A Dynamic Strategy to Cache Out-of-Sequence Packet in DPI System

Qingyun Liu[1,2,3] , Wenzhong Feng[1], and Qiong Dai[1]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2] Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[3] Graduate School of the Chinese Academy of Sciences, Beijing, China
liuqingyun@iie.ac.cn,
hectorinsane@gmail.com,
daifq@hotmail.com

Abstract. As a major approach for a network security system to discover threats or forensics, DPI (Deep Packet Inspection) technique is widely used in monitoring network flow. With the rapid development of Internet bandwidth, DPI system is facing more and more challenges on performance. One of these challenges is that out-of-sequence packets in TCP transmission will greatly affect memory consumption and data-recall. For a large scale DPI system, each DPI node has to monitor a huge amount of TCP session. It will consume too many resources to allocate plenty of space for storing all out-of-sequence packets. Meanwhile, insufficient space for buffer results in dropping packets and thus unable to reassemble network flow. We analyze the out-of-sequence characteristic of different Internet flow, and implement a dynamic strategy to cache out-of-sequence packet, which provide a more flexible way to keep track of the sessions. Experiment shows that based on the new strategy, a DPI system can greatly improve the completeness of data recall with little extra consumption of space.

Keywords: TCP out-of-sequence, out-of-sequence packet buffer, network flow identification.

1 Introduction

DPI (Deep Packet Inspection) is the technique used to detect and control on network flow of application layer, and it has a good effect in many fields, such as management on network flow, analysis on network and security, and so on. When IP packets, TCP packets or UDP packets flow through the DPI-based bandwidth management system, it will reassemble the application layer data of the OSI model by inspecting the IP packets deeply, and get the full contents of the application, and then deal the flow according to the pre-defined management policies. With the rapid development of network bandwidth, the TCP sessions processed by the DPI-based network security product also increase rapidly. Meanwhile, the out-of-sequence TCP packets become more obvious in WAN link. The process ability of DPI-based network security

Y. Yuan, X. Wu, and Y. Lu (Eds.): ISCTCS 2012, CCIS 320, pp. 564–571, 2012.

product is limited by its physical attributes (such as memory, etc.). With the rapid development of network bandwidth, DPI system is also facing more challenges. As Gilder's law points out, the growth rate of the network bandwidth is triple of that of computing power. So DPI system needs to process more and more data with limited time and memory resources.

The key point of designing the DPI system based on the content scanning is to process the out-of-sequence packet. TCP/IP are the stream-oriented protocol and provides a stable transfer mechanism, the two parties in the network check each other's response to determine whether the data is successfully transmitted. Even so, the arriving sequence of the TCP packets may be different from that of sending due to factors like delay, packet loss and different routing path. DPI system needs to cache and reassemble the out-of-sequence data, and then transfer it to pattern matching module for further processing, which greatly increased the memory burden of DPI system. Especially for the embedded device with less memory resource, it is requisite to utilize the memory more efficiently. With the rapid development of network bandwidth, DPI system needs adapt to monitor more network applications, such as scanning the compressed file, decoding and detecting the audio and video files, and so on. More and more new requirements on the performance of out-of-sequence data process are put forward for DPI system.

In summary, the out-of-sequence packet processing will be directly related to the performance of the DPI system. In order to effectively use the memory resource of the packet buffer, this work designed an adjustment mechanism for caching the out-of-sequence data. The mechanism obviously improved the recall ratio of DPI system. The remained of this paper is organized as follows:

1) Section 2: Introduce the research on the out-of-sequence data processing.
2) Section 3: Introduce the analysis on the features of the out-of-sequence data.
3) Section 4: Introduce the system architecture design of the dynamic out-of-sequence packet caching based on the application layer protocol identification.
4) Section 5: Introduce the system verification result.
5) Section 6: Summarize and evaluate on the system design.

