SCIENCE CHINA
Information Sciences

• LETTER •

August 2016, Vol. 59 089301:1–089301:3
doi: 10.1007/s11432-016-5523-y

RMI-DRE: a redundancy-maximizing identification scheme for data redundancy elimination

Nan ZHANG, Xiaolong YANG*, Min ZHANG & Keping LONG

School of Computer and Communication Engineering, University of Science and Technology Beijing (USTB), Beijing 100083*, China*

Received September 16, 2015; accepted September 30, 2015; published online March 10, 2016

Citation Zhang N, Yang X L, Zhang M, et al. RMI-DRE: a redundancy-maximizing identification scheme for data redundancy elimination. Sci China Inf Sci, 2016, 59(8): 089301, doi: 10.1007/s11432-016-5523-y

Dear editor,

Data redundancy elimination (DRE) technology can efficiently reduce the redundant IP traffic in the network [1,2]. A detailed DRE process consists of many steps, such as the fingerprint selection, chunk matching, packet encoding, and packet decoding [3]. Fingerprint selection is one of the most important factors for DRE. In traditional DRE, the redundant chunk size is set to a fixed value. The fixed chunk size will affect the DRE performance seriously, so keeping the redundant chunk size variable is necessary [4]. Hence, multi-resolution chunking (MRC) has been proposed. However, MRC does not consider the overlap between the adjacent chunks, which inevitably brings some unnecessary overhead. Besides, chunk-matching is another key point for DRE. There are two popular chunk matching mechanisms, i.e., Chunk-Match and Max-Match. Chunk-Match may not bring much storage overhead, but the size of chunk-matching is limited, which results in some redundant bytes to be missed. By contrast, Max-Match can maximize the size of chunk-matching by extending the boundary of the identified redundant chunk towards its front or rear. Nevertheless, Max-Match brings high storage overhead because it needs to cache the whole packet for the maximizing match. Based on the assumption that consecutive popular chunks have a fair chance of appearing together, CombiHeader proposed a matching mechanism to maximize the chunk-matching size by using the chunk aggregation technique [4]. The disadvantage of CombiHeader's mechanism is that the redundant chunk size must be a multiple of the size of a basal chunk. Thus, in order to overcome the problems, we propose a new DRE scheme named as redundancy-maximizing identification for data redundancy elimination (RMI-DRE). RMI-DRE includes two main parts: a new fingerprint selection algorithm (i.e. w-MAXP+) and a dynamic chunk matching mechanism.

To illustrate RMI-DRE scheme in detail, four sub-tasks are presented, i.e., the setting of basal chunk size and upper bound, the generation of representative fingerprints, the chunk-matching to find redundant bytes as more as possible, and the encoding/decoding for redundant chunks.

1. How to set the basal chunk size and the upper bound?

According to the distributions of matched chunk size in enterprise network and campus network in [4], we can find that more than 70% of matched chunks are shorter than 150 bytes while only less than 10% of matched chunks are near upon 1500 bytes. What is more, more than 60% of matched

* Corresponding author (email: yangxl@ustb.edu.cn)

The authors declare that they have no conflict of interest.

chunks range from 42 to 68 bytes according to [4]. In RMI-DRE, we need to define two boundary values: one is the size value of basal chunk, and the other is the upper bound value of extended chunk. It is known that 32 and 64 bytes are suggested to be the best choice of optimizing the performance of DRE. Thus, the size of basal chunk can be set as 32 bytes and the upper bound of extended chunk can be set as 64 bytes in this letter.

