

A Covert Channel in MAC Protocols Based on Splitting Algorithms

Invited Paper

Song Li, and Anthony Ephremides
ECE Dept. and the Institute for System Research
University of Maryland
College Park, MD, 20740
{lis, [etony](mailto:etony@glue.umd.edu)}@glue.umd.edu

Abstract—We investigate a covert channel implemented on top of MAC protocols that are based on splitting algorithms. Covert information is embedded in nodes' splitting decisions. The covert channel can operate in three modes. The conservative mode is the safest in the sense that use of the covert channel is undetectable. The aggressive mode generates best throughput, but is more vulnerable to detection. A strategic mode is also available which allows the covert users to take a tradeoff between detectability and covert capacity. Simulation shows that the covert throughput ranges from 0 to as high as 0.3 bits per slot, depending on various parameters. It is easy to implement and very difficult to detect.

I. INTRODUCTION

Covert channels, introduced in [1], are concealed communication paths whose usage or even the very existence is not expected in the original design of a communication system. Covert communication happens when one user intentionally manipulates and embeds information into some properties of the system in such a way that the extra information can be detected by specific designated users in the system. A covert channel, to be useful, does not need high bit rate or high capacity or even low loss rate. It is generally satisfactory if it can transmit a few bits per second with some positive probability. For example, only a few bits are needed to disclose the time of an attack or the PIN number of a personal bank account. However, a covert channel must be difficult to detect. This is a paramount requirement.

Most of the past studies on covert channels in computer networks share the idea established in [3] which is to implement covert channels by utilizing special packet portions that are not used for normal transmissions or portions that are optional fields to be set as needed. Covert

information is generally embedded in those special packet portions.

This paper investigates a covert channel where the covert information is not encoded anywhere in the packet, but implied in the protocol operation procedures. It does not affect the normal network operation, and thus is very difficult to detect. What is more, the covert channel can operate in three different modes. The conservative mode is the safest in the sense that use of the covert channel is undetectable. The aggressive mode generates best throughput, but is more vulnerable to detection. A strategic mode is also available which allows the covert users to take a tradeoff between detectability and covert capacity.

Performance of this covert channel depends on various factors, including the network size, traffic rate, the operation mode, and the way the splitting algorithms are actually implemented. Simulation results show that the covert throughput ranges from 0 to as high as 0.3 bits per slot. Measures are needed for protection against this covert threat

The remainder of this paper is organized as follows. Section II introduces the basic idea of splitting algorithm and describes three versions of its implementation. Section III describes the covert transmission scheme and its three operation modes. Detectability of this covert channel is also briefly discussed in this section. Section IV presents the results of performance evaluation. Section V covers some related work. In section VI, we draw some conclusions and point to future work.

II. SPLITTING ALGORITHMS

One class of channel access algorithms is called splitting algorithm. Assume the channel is slotted with instant feedback of '*i(dle)*', '*s(uccess)*', and '*c(ollision)*'. Collision happens when two or more nodes transmit in the same slot. The basic idea of a splitting algorithm is to divide the

Manuscript received Sept 15, 2004. This material is based upon work supported by, or in part by, the U.S. Army Research Laboratory and the U.S. Army Research Office under contract/grant number DAAD19-01-1-0494.)

collided nodes into smaller subsets and these subsets retransmit in order. Successive collisions result in the nodes splitting into smaller subsets, thus the probability of collision happening again is reduced.

Splitting algorithms can differ from each other in several aspects, including the number of subsets it splits into, what to do with new arrivals during a collision resolution period (CRP), and etc. We start from its basic form, and introduce two of the modified versions based on that.

A. The Basic Binary Tree Algorithm

In the basic binary tree algorithm, after a collision, the involved nodes decide independently to join one of the two subsets with equal probability. The transmissions among the two subsets are resolved in turns. Packets that arrived during the current CRP are blocked and wait for transmission until the new CRP starts. Figure 1 shows an example of two nodes collided.