2 Research on the Processing of the Out-of-Sequence Data

Some researchers have researched the features of the out-of-sequence TCP packet in different application scenarios. Paxson[1] tested 20000 TCP sessions, and found that about 12% sessions are out of sequence. Jasiwal[2] experimented on the Tier-1 backbone network, and the result is that about 4% data is out of sequence, the main cause is that the packet is transmitted again if it doesn't arrive at the destination correctly or the routing path is different.

There are two basic strategies on the processing of the out-of-sequence packet[3]: caching the out-of-sequence packet and discarding the out-of-sequence packet. The method of caching the out-of-sequence packet is to cache the out-of-sequence packet till the missing packet arrives. Then, re-order the packets according to the sequence number (SEQ) in the TCP header, and send it to the detection unit of upper layer. This method will occupy a large amount of buffer size in congested network. Semke[4] puts forward an automatic buffer adjustment algorithm, and as far as possible, it makes that each TCP session obtains a relatively balanced number of buffer. It creates a special buffer to cache the out-of-sequence data, and divides the buffer into N blocks. One session only use one block to cache data, this case considered the global resources. It ensures that the maximum and minimum size of available buffer used to cache the out-of-sequence is optimal, and avoids exhausting the resources. However, the defect is that it can't save the related data completely if a large amount of out-of-sequence data is transmitted. Amit[5] put forward an assignment algorithm based on probability, it checked the remained memory of the system, and when an out-of-sequence packet is received, system rejects it according to the changed probability to avoid exhausting the resources. Fisk[6] put forward an algorithm of discarding the follow-up packets whose sequence number (SEQ) order is inconsistent. Because of the TCP's retransmission timeout mechanism, the packets with the consistent logic order can be obtained finally. However, this case leads to discard a large amount of packets and also largely reduce the TCP sending windows, which result in the reducing of network throughput.

3 Analysis on the Features of the Out-of-Sequence Data

It isn't rare that the network traffic appears out-of-sequence situation[7]. Network congestion, different routing path, and so on, all possibly result in an out-of-sequence packet. As mentioned in above, DPI needs to cache the out-of-sequence TCP packet which flows through the system. Here, we define 'the maximum buffer size allocated for a TCP session *S* for caching the out-of-sequence packets' as the disordering tolerance of S, noted tol(S). It is to say that tol(S) is the maximum number of the system cached packets before the last packet (Pkt_n) which makes the out-of-sequence packets can be reassembled arrives. When the number of cached packets is larger than tol(S), if the packet can't be reassembled, then send all the cached data to upper layer. Later, if the packet is out-of-sequence and its sequence number equal to that of some one packet cached before, it will be discarded.

In practice, the different types of transmission data influence the DPI system in different degree. For example, during a video file transmission, if one critical packet was lost, the reassemble of the transmitted content for file decoding and detecting may become impossible; whereas, it harms rareness for an IM chat session if the same case occurred. Various protocols are designed to transmit different types of data, they show different sensitive to out-of-sequence.

To check out the size of buffer required to cache all the out-of-sequence packets, we designed and conducted the following experiments. The dataset for these experiments is composed of the traffic traces of 190 emails with audio/video attachments. The sizes of these attachments are range from 2 to 15 MB. The collected trace was fed to DPI modules repeatedly with setting various maximum buffer sizes, and we collect the number of audio/video files which are reassembled correctly.

Table 1. The number of complete audio/video flows on different tol(S)

tol(S)	5	10	20	40	60
processed audio/video flows (Bytes)	1348M	1354M	1358M	1362M	1364M
number of complete audio/video flows	1	4	21	64	171
ratio of complete audio/video flows	0.6%	2.3%	14%	37%	90%

The experiment shows that, the recall ratio of audio/video files is sensitive to tol(S) of DPI modules. The recall ratio is only 2.3% when tol(S) is 10. It also shows a dramatic promotion when tol(s) increases from 40 to 60. Finally, the recall ratio reaches a plateau at a high level as tol(S) >= 60.

