Capacity Estimation of Non-Synchronous Covert Channels

Zhenghong Wang and Ruby B. Lee
Department of Electrical Engineering
Princeton University
{zhenghon,rblee}@princeton.edu

Abstract

Capacity estimation is an important part of covert channel analysis. It measures the severity of a covert channel by estimating the maximum information rate attainable over it. Traditional capacity estimation methods usually calculate the channel capacity based on a synchronous model, assuming that the channel is synchronous or there are mechanisms that can be utilized to synchronize the transmission. The overhead for synchronization is ignored.

In this paper we argue that covert channels in general are non-synchronous and the overhead for synchronization is not negligible. Instead of assuming a synchronous model, we propose to use the deletion-insertion channel as a more general basis of capacity estimation. Capacity estimation is extended to be able to evaluate the overhead for overcoming non-synchronous effects. Our study shows that reliable communication over a non-synchronous channel is still possible even without synchronization mechanisms. Such non-synchronized communications, however, are not as effective as the synchronized ones. The capacity degradation due to the non-synchronous effects is derived. A tight upper bound of the capacity of synchronized channels is also given.

1. Introduction

A covert channel is a communication channel that is neither designed nor intended to transfer information at all [2]. With "legitimate" use of shared resources and operations of a system, it allows leakage of sensitive or private information. Though usually slow, covert channels have been regarded as a serious risk to data security in computer systems and networks. The National Computer Security Center (NCSC) has included the Covert Channel Analysis (CCA) as an important set in its Trusted Computer System Evaluation Criteria (TCSEC).

The severity of a covert channel is often measured in terms of how fast it can transmit information. Millen first established a connection between Shannon's theory of communications and information flow models [5]. The term *capacity* is borrowed from Shannon's theory and is used as a synonym of the maximum information rate that could be achieved over a covert channel. Using Shannon's theory, a communication channel is often modeled as a mapping from the input symbol space to the output symbol space. The capacity is then derived by maximizing the mutual information over the distribution of the input symbol space. Such a model however, implies a synchronous channel model: for each input symbol, the channel always generates an output symbol in response. In other words, a transmitted symbol may be corrupted by noise, but it will never get lost and the receiver will never receive extra symbols. There are also other ways to measure the *capacity* of a covert channel, e.g., the "informal method" described in [3]. Although different methods other than Shannon's information theory are used in these methods, the synchronous property is also assumed either explicitly or implicitly.

Assuming a synchronous model in communication systems is usually not a problem. Most communication systems are designed to avoid symbol loss and/or insertion with little or no overhead. This conclusion is not true for covert channels, however. First, there is usually no handy mechanism available for synchronization since it is not designed for information transfer. Second, in systems where covert channels are a concern, the increasing awareness of covert channels has pushed the designers to make the covert channels harder to exploit. This may make the synchronization problem more severe.

In this paper, we focus our discussion on capacity estimation based on a non-synchronous channel model. We try to answer questions on the existence, capacity and mechanisms for covert communications over non-synchronous channels.

The rest of this paper is organized as follows. Related past work are reviewed in section 2. In section 3, we model non-synchronous channels as deletion-insertion channels. Capacity bounds of deletion-

Zhenghong Wang and Ruby B. Lee, Capacity Estimation of Non-Synchronous Covert Channels, Proceedings of the 2nd International Workshop on Security in Distributed Computing Systems (SDCS), pp.170-176, June 2005.

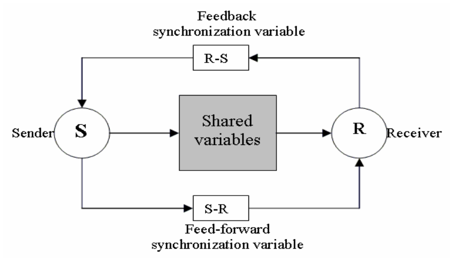


Figure 1. Synchronization mechanism using two synchronization variables

insertion channels are given in section 4. The synchronized form of communication and possible synchronization mechanisms are studied. Capacity degradation due to synchronization overhead is also shown. In section 5 we draw our conclusions.