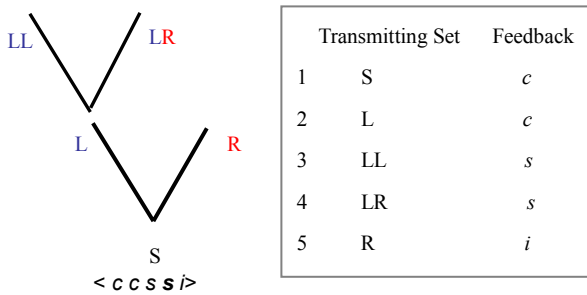


Figure 1. The Basic Binary Tree Algorithm

B. Improvement 1

Improvement 1 consists of two parts. The first part is based on the observation that if an idle is observed in the first subset's transmission, it is guaranteed that the second subset is going to collide. So we can save a slot by splitting the second set directly.

Another observation is that if the first subset collided, then the second subset is expected to contain a small number of packets. This has motivated the second part of this improvement: instead of coming back to resolve the second subset, we can put it into the waiting group and work on it in the next CRP.

C. Improvement 2

In the basic binary tree algorithm, during a CRP, the new packet arrivals are blocked and get to be transmitted in the beginning of the next CRP. In the event that the previous CRP has taken a very long time, the number of waiting packets is expected to be large. They are going to continue to collide with each other before they are split into small enough subsets.

One possible solution is to directly split this waiting set, i.e. root of the tree, into multiple j subsets. By estimating how

many nodes are in the root set, the number j is chosen such that the expected number of packets per subset is slightly greater than one.

D. Unblock algorithms

All the splitting algorithms described above require every node to monitor the channel feedback and to keep track of each collision resolution procedure. This is undesirable when receivers are turned off, especially in wireless networks to save battery power.

One way to avoid this disadvantage is to add new arrivals right into the next subset to process. The new packets are transmitted immediately in the next slot after their arrival. In this way, only currently transmitting nodes need to track the collision resolution procedure. Since the new arrivals are no longer blocked, this type of algorithm is called unblocked stack algorithms.

III. COVERT COMMUNICATION

Covert transmission can be realized via controlling the splitting procedure. Upon collision, a covert transmitter decides which subset to join according to the covert symbol to transmit. For example, '1' is transmitted if it joins the left subset, and '0' is transmitted if it joins the right subset. The covert receiver only needs to passively track the collision resolution procedure. When the covert receiver detects a successful transmission from the covert transmitter, the covert receiver can retrieve the past splitting decisions made by the covert transmitter. Figure 2 demonstrates how two covert bits "10" are transmitted in a collision resolution period. The covert transmitter collided with the other node twice, and succeeded in the fourth slot.

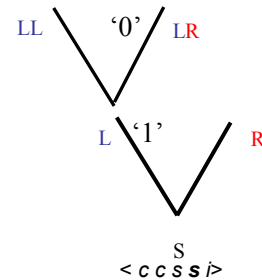


Figure 2. Example Covert Transmission

Denote the covert transmitter as CT and covert receiver as CR. A covert channel can be built based on the following assumptions:

- 1) The starting set of colliding nodes is named S_0 ;
- 2) Each time a collision happens, the involved nodes split into s subsets, S_1, S_2, \dots, S_s ;
- 3) The CT and CR share a codebook $\{S_1, S_2, \dots, S_s\}$;
- 4) The CT can choose to join one of the s subsets when its transmission failed due to collision;

- 5) The CR passively tracks the collision resolution procedure, and retrieves the covert information from CT's partition decisions

Assumption 4 allows the CT to embed covert information into its partition decisions. Upon splitting, instead of randomly choosing a subset to join, the CT chooses the subset index to be the next covert symbol to transmit. Figure 3 shows the basic procedure of covert transmission.

Assumption 5 indicates that the CR needs only to passively monitor the channel in order to receive the covert information. The CR keeps track of which subset is transmitting in the current slot by observing the channel feedbacks. When a successful transmission from the CT is observed, the position of the CT on the splitting tree records exactly its splitting history. For example, if the CT has transmitted as the subset $S_0S_iS_jS_k$, the CR can decide that the CT has joined subset S_i , S_j , and S_k upon three partitions, and thus three covert symbols, S_i , S_j , and S_k , have been transmitted.