In summary, we learn that the out-of-sequence of network traffic exhibits the following features:

1. Various protocols show a diverse sensitive to out-of-sequence.
2. For audio/video traffic, DPI needs a large buffer to cache all the data for reassembling.

The increasing bandwidth requires the growing memory of the out-of-sequence processing module of the DPI system. If using a memory pool to support "allocate on demand", the out-of-sequence module can manage the memory more effectively, however, it can neither handle the large volume of cached data under network congestion, nor deal with those malformed attacking packets. Whereas, on the other hand, if we manage each TCP session individually, it is hard to determine the cache size: too large size will cause resource wasting, while too small will harm the data reassembling. Consequently, under the current network environment, the DPI system needs an out-of-sequence process strategy which incorporates the merits of the above two cache managing approaches.

4 Design of the Dynamic Out-of-Sequence Packet Caching Based on the Application Layer Protocol Identification

The existing works on TCP disordered-packet-reassembling mostly focus on the factors like delay and space. They use the same reassembling strategy for all TCP packets. In this work, we propose and implement a framework which incorporates DPI technique and cache size adaptation. Thus, it can dynamically adjust the cache size for disordered packets, and get more effective memory utilization. At the meantime, based on the observation that different applications behave diversely on packets disordering, we determine the cache strategies of out-of-sequence packets for TCP sessions according to its application-layer characteristics in the management of massive TCP sessions.

The core idea of the framework is that: by feeding the output of application layer protocol identification back to the connection management module, it can assign a proper monitoring level based on the monitoring requirements. Each monitoring level has a corresponding maximum cache size for disordered packets, where the size can be adjusted according to the utilization of current cache space.

The implementation of the adjustment thread follows the following principles:

1. The adjustment thread should occupy memory and time as little as possible when collecting the information of the main program.
2. There should be a correlation between free cache size and the disordering tolerance of various monitoring levels.
3. Under the same memory occupation, the maximum number of disordered packets for each monitoring level should be consistent after a long running.
4. The tolerance variation caused by adjustment thread only affects the follow-up packets. Thus the previous session could keep untouched. It can prevent the performance loss caused by frequent interactions and lock/unlock operations, and meanwhile avoid the churn of space occupancy.

4.1 Process on Transport Layer

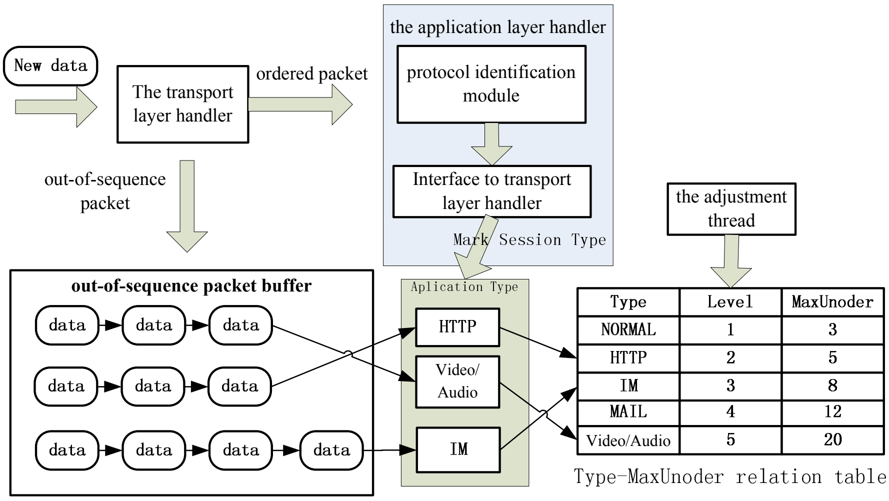
The transport layer handler should maintain a session table to record relevant information (including SEQ/ACK sequence number, corresponding application layer protocol, etc.) of unidirectional TCP flows. Upon a packet arrived, system looks up the session table firstly (to create it if no item exists in the table), and then compare the sequence number with the expected one to decide whether it’s an out-of-sequence packet. If it is, allocate appropriate memory space to cache it and count the number. When the number of disordered packets (of a unidirectional TCP flow) exceeds tol(S), all the allocated memory should be freed, and these packets are dropped. Otherwise (not an out-of-sequence packet), send it to the upper layer for the processing, such as application layer protocol identification, and so on.