2. Related work

The notion of covert communication was first introduced in [2]. It was then defined and analyzed in [1][6]-[9]. Currently research in covert channels focuses on four disciplines: covert channel identification, covert channel capacity estimation, covert channel handling and covert channel mitigation.

In measuring the significance of a covert channel, Millen first established a connection between Shannon's theory of communications and information flow models [5]. The term "channel capacity" is borrowed from Shannon's theory which stands for the maximum information rate that could be achieved over a covert channel.

In 1989, Millen modeled an important class of covert channels as finite state machines [15]. The covert channels that are noiseless and have non-uniform transition times are studied. Using Shannon's theory, he derived the channel capacity of such covert channels.

Moskowitz [10] studied a class of covert channels that is discrete, noiseless and memoryless, called the Simple Timing Channels (STC) in 1994. The capacities of such covert channels can be regarded as an upper bound for more complicated channels and may give a worst case scenario. The bounds of capacities of such channels were derived. In 1996, he analyzed a class of covert timing channel, called the timed Z-channel, and showed the bound on its capacity [11].

Comprehensive information and examples about covert channel analysis can be found in Virgil Gligor's Covert Channel Analysis guideline [3] and McHugh's Covert Channel Analysis chapter [4].

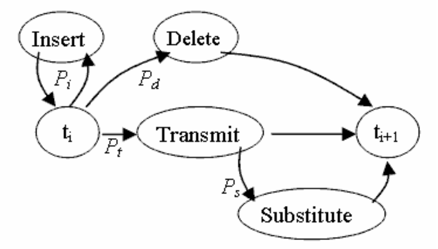


Figure 2. Insertion-Deletion channel with probabilities P_d, P_i, P_t and $P_{s,}$ of deletions, insertions, transmissions and substitutions.

3. Non-synchronous covert channels

3.1. Overview

Covert channels are often very different from normal communication channels. First of all, unlike communication systems where synchronization is often specifically designed for reliable and efficient communication, synchronization mechanisms are usually not available for covert channels. Secondly, in covert channels the communicating parties often have limited or even no control in choosing the proper time to perform an operation, e.g., send a symbol to the channel or sample the channel to receive a symbol, which is not a problem in normal communication channels. Below is an example.

Consider a uniprocessor system where the communicating subjects are processes. To transmit a symbol, the sender has to make a change in the system and the receiver receives it by detecting the change. As there is only one CPU in the system, at any time only one of the two processes can be active. In other words, the sender has to relinquish the CPU after it sends a symbol so that the receiver can get the CPU to read the symbol. In most operating systems, the scheduler determines when and who can gain the CPU. Depending on the scheduling algorithm, it is very likely that the sender is woken up twice without the receiver being able to run in between, or the receiver is woken up twice without the sender being able to run in between. In the former case a symbol is dropped while in the later case an extra symbol is inserted.

Finally, coherent time references are often unavailable in covert channels. Time references are known as key components in exploiting many covert timing channels. Furthermore, as long as the local timers at the two sides of the channel are coherent enough, it is not difficult to synchronize operations performed by the sender and the receiver. Being aware of these facts, high assurance systems have made efforts to remove event sources that can serve as such time references to user processes.

In summary, covert channels are inherently non-synchronous in general, although there may be some exceptions.

3.2. Capacity degradation

Though the operations of sending and receiving symbols are non-synchronous in most covert channels, one may still synchronize the symbol transmission with certain techniques. Figure 1 shows an example. The sender makes a change on the S-R variable once a symbol is sent; the receiver checks the S-R variable and reads the symbol when ready; the receiver then makes a change on the R-S variable to inform the sender; the sender checks the R-S variable and sends the next symbol once the last symbol has been received.

There may be other methods that can maintain the correct order of operations, but in essence they do the same thing: let the sender know if the receiver has read the previous symbol and let the receiver know if a symbol has arrived. With such information, each time when the sender is able to perform an operation it can determine whether a new symbol can be sent. However, due to the non-synchronous nature of the covert channels, it is very likely that the sender finds that the previous symbol has not been read by the receiver and it has to give up the CPU and wait for the next chance. In other words, some time is wasted for waiting and therefore the channel capacity is reduced.