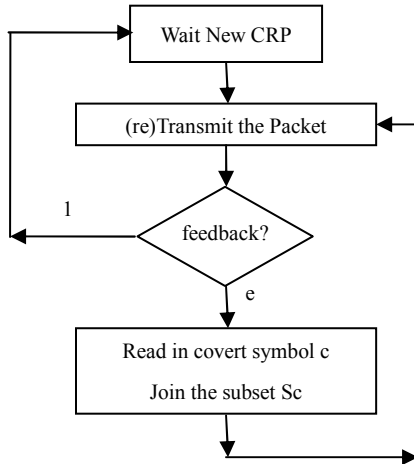


Figure 3. Flow Chart of the Basic Covert Transmission Procedure

With slight modification, the CT can make more aggressive covert transmissions and improve the covert throughput, at the cost of been caught of using the covert channel. Three different operation modes are implemented, which allow the CT and CR to tradeoff between the covert throughput and their undetectability.

A. Conservative Mode

In the conservative mode, the CT transmits only when it has a valid packet to send. The CT is different from the other nodes only in the way that it makes its splitting decisions according to the covert source instead of randomly. It is difficult to catch the CT, especially when the covert symbols are equally probable so that the CT behavior appears to be exactly the same as the others.

One limitation is that occurrence of the covert transmission depends on packet arrivals. If no new packet has arrived at

the CT before the start of a new CRP, the CT will not be able to do covert transmission in that CRP.

B. Aggressive Mode

The aggressive mode solves this problem by allowing the CT to create dummy packets such that the CT can participate in each and every CRP. Under this mode, the CT always transmits in the first slot after the end of the current CRP.

An obvious shortcoming of this mode is the abnormal activeness of the CT, which could result in detection of the use of the covert channel and identifying the CT. Note that the CR always remains safe from detection.

C. Strategic Mode

The aggressive mode is mostly useful when the traffic is light. We observed that the covert transmitter's effort is wasted if nobody collides with it. But the CT still suffers from the high risk of exposure by transmitting those packets. It would be more risk-throughput efficient if collisions can be guaranteed for the dummy packet transmission.

A simple strategy can be taken by letting the CT jump into a CRP when it observes collision in the first slot of that CRP. The CT simply pretends that it is among the set of collided nodes S_0 .

More complicated strategies can provide intermediate covert transmission rate by adapting the dummy packet generation rate according to the CT's eagerness to transmit and its willingness to get exposed.

Before evaluating this covert channel through simulation, we would like to summarize some of its properties first.

IV. DISCUSSION

A. Performance

This covert channel is error free. With correct channel feedback, the CR always can successfully track the CRP.

Throughput of the covert channel depends on multiple factors. First, the CT has to transmit some data packet and participate in the CRP. At most one packet can be transmitted per CRP. Second, only when this packet collides with other nodes' transmission can the CT embeds one covert symbol into its splitting decisions. The number of covert symbols sent in one CRP is equal to the number of collisions that the CT has met before the successful transmission. Finally, the transmission rate also depends on the length of the CRP. Not all of the slots are devoted to solve the collisions that involve the CT. For the rest of the time, the CT is either waiting for its turn to transmit, or waiting for the end of the current CRP. The covert throughput can be roughly expressed as:

$$\text{throughput} \approx f_{CRP} \cdot \frac{c_{CRP}}{l_{CRP}} \cdot \log_2 s \quad (1)$$

- f_{CRP} is the frequency of the CT participating in CRPs;
- c_{CRP} is the number of collisions the CT meets in a CRP;
- l_{CRP} is the length of the CRP;
- s is the number of subsets that nodes are divided into.

When the traffic rate is low, f_{CRP} is small and the covert throughput is limited by CT's data packet transmission rate.

When the traffic rate is high, f_{CRP} is close to 1. Almost all the users participate in each CRP. According to the results of Janssen and Jong in [13], for large number of users, m , we can rewrite equation (1) as:

$$throughput \approx \frac{\log_2(m-1)}{m} \ln s \quad (2)$$

So, with high traffic rate, the covert throughput is expected to decrease as the number of users becomes too large.

The covert throughput is upper-bounded by $\log_2 s$. Upon each collision, the CT can embed at most $\log_2 s$ covert bits in its split decision.

B. Detectability

The covert receiver is undetectable since it only passively monitors the channel to track the collision resolution procedure.