4.2 Interface of Application Layer

The transport layer handler should provide an interface to the application layer handler, by which the application layer can obtain data from transport layer, and label the corresponding protocol of the flow.

4.3 Design of Adjustment Thread

The role of adjustment thread is to adjust the disordering tolerance of different traffics according to current cache occupancy (refer to Fig. 1). In the following experiments, we define several levels of disordering tolerance and adjustment functions. According to the statistical sensitive degree of the data transmit by different protocol, we set different level for every protocol. Different type of traffic can be set as a relevant level based on the network traffic characteristics and the actual detecting requirements. After the application protocol identification module recognized the protocol,


New data
The transport layer handler
ordered packet
the application layer handler
protocol identification module
Interface to transport layer handler
Mark Session Type
out-of-sequence packet
the adjustment thread
out-of-sequence packet buffer
data
Aplication Type
HTTP
Video/ Audio
IM
Type-MaxUnoder relation table

Type	Level	MaxUnoder
NORMAL	1	3
HTTP	2	5
IM	3	8
MAIL	4	12
Video/Audio	5	20

Fig. 1. The architecture of dynamic strategy to cache out-of-sequence packet

system set an appropriate tolerance level for the session. Each level has multiple thresholds of the out-of-sequence occupancy ratio, and once the actual occupancy exceeds a threshold, the tolerance of all levels will be altered. This will affect the connection management of successive packets.

Based on the above framework, we implement a timed adjustment thread. Once the thread started, will collect the cache space occupancy for disordered packets of the connection management module per second. When there is a large bulk of idle space, the adjustment thread will increase tol(S) so that each connection can cache more disordered packets. Otherwise, it decreases tol(S). The experiments show that, even though the DPI's load balance behaves normally, the cache occupancy of various threads shows a significant difference, which can up to 20% of the cache size. Thus it is requisite to individually regulate each thread's space occupancy, and make the maximum cached packets number of a connection can be adjusted at a small granularity.

5 Verification on the Effect of Dynamic Cache Adjustment

To evaluate the effect of memory adjustment thread, we configured the following experimental environment (refer to Fig. 2). It includes a node on a backbone network with 2Gbps bandwidth, a traffic replay device which can replicate the input traffic to multiple output port, and three DPI server modules. The input traffic is injected with some labeled audio/video streams[1]. The three DPI modules adopt the following three cache strategies respectively: with tol(S) = 5, 80, and a dynamic adjustment.

[1] Under real network environment, the interrupt of the data transportation happens easily, that is, the traffic passing through DPI module may be not a complete connection. So we inject some labeled flows into traffic to accurately detect the recall ratio of DPI system.

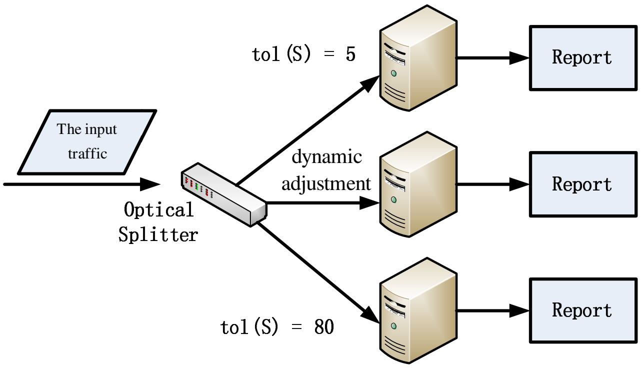


Fig. 2. The experimental environment for three cache strategies evaluate

The experiment lasted 6 hours. The DPI module will record the memory utilization during processing traffic data, and calculate the recall ratio of audio/video flows once experiment terminated.