In previous work, the synchronous model excludes this part of the time in symbol transmission. Only the time associated with transmitted symbols is taken into account. In contrast, our method considers such wasted time and gives more accurate estimations. It can be regarded as a more general form of capacity estimation where the methods based on a synchronous model are special cases. Furthermore, unlike traditional methods which calculate a single upper bound of the capacity, our method reflects the non-synchronous behaviors which are determined by the system implementations. Our method can be used to evaluate the effectiveness of candidate system implementations, e.g., the scheduler, in reducing covert channel capacities.

3.3. Deletion-insertion channel

Non-synchronous operations at the two sides of the channel may lead to loss of real symbols and insertion of false symbols. Such a channel can be modeled as a deletion-insertion channel. We adapt the definition in [13] as follows:

Definition 1: A *binary deletion-insertion channel* is a channel with four parameters: P_d, P_i, P_t and P_s, which denote the rates of deletions, insertions, transmissions and substitutions, respectively.

The symbols to be transmitted are imagined entering a queue, waiting to be transmitted by the channel. Each time the channel is used, one of four events occurs: with probability P_d the next queued bit is deleted; with probability P_i an extra bit is inserted; with probability P_t the next queued bit is transmitted, i.e., is received by the receiver, with probability P_s of suffering a substitution error (see Figure 2).

A deletion-insertion channel should not be confused with an erasure channel. In an erasure channel, channel symbols may be corrupted or lost, which is similar to the substitution or deletion in a deletion-insertion channel. However, the receiver of an erasure channel knows exactly which symbols are corrupted or dropped while in a deletion-insertion channel, the receiver knows nothing about any deletion, insertion or substitution (corruption) of symbols. This makes the recovery of a message much harder.

4. Capacity estimation

Our discussions focus on two sets of questions:

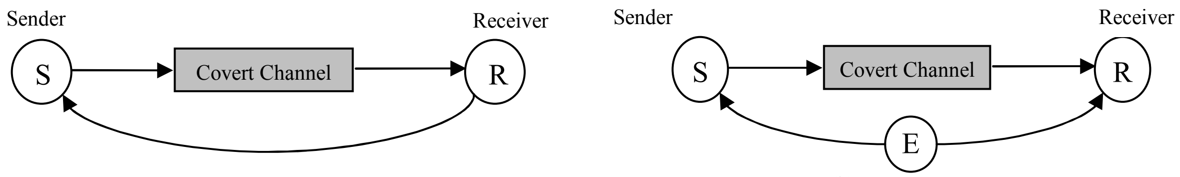
1. **Existence and capacity:** Without any form of synchronization, is reliable communication still possible? If the answer is yes, what's the capacity of such channels?
2. **Construction and capacity:** How can reliable synchronization mechanisms be constructed for non-synchronous covert channels? What is the maximum information rate one can achieve over such channels? Compared with the capacity of an inherently synchronous channel, what is the degradation of information rate due to the non-synchronous effect?

The first question is a theoretical existence or feasibility one, while the second set of questions shows how such a covert channel can actually be constructed.

It is worth noting that the first question indeed is asking if synchronization is always necessary. Previous work all assumes a synchronized form of communication. But it is not clear if it is the only way for reliable communication. In fact, another interesting question is: can a non-synchronous form of communication have higher information rate than the synchronous one as the overhead associated with synchronization is totally avoided?

4.1. Capacity of deletion-insertion channels

Intuitively a channel with symbol insertions and drop-outs is hard to use and not efficient. But as maintaining synchronized communication also introduces overhead, we wish to know how fast the information can be delivered over such channels, compared to the synchronized channels.



(a) Feedback (b) Common Events

Figure 3. Two general synchronization mechanisms

Doburshin [17] first showed that the fundamental theorem of information theory concerning the existence of an upper bound for the transmission rate, for which error probability can be made arbitrarily small, holds. This implies that reliable communication is possible even if no reliable synchronization mechanisms are available. However, explicit expression of the capacity of such a channel is generally not available. A variety of approximations of the capacities and numerical bounds can be found in [18][19]. The accurate capacity of a deletion-insertion channel is still unavailable, according to the state-of-the-art research in this area.