The covert transmitter is undetectable under the conservative mode. In fact, some splitting algorithms use different splitting criteria. The first-come-first-serve (FCFS) algorithm [11] uses the packet arrival times to decide which subset the packet should join in. Sagduyu and Ephremides [12] include the node residual battery energy into its decision factors to save the nodes' energy and lengthen the network lifetime. After all, none of this information is directly known to the other nodes except the owner itself. The transmission decision is made by a node locally. It is unclear what factors have influenced this decision. As a result, use of this covert channel is very difficult to detect especially when the covert source has similar distribution as the splitting decision does.

V. SIMULATION RESULTS

To evaluate the performance of the covert channel, we developed a packet-level discrete event simulator which allows us to measure the covert channel performance under a variety of conditions including the number of nodes, traffic rate, covert operation modes, and different improved versions of the splitting algorithm. Although the simulation results are only isolated data points which can not describe the covert channel completely, the purpose is to demonstrate some of the covert channel characteristics quantitatively.

A. The Simulation

We simulate the collision resolution protocol with finite number of sources, denoted as m , which is finite and fixed. Each source can have at most one packet waiting in the

middle of a CRP and at most one other packet waiting to be transmitted in the next CRP. Packet arrivals at each user are i.i.d. Poisson processes, with mean λ/m packets per slot. The total average traffic rate is λ packets per slot.

The channel is assumed to be slotted. The length of each slot equals to that of a packet. A source node successfully transmits a packet only if there are not other nodes transmitting in the same slot. We also assume instant and perfect feedbacks with which each node, at the end of any slot, can determine whether there was none, one, or multiple nodes transmitting in that slot.

The performance metric that we are interested in is covert throughput. Denote Tr as the covert throughput and Nc as the number of covert bits transmitted within T number of slots. The covert throughput is calculated as:

$$Tr = Nc / T \quad (3)$$

B. Implementation

The basic splitting algorithm is implemented as described in section II. Collided nodes are always divided into two subsets. Features of the first and second improvements are added into the binary tree algorithm separately.

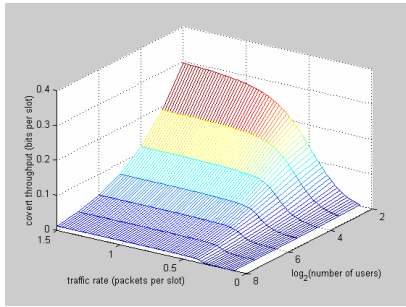
For the basic binary tree algorithm, all the three operation modes of covert transmission are implemented and evaluated. For the modified versions of the algorithm, the conservative mode is implemented and we compare the results obtained under the different algorithms.

Extra care has to be taken to implement the covert channel in the unblocked type of algorithms, where there is no longer a clear definition of "start" and "end" of the collision resolution periods. To correctly track the splitting history of the CT, the CR needs to know when the CT started to transmit the packet.

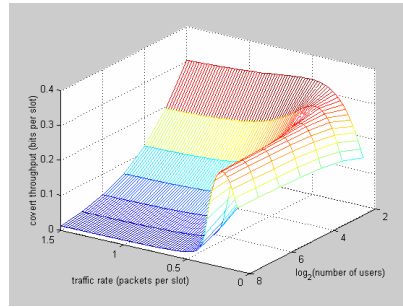
A practical solution is to use a specific node's success as the synchronization signal. The CT always starts its transmission right after it observes a packet is successfully sent by that node. There is an advantage to use the CR's own success, which allows the CR to control the covert transmission rate by adjusting its own transmission rate. In our implementation, the CR transmits valid packets upon their arrivals, and the CT joins the collision resolution procedure right after CR's success. Dummy packets are created when necessary.

C. Results

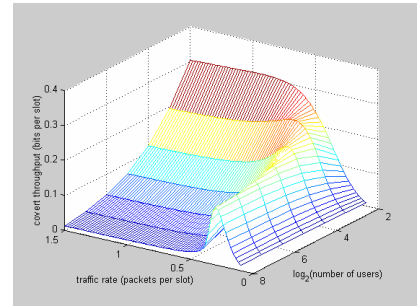
Simulation results are obtained under varying number of users and traffic rate. The simulation warms up with 20 CRPs. It lasts at least 1,000,000 slots and terminates at the end of the first CRP after the 1,000,000th slot, except for the unblocked algorithm where the simulation warms up with 100 time slots and terminates at the 1,000,000th slot.