2. How to generate representative fingerprints?

We propose a new fingerprint selection algorithm called w-MAXP+ based on the local-maximum in this section. In w-MAXP+, the packet payload is divided into many groups, and each group includes p bytes. The local-maximum of every group will be identified out first. Then, the w bytes at the rear of the local-maximum will be selected as a basal chunk. The byte region at the rear of the basal chunk is considered as the extended part. The w-MAXP+ algorithm can make the fingerprint distribution more uniform, because at least one fingerprint in every p bytes will be selected as the representative fingerprint. The computing overhead in w-MAXP+ is reduced greatly, because we only compute the fingerprints that have been identified out as representative fingerprints. However, if the byte number between two adjacent local-maximums is less than the byte number in redundant region, the two adjacent chunks will overlap each other. In order to avoid the overlapping of selected adjacent chunks, w-MAXP+ algorithm makes an improvement: if Chunk A and Chunk B are two overlapped adjacent chunks, w-MAXP+ will skip some bytes between Chunk A and Chunk B. In cache, the chunk is stored in two parts, i.e., the basal chunk part and the extended chunk part. We assume that the size of basal chunk stored in cache is w bytes and the size of extended chunk stored in cache is E bytes (E is set as 32 bytes in this letter), so $(w+E)$ is the maximum of the redundant region size. For two adjacent groups, we set the first byte of current group as the benchmark. We assume that the position of the local-maximum in this group is j-th byte and the beginning position in next group is i-th byte. If $(j+w+E) > i$, the redundant region of this group will overlap with the next group. In this case, w-MAXP+ will skip K bytes to make the beginning position of next group change from i-th to i'-th. What is more, we can infer that the value of K should be $(w+E)-(p-j)$ at least. In the formula, $(w+E)$ represents the maximum size of the redundant region and $(p-j)$ represents the number of bytes from the local-maximum of this group to the beginning position of next group, so the number of overlapping bytes cannot exceed $(w+E)-(p-j)$. On the contrary, if $(j+w+E) < i$, the overlapping will not happen, so we do not need to skip any byte.

3. How to dynamically match the redundant bytes as many as possible?

Generally, at both source and destination end, the fingerprint table and corresponding chunks are stored in cache. In RMI-DRE and traditional DRE, the fingerprint table cache is the same, but the chunk cache is not. For RMI-DRE, the chunk cache consists of two parts: one is the basal chunk (like traditional DRE) and the other is the extended part. In order to describe RMI scheme more directly, we will take an example. We assume that FP1 is one representative fingerprint identified out using w-MAXP+ algorithm. At first, FP1 will be compared with the items in fingerprint table to find out the same fingerprint. Chunk1 is the redundant chunk corresponding to FP1. In traditional DRE, Chunk1 will be identified as the final redundant chunk. But in RMI-DRE, Chunk1 is only a basal redundant chunk. The subsequent bytes of Chunk1 (i.e., 32nd-63rd byte) will be compared with E1 (E1 is the extended part stored in cache) byte by byte until matching unsuccessfully. After extending the basal chunk, the size of extended-Chunk1 is 44 bytes. The extended-Chunk1 will be identified as the final redundant chunk. Whereas, if no fingerprint is the same as FP1 in cache, Chunk1 together with its extended part and FP1 will be stored in the end of the cache. The new algorithm which matches the redundant bytes as more as possible is called Redundancy-maximizing.

4. How to encode in the source node and decode in the destination node?

After chunk-matching, the next step is encoding the original packet by meta-data. Every meta-data should include enough information for the recovery in the destination end. Besides the essential information contained in traditional DRE, the chunk size after extending (for example, extended-Chunk1 is 44 bytes) must be included in meta-data. Likewise, the encoded packet is shorter than the original one. Then, the encoded truncated packet will be transferred to the destination end. At the destination end, taking Chunk1 and FP1 as examples, the fingerprint (FP1) and the size of extended chunk (44 bytes) will be extracted from the packet firstly. Secondly, the fingerprint which is the same as FP1 will be checked out from the cache and its corresponding chunk (Chunk1) will be pointed out. Thirdly, Chunk1 will be extended to 44 bytes according to the extension cache (E1) in the destination end. Till now, the original