Despite the unavailability of an accurate capacity, we can give an upper bound of the capacity for the purpose of comparison with synchronized channels.

Consider a deletion-insertion channel and an erasure channel which are identical except that in the erasure channel the location of symbol drop-outs and insertions are known. It is not difficult to show that the capacity of the deletion-insertion channel is no greater than the capacity of the erasure channel.

Theorem 1. An upper bound of the capacity of a deletion-insertion channel is the capacity of the erasure channel:

$$C_{max} = N(1-P_d) \tag{1}$$

where N is the number of bits per symbol and P_d is the deletion probability

The derivation of the capacity of a binary erasure channel can be found in many information theory textbooks such as [16] and it is straight-forward to generalize it to non-binary cases.

Note that here we use the term *erasure channel* to refer to the one that is identical to the corresponding deletion-insertion channel except that the locations of symbol insertions/drop-outs are known. In the rest of the paper, unless otherwise specified, we will use erasure channel to refer to this specific erasure channel.

Rather than giving a formal proof of theorem 1, we give a more intuitive explanation. The two channels have the same symbol drop-outs and insertions, but the erasure channel knows more information: it knows which symbols are deleted or inserted. The erasure channel therefore has equal or higher capacity than that of the deletion-insertion channel.

The above capacities are only upper bounds of the channel capacity. They are very hard, if not impossible, to achieve in practice. Using existing coding schemes such as convolutional code and watermark code, some work [12-14] have shown reliable communication over such channels. However, they all showed that the capacity is quite low and in practice sophisticated coding techniques are required.

4.2. Synchronization and capacity estimations

Unlike our first question on theoretical existence and capacity, past literature does not provide any clues to answering our second set of questions. We present new work on how synchronization mechanisms can be constructed for inherently non-synchronous covert channels and what capacity can be achieved.

We will consider two general ways to achieve synchronization: using feedback or using common events, as shown in Figure 3.

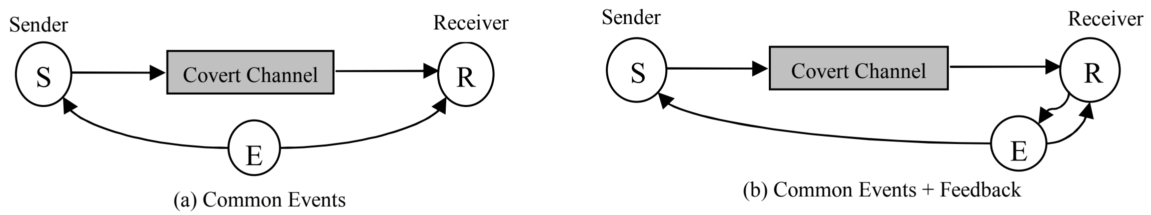
We assume that the feedback path and the two paths from the event source E to the sender S and the receiver R are perfect. This simplifies the analysis, and is also a requirement for deriving the maximum information rate. To focus on the synchronization problem, we assume that the channel is noiseless. Let p_d and p_i denote the probability of deletion and insertion respectively.

4.2.1. Channels with feedback

We now show that the capacity of a channel with deletions can achieve the capacity of an erasure channel by utilizing feedback. We then extend the result to a channel with insertions.

Theorem 2. The upper bound of the capacity of a deletion channel with perfect feedback is the capacity of the erasure channel.

Proof: Consider a deletion channel with a deletion probability p_d and its corresponding erasure channel. Add perfect feedback path to both of them. Since the erasure channel knows where the symbol drop-outs occurs which the deletion channel does not know, the erasure channel knows more information than the deletion channel. Therefore the erasure channel with feedback will gain equal or higher capacity than the deletion channel with feedback. Since an erasure



(a) Common Events

(b) Common Events + Feedback

Figure 4. Using common events won't get better capacity than using feedback

channel is a memoryless channel and it is well known that adding feedback will not increase the capacity of a memoryless channel [16], the upper bound of the capacity of a deletion channel with perfect feedback is the capacity of the erasure channel. □

In theorem 2 we only show an upper bound of the capacity, we now show that the bound is tight.