(a) Conservative Mode

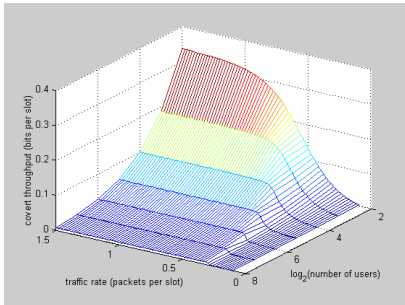


(b) Aggressive Mode

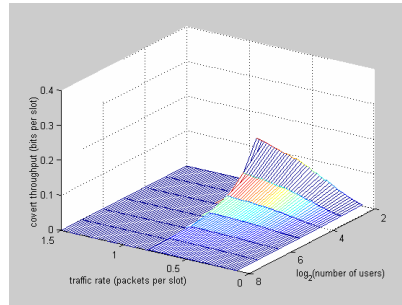


(c) Strategic Mode

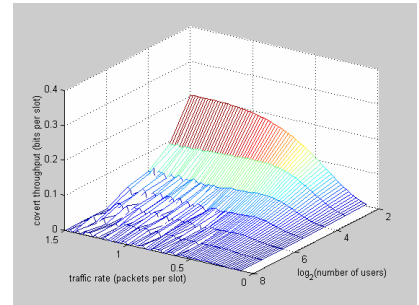
Figure 4. Covert Channel Throughput Using Binary Tree Algorithm



(a) Improvement 1



(b) Improvement 2



(c) Unblocked Algorithm

Figure 5. Covert Channel Throughput Under various Splitting Algorithms

Figure 4 presents the covert channel performance in the binary tree algorithm under all the three operation modes. It confirms the analysis discussed in the last section. The best throughput is obtained at high traffic rate and small number of users in the network. Occurrence of collisions is the fundamental requirement for covert transmissions. Light traffic implies rare collisions and thus holds back covert transmissions.

For the conservative mode, small covert throughput at low traffic rate is because of not having enough chance to participate in the CRPs. On the other hand, with high traffic rate and large network, collisions happen frequently. Large portion of time is used to resolve the collisions that do not involve the CT. So, the covert throughput degrades. The best covert throughput is about 0.3 bits/slot.

The aggressive operation mode effectively improves the throughput in the case of low traffic. Throughput improvement under the strategic mode is not as obvious. This is because in the aggressive mode, the CT not only makes use of every CRP with collisions, but also creates collisions through its aggressive transmission.

Figure 5 (a) presents the covert channel performance in the different versions of the splitting algorithms. The first part of improvement 1 does not affect the splitting procedure, but only reduces the length of the CRPs. The second part of the

improvement is based on the assumption that the relocated subsets are most probably empty, so it merely affects the splitting procedures except that it reduces the CRP length by saving some idle slots. As a result, the covert throughput. This improvement is especially obvious at the high traffic rate and small network sizes.

The second improvement to the basic binary tree algorithm splits the root of tree depending on how many nodes are expected to be in the root. This improvement takes effect when the traffic rate is high and many new arrivals happened during the last CRP. By splitting the new arrivals immediately, extra collisions are avoided and covert transmission is held back. In Figure 5 (b), the covert throughput drops steeply as the traffic rate increase above certain value.

Figure 5 (c) illustrates the case of unblocked algorithm. It shows similar features as it does with the blocked algorithm under conservative mode, except that the covert throughput is worse, especially at high traffic rate. This is the cost for synchronization between the covert transmitter and receiver. With high traffic rate, when the CT successfully transmits one packet, there almost always is a packet waiting to be transmitted the next. Under the blocked algorithm, the CT can continue its cover transmission immediately. But under the unblocked algorithm, it has to wait until it observes a success from the CR in order to stay synchronized with the CR.

VI. RELATED WORK

Covert channels have been studied in both the contexts of multi-level computer systems and wired computer networks. We review some of the past work on networks.

Handle and Sandford [3] provided a comprehensive analysis of hiding data in the OSI network model. Covert channels in the practical TCP/IP protocol suite are investigated in [4-8]. The basic idea, as established in [3], is to implement covert channels by utilizing special packet portions that are not used for normal transmissions or portions that are optional fields to be set as needed.

Girling [9] investigated the local area networks' susceptibility to covert channels in the use of the low level network protocols. While legitimate data are carried in the forms of packets, extra information may be embedded into other properties of the packets, such as the destination ID, the packet length, and the time between successive transmissions. Any user in the network who is aware of the covert communication may retrieve the covert information from observing these data packets.

