 outputs codebook $\mathcal{C}_1$, but one claims that codebook $\mathcal{C}_2$ resulting from another algorithm is optimum where both $\mathcal{C}_1$ and $\mathcal{C}_2$ have M codewords. We first find the subtree which is common between $\mathcal{C}_1$ and $\mathcal{C}_2$, which implies that two algorithms are equivalent until, say, step m. From that step, all the replacements are different in two algorithms. The first replacement in Algorithm 1 gives a smaller cost (because we assumed to choose the minimum cost replacement). For the next replacement in step $m+1$, Algorithm 1 had the option of the other algorithm's replacement in step m, yet it did not choose that. This means that again a better replacement

was possible. Adding this to the fact that the costs of children of a codeword are greater than its own cost leads to the conclusion that the replacement in step $m+1$ for Algorithm 1 was also a better choice. This reasoning applies to all steps in which two algorithms are different and leads to the conclusion that $\mathcal{C}_2$ cannot be optimum.

C.4 Proof of Theorems 5 and 7

To show that as the number of messages goes to infinity, the information rates of our proposed optimum codebooks resulting from Algorithms 1 and 2 converge to the capacity, we do the following: We prove that the information rate of another non-optimum codebook with rate lower than the rates of both aforementioned codebooks achieves the capacity.

Consider a codebook with fixed-length codewords from the l-th level of the codeword tree. We choose each codeword to have exactly $\lfloor lp \rfloor$ bits '1', where the parameter $p \in [0,1]$ can be selected in a manner to maximize the information rate. Such a codebook consists of $M = \binom{l}{\lfloor lp \rfloor}$ messages, all of which have equal transmission time $2 \times \lfloor lp \rfloor + (l - \lfloor lp \rfloor)$. We show that the information rate of this codebook asymptotically converges to the capacity as l (or equivalently the number of messages) goes to infinity. From Definition 5, we have

$$\begin{aligned}
\sup_p \lim_{n\to\infty} R_p &= \sup_p \lim_{n\to\infty} \frac{log\binom{l}{\lfloor lp \rfloor}}{n} \\
&\overset{(a)}{=} \sup_p \lim_{l\to\infty} \frac{l \cdot h(\frac{\lfloor lp \rfloor}{l}) + o(l)}{l + \lfloor lp \rfloor} \\
&\overset{(b)}{=} \sup_p \frac{h(p)}{1+p} = C,
\end{aligned}$$

where $n = l + \lfloor lp \rfloor$, (a) follows because using Stirling's approximation it can be shown that $log\binom{l}{k} = l \cdot h(\frac{k}{l}) + o(l)$, (b) holds since the entropy function, $h(\cdot)$, is continuous, and the last equality follows from Theorem 3.

C.5 Proof of Theorem 6

It suffices to show that the best rate is contained in the search range of $l = \hat{l}$ to $2\hat{l}$. First, note that if $l < \hat{l}$, then $2^l < M$ which implies that there is an insufficient number of codewords in level l to cover all messages. On the other hand, since $l < \frac{\eta(\mathcal{C}_l)}{M} < 2l$, we have:

$$\frac{\log M}{2l} < R_l < \frac{\log M}{l}.$$

Therefore, for all $l > 2\hat{l}$,

$$R_l < \frac{\log M}{l} < \frac{\log M}{2\hat{l}} < R_{\hat{l}},$$

where R_l is the information rate of the optimum codebook at level l. In other words, for all $l > 2\hat{l}$ the optimum information rate is less than the optimum information rate of $\mathcal{C}_{\hat{l}}$. This implies that there is no need to check any level lower than $\hat{l}$.

REFERENCES

[1] V. Gligor, "Covert channel analysis of trusted systems. a guide to understanding," DTIC, Tech. Rep., 1993.

[2] S. Kadloor, X. Gong, N. Kiyavash, T. Tezcan, and N. Borisov, "Low-cost side channel remote traffic analysis attack in packet networks," in *Communications (ICC), 2010 IEEE International Conference on.* IEEE, 2010, pp. 1–5.

[3] D. Wakabayashi, "Breach complicates Sony's network ambitions," *The Wall Street Journal*, April 28, 2011.