Theorem 3. The capacity of a deletion channel with perfect feedback equals the capacity of the erasure channel.

Proof: Here we construct a protocol by which the capacity of the erasure channel can be achieved. The protocol is as follows: let the receiver notify the sender via the feedback path once it receives a symbol. The sender will keep resending the symbol until it knows that the symbol has been received. Therefore no dropouts will occur. While the probability of deletion is p_d, a symbol gets through with probability of 1- p_d, therefore the effective information rate is $N(1-p_d)$, which is the capacity of an erasure channel with an erasure probability p_d. Since the upper bound of the capacity can be achieved, it is the actual capacity. □

When symbol insertions are present in the channel, a theorem similar to theorem 2 can be proved. We first define an extended erasure channel as follows:

Definition 2: An *extended erasure channel* is a channel where symbols may be inserted and/or dropped but the locations of all insertions and dropouts are known.

Theorem 4. The upper bound of the capacity of a deletion-insertion channel with perfect feed back is the capacity of the equivalent extended erasure channel, i.e., $C_{upper\text{-}bound} = N(1\text{-}P_d)$.

The proof is similar to that for Theorem 2.

A lower bound of the capacity can also be derived with a constructive protocol. It can be shown that under certain conditions, this lower bound and the upper bound shown in Theorem 4 asymptotically converge.

Theorem 5. A lower bound of the capacity of a deletion-insertion channel with perfect feed back is:

$$C_{lower-bound} = \frac{1-P_d}{1-P_i} C_{conv} \tag{2}$$

where

$$C_{conv} = N - \alpha p_i \log_2(2^N - 1) - H(\alpha p_i) \tag{3}$$

$$\alpha = 1 - \tfrac{1}{2^N} \tag{4}$$

$$H(p) = -p\log_2 p - (1-p)\log_2(1-p) \tag{5}$$

N is the number of bits contained in each symbol.

A simple protocol is constructed to prove Theorem 5. The above capacity can be achieved using this protocol and therefore is a lower-bound of the actual capacity. A detailed proof can be found in Appendix A.

To show the asymptotical convergence, let $P_i = P_d$ and N → ∞, we then have

$$C_{lower-bound} = C_{conv} \approx N(1-P_d) - H(P_d) \tag{6}$$

$$\lim_{N\to\infty} \frac{C_{lower-bound}}{C_{lupper-bound}} = \frac{N(1-P_d) - H(P_d)}{N(1-P_d)} = 1 \tag{7}$$

4.2.2. Channels with common event source

There may be several ways to exploit a common event E for synchronization. For example, E can be a self-incrementing counter which serves as a common clock for the sender and receiver. However, as we show below, exploiting E will not get higher capacity than using a feedback path in general.

If one more path from R to E is added, as shown in Figure 4(b), E may gain more information. Therefore an equal or higher information rate may be achieved than without the added path. In the best case, E and R communicate with each other without any overhead, i.e., they indeed can be regarded as one single party and such a configuration actually becomes the synchronization method using feedback. Therefore a similar system using feedback will get equal or better performance for channel capacity.

4.3. Remarks

We have answered the two sets of questions we posed: (1) Reliable communication over non-synchronous channels without synchronization is possible, but it is not as effective as synchronized communication and requires complicated coding schemes. (2) The capacity degradation due to non-synchronous effects is roughly proportional to P_d, the probability of deletions.

According to the above discussion, with a good feedback path, synchronization is not a problem for a covert channel in general. Furthermore, with the help of the feedback, the theoretical capacity of the channel can be practically achieved using a very simple protocol. This has interesting implications for a multi-level security (MLS) system. Since the legal information flow (from low to high) can serve as a perfect feedback path, one may always exploit it to achieve the channel capacity. In other words, covert channels in MLS systems are relatively easy to exploit in general and tend to be fast.

Note that the capacity (1-P_d) we derived above is not a physical information rate. It is a relative ratio of the physical capacity estimated using traditional methods. Therefore for a given covert channel, one could first use traditional methods to estimate the physical capacity C. The probability of deletion P_d should then be estimated. The real capacity can then be estimated as C(1-P_d).