Our work is most similar to Dogu and Ephremides [10], in which a MAC-layer covert channel was implemented using the First Come First Serve (FCFS) splitting algorithm. The covert information is conveyed in the number of collisions observed in a collision resolution period. The covert transmitter controls this number by generating dummy packets and thus causing additional collisions. The covert receiver passively monitors the channel and keeps track of the collision resolution procedure to extract the covert bits.

Our covert transmission scheme differs from [10] by using the covert transmitter's splitting decisions as the carrier of covert information. The receiver still remains fully protected from detection. Correct operation of the splitting algorithm is based on the assumption that the CRP can be correctly tracked. This in turn implies our covert transmission to be error free. Our covert scheme also has the advantage of not affecting the normal data packet transmission, and thus is very difficult to detect.

VII. CONCLUSION

In this paper, we have investigated the covert threat in use of the splitting algorithms. The covert channel is realized through embedding extra information into the covert transmitters' splitting decisions. The designated receiver passively monitors the channel and retrieves the covert information. The covert transmission is error free. Performance of the covert channel depends on the network size and traffic rate. The best throughput is around 0.3 bits per slot, and is obtained under small network size and high traffic rate. At low traffic rate, the covert transmitter can improve the throughput by aggressively transmits dummy packets, at the cost of being more easily detectably. The covert transmitter can also make strategic moves according to its own eagerness to transmit and willingness of being exposed. Various improvements to the

basic splitting algorithms do not eliminate the covert channel, although they have different affects on it.

Future work requires fully evaluating the covert channel in joint with the normal network performance. Measures are needed for protection against the covert threat, when MAC protocols based on the splitting algorithms are used.

REFERENCES

- [1] B. W. Lampson, "A note on the confinement problem." *Comm. ACM* 16 (1973) 613-615
- [2] V. Gligor, "A guide to understanding covert channel analysis of trusted systems," *Tech. Rep. NCSC-TG-030*, National Computer Security Center, Ft. George G. Meade, MD, U.S.A., Nov 1993.
- [3] T. Handel and M. Sandford, "Hiding Data in the OSI Network Model," *First International Workshop on Information Hiding*, Cambridge, U.K., May-June 1996.
- [4] N. B. Lucena, D. F. Calvert, J. Pease, and S. J. Chapin, "Semantics-Preserving Application-Layer Protocol Steganography," <http://www.sai.syr.edu/facultypapers/ProtoStego.pdf>
- [5] C. H. Rowland, "Covert Channels in the TCP/IP Protocol Suite," *Peer Reviewed Journal on the Internet*, July 1997. http://www.firstmonday.dk/issues/issue2_5/rowland/
- [6] K. Ahsan and D. Kundur, "Practical Data Hiding in TCP/IP," *Proc. Workshop on Multimedia Security at ACM Multimedia '02*, French Riviera, Dec. 2002.
- [7] C. Abad, "IP Checksum Covert Channels and Selected Hash Collision," <http://www.gravitino.net/~aempirei/papers/pccc.pdf>
- [8] J. Giffen, R. Greenstadt, P. Litwack, and R. Tibbetts, "Covert Messaging through TCP Timestamps," *PET 2002 Workshop on Privacy Enhancing Technologies*, San Francisco CA, April 2002.
- [9] C. G. Girling, "Covert Channels in LAN's," *IEEE Trans. Software Engineering*, Vol. SE-13, No.2, Feb. 1987.
- [10] T. M. Dogu and A. Ephremides, "Covert Information Transmission through the Use of Standard Collision Resolution Algorithms," *IH'99*, Dresden, Germany, Sept./Oct. 1999.
- [11] D. Bertsekas and R. Gallager, "Data Networks," Prentice Hall, Inc., 1992.
- [12] Y. E. Sagduyu and A. Ephremides, "Energy-Efficient Collision Resolution in Wireless Ad-Hoc Networks", *Proc. IEEE INFOCOM 2003*, San Francisco, April 2003
- [13] Augustus J. E. M. Janssen and Mare J. M. de Jong, "Analysis of Contention Tree Algorithms", *IEEE Trans. Information Theory*, Vol. 46. No. 6. September 2000