[4] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[5] X. Gong and N. Kiyavash, "Quantifying the information leakage in timing side channels in deterministic work-conserving schedulers," *arXiv preprint arXiv:1403.1276*, 2014.

[6] M. Liberatore and B. N. Levine, "Inferring the source of encrypted http connections," in *Proceedings of the 13th ACM conference on Computer and communications security.* ACM, 2006, pp. 255–263.

[7] D. X. Song, D. Wagner, and X. Tian, "Timing analysis of keystrokes and timing attacks on SSH." in *USENIX Security Symposium*, vol. 2001, 2001.

[8] C. V. Wright, L. Ballard, S. E. Coull, F. Monrose, and G. M. Masson, "Uncovering spoken phrases in encrypted voice over ip conversations," *ACM Transactions on Information and System Security (TISSEC)*, vol. 13, no. 4, p. 35, 2010.

[9] X. Gong, N. Borisov, N. Kiyavash, and N. Schear, "Website detection using remote traffic analysis," in *International Symposium on Privacy Enhancing Technologies Symposium.* Springer, 2012, pp. 58–78.

[10] X. Gong, N. Kiyavash, and P. Venkitasubramaniam, "Information theoretic analysis of side channel information leakage in FCFS schedulers," in *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*. IEEE, 2011, pp. 1255–1259.

[11] S. Kadloor, N. Kiyavash, and P. Venkitasubramaniam, "Mitigating timing based information leakage in shared schedulers," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 1044–1052.

[12] S. Kadloor and N. Kiyavash, "Delay-privacy tradeoff in the design of scheduling policies," *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2557–2573, 2015.

[13] V. Anantharam and S. Verdu, "Bits through queues," *IEEE Transactions on Information Theory*, vol. 42, no. 1, pp. 4–18, 1996.

[14] S. Cabuk, C. E. Brodley, and C. Shields, "Ip covert timing channels: design and detection," in *Proceedings of the 11th ACM Conference on Computer and Communications Security*. ACM, 2004, pp. 178–187.

[15] S. J. Murdoch and S. Lewis, "Embedding covert channels into TCP/IP," in *International Workshop on Information Hiding*. Springer, 2005, pp. 247–261.

[16] D. Llamas, A. Miller, and C. Allison, "An evaluation framework for the analysis of covert channels in the TCP/IP protocol suite." in *ECIW*, 2005, pp. 205–214.

[17] M. H. Kang, I. S. Moskowitz, and D. C. Lee, "A network pump," *IEEE Transactions on Software Engineering*, vol. 22, no. 5, pp. 329–338, 1996.

[18] V. Berk, A. Giani, G. Cybenko, and N. Hanover, "Detection of covert channel encoding in network packet delays," *Rapport Technique TR536, de lUniversité de Dartmouth*, p. 19, 2005.

[19] S. Gianvecchio and H. Wang, "Detecting covert timing channels: an entropy-based approach," in *Proceedings of the 14th ACM conference on Computer and communications security*. ACM, 2007, pp. 307–316.

[20] A. Ghassami, X. Gong, and N. Kiyavash, "Capacity limit of queueing timing channel in shared FCFS schedulers," in *2015 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2015, pp. 789–793.

[21] R. Tahir, M. T. Khan, X. Gong, A. Ahmed, A. Ghassami, H. Kazmi, M. Caesar, F. Zaffar, and N. Kiyavash, "Sneak-peek: High speed covert channels in data center networks," in *IEEE International Conference on Computer Communications (INFOCOM)*. IEEE, 2016.

[22] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley & Sons, 2012.

[23] I. Csiszar and J. Körner, *Information Theory: Coding Theorems for Discrete Memoryless Systems*. Cambridge University Press, 2011.

[24] Y. Polyanskiy and Y. Wu, "Lecture notes on information theory, MIT (6.441), UIUC (ECE 563)," 2014.

[25] Q. Xie, A. Yekkehkhany, and Y. Lu, "Scheduling with multi-level data locality: Throughput and heavy-traffic optimality," in *IEEE International Conference on Computer Communications (INFOCOM)*. IEEE, 2016.