Note also that the capacity degradation modeled in our method is independent of the synchronization mechanisms used and does not include any specific overhead introduced by such mechanisms. Such degradation is inherent due to the non-synchronous nature of operations. It is unavoidable even if efficient mechanisms are deployed.

Finally, although our results are derived in the context of capacity estimation of covert channels, it may provide meaningful insights to researchers in other areas. Recently some ongoing work [20] in the communication community also shows interest in the capacity bounds of channels with asynchronous behaviors. Although the problems and models are different, similar insights may apply. It would be interesting to study the connections in our future work.

5. Conclusions

In this paper we considered the effect of non-synchronous operations in the capacity estimation of covert channels. We argue that covert channels in general are non-synchronous and the time taken for synchronization is not negligible. To model such effect in capacity estimation, we propose using the deletion-insertion channel as a more general basis for covert channel capacity estimations. Our study shows that reliable communication over a non-synchronous channel is still possible, though such non-synchronized communications are not as effective as the synchronized ones. In the case of synchronized communication, we show that the capacity degradation due to the non-synchronous effects is roughly proportional to the probability of synchronization errors.

6. References

[1] R.A. Kemmerer, "Shared Resource Matrix Methodology: An Approach to Identifying storage and Timing Channels," *ACM Transactions on Computer Systems*, vol. 1, issue 3, pp. 256-277, August 1983.

[2] B.W. Lampson, "A Note on the Confinement Problem," *Communications of the ACM*, vol. 16, issue 10, pp. 613-615, October 1973.

[3] National Computer Security Center, "A Guide to Understanding Covert Channel Analysis of Trusted Systems," NCSC-TG-30, November 1993, available at http://www.radium.ncsc.mil/tpep/library/rainbow/.

[4] John McHugh, "Covert Channel Analysis: A Chapter of the *Handbook for the Computer Security Certification of Trusted Systems*," December 1995, available at http://chacs.nrl.navy.mil/publications/handbook/.

[5] J.K. Millen, "Covert Channel Capacity," *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pp. 60-66, April 1987.

[6] S.B. Lipner, "A Comment on the Confinement Problem," *Operating Systems Review*, vol. 9, issue 5, pp. 192-196, November 1975.

[7] M. Schaefer, B. Gold, R. Linde, and J. Scheid, "Program Confinement in KVM/370," *Proceedings of the 1977 Annual ACM Conference*, pp. 404-410, October 1977.

[8] J.C. Huskamp, "Covert Communication Channels in Timesharing Systems," Technical Report UCB-CS-78-02, Ph.D. Thesis, University of California, Berkeley, CA, 1978.

[9] D.E. Denning, "Cryptography and Data Security," Addison-Wesley, Reading, Massachusetts, 1983.

[10] I.S. Moskowitz and A.R. Miller, "Simple Timing Channels," *Proceedings of IEEE Computer Symposium on Research in Security and Privacy*, pp. 56-64, May 1994.

[11] I.S. Moskowitz, S.J. Greenwald, and M.H. Kang, "An Analysis of the Timed-Z Channel," *Proceedings of IEEE Computer Symposium on Security and Privacy*, pp. 2-11, May 1996.

[12] K.Sh. Zigangirov, "Sequential Decoding for A Binary Channel with Drop Outs and Insertions," *Problemy Peredachi Informatsii*, vol. 5, issue 2, pp. 22-30, 1969.

[13] M.C. Davey and D.J.C. Mackey, "Reliable Communication over Channels with Insertions, Deletions, and Substitutions," *IEEE Trans. on Information Theory*, vol. 47, no.2, pp. 687-698, February 2001.

[14] Dave Leigh, "Capacity of Insertion and Deletion Channels," Project Report, 2001, available at http://www.inference.phy.cam.ac.uk/is/papers/

[15] J.K. Millen, "Finite-State Noiseless Covert Channels," *Proceedings of the Computer Security Foundations Workshop II*, pp. 81-86, June 1989.

[16] T. Cover and J. Thomas, "*Elements of Information Theory*," John Wiley & Sons Inc., New York, 1991.