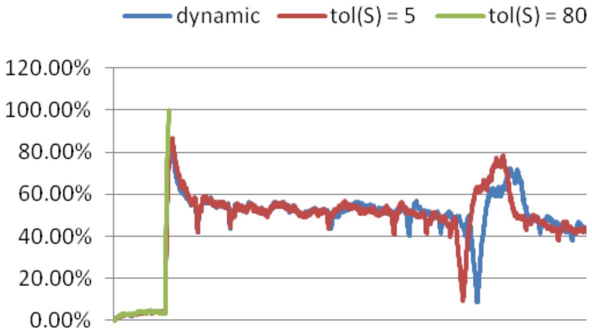


Fig. 3. The memory utilization on three cache strategies

Table 2. The number of complete audio/video flows on three cache strategies

tol(S)	5	80	dynamic
number of complete audio/video files	6	65[2]	698
ratio of complete audio/video files	0.73%	9.3%	85.01%

[2] The fixed strategy with tol(S) = 80 will cause the buffer always full, and not out-of-sequence packets can be saved.

For the cache memory occupancy, the dynamic cache strategy behaves similar as that with a fixed number of cached packets. While for the recall ratio, dynamic strategy can achieve a higher level. The strategy of fixing large number of cached packets will casue DPI module laid off once the large cache space was exhausted, and it will affect the recall ratio.

6 Discussion and Conclusion

In this work we analyzed the difference of out-of-sequence data for various type of network traffic, and proposed a dynamic strategy for caching out-of-sequence packets. The core idea is to rationally allocate the memory to various traffics for a higher recall ratio when the total memory resource is limited. This new strategy is deployed in the real network environment, and it promotes the audio/video flow recall ration by about 7% with the same cache space occupation. However, the effect of the dynamic strategy relies on the application layer protocol identification. Thus for those flows, where the application protocol was not identified even if the maximum number of cached packets arrived, cannot be recalled completely by the new strategy. The future research can focus on the TCP connection management of DPI system, such as how to use the application layer identification result to optimize the phase-out strategy of TCP connection.

Acknowledgements. Supported by the "Strategic Priority Research Program" of the Chinese Academy of Sciences (Grant No. XDA06030200) and the National High-Tech Research and Development Plan 863 of China (Grant No. 2011AA010703) and the National Natural Science Foundation (Grant No. 61070026).

References

1. Paxson, V.: Automated Packet Trace Analysis of TCP Implementations. In: Proceedings of the 1997 SIGCOMM Conference, Cannes, France, pp. 167–179 (September 1997)
2. Jaiswal, S., Iannaccone, G., Diot, C., Kurose, J., Towsley, D.: Measurement and Classification of Out-of-Sequence Packets in a Tier-1 IP Backbone. IEEE IEEE/ACM Transactions on Networking 15(1) (February 2007)
3. Xu, K., Li, Y., et al.: line speed deep packet detecting techniques on high speed link. 徐克付，李阳等，高速网络线速深度分组检测技术，信息技术快报 9(3) (May 2011)
4. Semke, J., et al.: Automatic TCP Buffer Tuning
5. Amit, S., Jaggi, M.: (Sunnyvale, CA, US) , Buffer allocation using probability of dropping unordered segments
6. Fisk, M., Varghese, G.: Fast content-based packet handling for intrusion detection. Technical Report CS2001-0670, Department of Computer Science, University of California, SanDiego (May)
7. Bennett, J.C.R., Partridge, C., Shectman, N.: Packet Reorder Is Not Pathological Network Behavior. IEEE/ACM Trans. Net. 7(6) (December 1999)