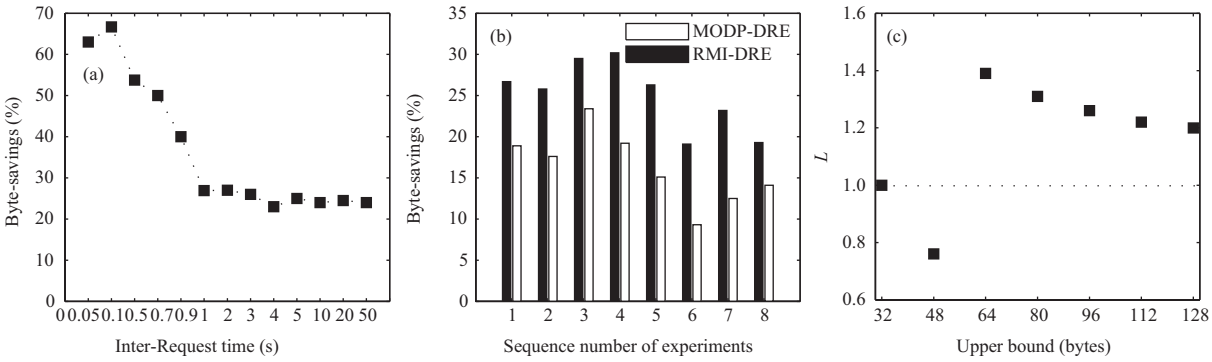

Figure 1 Performance evaluation of RMI-DRE. (a) Byte-savings with different Inter-Request time; (b) comparison in byte-savings; (c) marginal income.

packet is recovered completely.

In this letter, experiments are performed on OPNET Modeler14.5. Two FTP servers and two routers are set at the source end and the destination end respectively in the network scenario. Figure 1(a) shows the byte-saving ratio with different packet transmission interval time. In this letter, "byte-savings" is defined as the ratio of reduced flow and the original flow. We can find that the byte-savings is pretty high and unstable when the Inter-Request time is very short as shown in Figure 1(a). In these conditions, many packets cannot be disposed timely, so the savings is not displayed appropriately. But when Inter-Request time exceeds 1 s, the byte-savings is stable. We can find that the byte-saving ratio of RMI-DRE is always exceeding 20 percent.

MODP is a traditional fingerprint selection algorithm. We compare RMI-DRE with the MODP-based DRE scheme (for short MODP-DRE) in byte-saving rate. Figure 1(b) shows the comparison from eight random experiments. RMI-DRE is always better than MODP-DRE as Figure 1(b) shown. But we can infer that the execution time in RMI-DRE should be longer than MODP-DRE because of more storage and computing overhead. So it should be considered that if the increase of byte-saving ratio is enough to compensate the increase of overhead.

In this section, the overall overhead of computing and storage is represented as "RTT". And it is obvious that "RTT" will increase with the increasing of upper bound. Now Figure 1(c) reflects the simulation results of the parameter L which can be regarded as the marginal income. The trades between the byte-savings ratio and the overhead will be showed by L. As the following formula shown, "ΔSavings" denotes the increasing range of byte-saving ratio, which can be described concretely as the difference value of byte-saving ratio between RMI-DRE and MODP-DRE. In addition, "ΔRTT" is the growing range of delay time caused by extra byte-matching and computing. "ΔRTT" can be described as the difference value of the execution time between MODP-DRE and RMI-DRE. The parameter L is set as $((\Delta\text{Savings})\%)/((\Delta\text{RTT})\%)$. As shown in Figure 1(c), when the value of L is less than 1, it can be inferred that the byte-saving ratio has not increased very fast but the overhead bas increased rapidly. So we can infer that the increase of savings cannot compensate the growth of overhead when upper bound is 48 bytes. However, the situations are reversed when the value of L is more than 1. RMI-DRE can achieve a better performance when the upper bound is in the range of about 64–128 bytes, because the increase of byte-saving ratio is enough to offset the increase of overhead. Furthermore, RMI-DRE can get the peak value when the upper bound is 64 bytes.

Acknowledgements This work was supported by National Basic Research Program of China (973) (Grant No. 2012CB315905), National Natural Science Foundation of China (Grant Nos. 61172048, 61100184, 61201128), and National High Technology Research and Development Program of China (863) (Grant No. 2013AA01A209).

References

1 Shi Y, Qiu X S, Guo S Y. Genetic algorithm-based redundancy optimization method for smart grid communication network. China Commun, 2015, 12: 73–84
2 Wang X, Qi Y X, Wang K, et al. Towards efficient security policy lookup on many-core network processing platforms. China Commun, 2015, 12: 146–160
3 Halepovic E, Williamson C, Ghaderi M. Low-overhead dynamic sampling for eedundant traffic elimination. J Commun, 2012, 7: 28–38
4 Anand A, Muthukrishnan C, Akella A, et al. Redundancy in Network Traffic: Findings and Implications. In: Proceedings of the eleventh international joint conference on Measurement and modeling of computer systems, New York, 2009. 37–48