[17] R.L. Dobурshin, “Shannon’s Theorems for Channels with Synchronization Errors,” *Problemy Peredachi Informatsii*, vol.3, No.4, pp.18-36, 1967

[18] N.D. Vvedenskaya and R.L. Dobурshin, “The Computation on a Computer of The Channel Capacity of a Line with Symbol Drop-out,” *Problemy Peredachi Informatsii*, vol.4, No.3, pp.92-95, 1968.

[19] A.S. Dolgopolov, “Capacity Bounds for a Channel with Synchronization Errors,” *Problemy Peredachi Informatsii*, vol.26, No.2, pp.27-37, 1990.

[20] J. Luo, A. Ephremides, "On the Throughput, Capacity and Stability Regions of Random Multiple Access", submitted to IEEE Trans. on Information Theory, February 2005.

Appendix A: A Proof of Theorem 5

A protocol is constructed based on which a capacity can be derived. The receiver keeps a counter that records the number of symbols that it has received. Each time when the receiver gets the chance to perform an operation, it reads the channel and believes that a symbol is received. It then updates the counter and informs the sender how many symbols it has received. At the sender side, the sender keeps a counter that records how many symbols of the message have been sent or skipped. Each time when the sender gets a chance to perform an operation, it checks the number of symbols the receiver has received. If the number is smaller than its own counter, it means that a symbol sent in the sender’s last operation has not been received by the receiver. The sender then does nothing and waits for the next opportunity. If the two numbers are equal, it means that last symbol has been received. The sender then sends the next symbol and updates its own counter. If the number is larger than the sender’s counter, it means that some symbols have been inserted. To synchronize the transmission, the sender skips some symbols in the message so that the next symbol to be sent will appear in the same location in the received message at the receiver’s side as in the original message. The sender then sends this symbol and updates the counter with the location of the symbol next to the one that was just sent.

Using such a protocol, the sender can always ensure that the number of symbols received by the receiver equals the number of symbols it sent. For symbol insertions, since the sender skips the same number of symbols to be sent, in the received symbol sequence the skipped symbols are replaced by those inserted ones. Also, since the sender will not send a new symbol until at least one symbol is received, no symbol deletion can occur. Therefore the resulting channel is a synchronous channel, but with some symbols replaced.

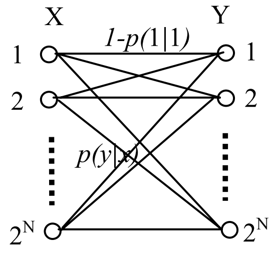


Figure 5. Converted channel model

To calculate the capacity of such a channel, Shannon’s theory can be used. For the sake of simplicity, we assume that the data channel is noiseless, i.e., $P_s = 0$. We also assume the channel is memoryless. Figure 5 shows the channel model. It is indeed an M-ary symmetric DMC (discrete memoryless channel) with the following transition probabilities:

$$P(y \mid x) = \begin{cases} 1-(1-\frac{1}{2^N})p_i & when\ x = y \\ \frac{1}{2^N}\, p_i & when\ x \neq y \end{cases} \qquad (1)$$

where N is the number of bits of the data channel and P_i is the insertion probability. It is straight forward to get the following capacity:

$$C_{conv} = N - \alpha p_i \log_2(2^N - 1) - H(\alpha p_i) \qquad (2)$$

where

$$\alpha = 1 - \frac{1}{2^N} \qquad (3)$$

$$H(p) = -p\log_2 p - (1-p)\log_2(1-p) \qquad (4)$$

when N is large,

$$C_{conv} \approx N(1-p_i) - H(p_i) \qquad (5)$$

Note that the above calculation is based on the synchronous model where the time wasted for waiting is not taken into account and the time for skipped symbols indeed should be 0, we need to adjust the above results with a coefficient. It is not difficult to show that this coefficient should be (1- P_d)/(1- P_i), therefore the actual capacity should be

$$C = \frac{1 - P_d}{1 - P_i} C_{conv} \qquad (6)$$

where C_{conv} is given in (2) and (5). As the above capacity can be achieved with a real protocol, it is the lower bound of the actual capacity.