Content-Centric Networking: Effect of Content Caching on Mitigating DoS Attack

Kai Wang, Jia Chen, Huachun Zhou and Yajuan Qin

National Engineering Laboratory for Next Generation Internet Interconnection Devices, Beijing Jiaotong University Beijing, 100044, China

Abstract

Content-Centric Networking (CCN) is a novel networking paradigm making named data the first class entity rather than IP address. That is, it cares about which data to fetch rather than which host to reach. One key feature of CCN is the content caching that CCN routers are caching named contents instead of IP addresses, which makes the effect of Denial-of-Service (DoS) attack different from in TCP/IP networking. In this paper, we propose a DoS attack model for CCN. Comparing with TCP/IP networking, we use the model evaluations and NS2 simulations to show the effectiveness of CCN's content caching on limiting DoS attack. In addition, we analyze how the content caching Time-To-Live (TTL) in CCN affects DoS attack. Both the results show CCN has better survivability and resilience than TCP/IP networking when suffering DoS attack and larger content caching TTL brings in better DoS limiting effect.

Keywords: *Computer network reliability, Network Modeling and Simulation, Content-Centric Networking, content caching, Denial-of-Service.*

1. Introduction

Denial-of-service (DoS) attack is one of the most notorious security threats to the Internet using TCP/IP networking today. DoS attack is based on the botnet that is composed of a large number of zombies, which are formed by worm infecting or other malicious software permeating. It attempts to make a computer resource unavailable to its legitimate users and can cause unprecedented damage [1].

The common methods of DoS attack involve saturating the target machine with external communications requests and consuming its resources, so that it can no longer provide its intended service to the legitimate users. For example, an attacker can attempt to infect the vulnerable computers on the Internet by some malicious programs such as worms [2-9], which are self-replicating software and can spread throughout the Internet by themselves. The malicious programs are used to infect a large number of vulnerable computers and then recruit them as zombies, which form the botnets [10, 11]. These botnets can be coordinated and launch massive Denial-of-Service (DoS) attacks to the target so that it cannot be accessed by the legitimate users, which leads to the denial of service.

DoS attack is recalcitrant because it exploits the vulnerability of the fundamental of the Internet architecture using TCP/IP networking: no matter how over-provisioned you are, if everyone in the world sends you a single packet, legitimate traffic will be denied.

While there are so many researches on DoS attack [12-22, 36-38] in the current Internet using TCP/IP networking, there is a growing consensus that the Content-Centric Networking (CCN) [23-34] will be the promising architecture for future Internet, where the named content is the first class entity used for routing and locating in the network topology rather than IP address. Recently, there have been several proposals sharing similar ideas with CCN [23, 27], such as TRIAD [25], ROFL [26], DONA [28], CURLING [35].

Finding the requested named content in CCN is executed by the CCN Router. The core packet forwarding engine of the CCN Router has three main data structures [23]: the FIB (Forwarding Information Base), Content Store (buffer memory) and PIT (Pending Interest Table).

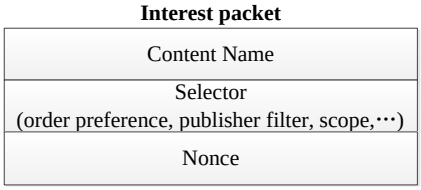

Fig. 1. Interest Packet

The FIB is used to forward Interest packets (Figure 1) to potential source(s) of matching Data, and it allows a list of outgoing faces rather than a single one, which is different from the IP FIB. The Content Store has the same function as the buffer memory of an IP router except for a different replacement policy. The Content Store remembers the arriving Data packets as long as possible because the CCN packets are idempotent, self-identifying and self-authenticating and each packet is potentially useful to many consumers (e.g., many hosts reading the same newspaper or watching the same YouTube video). The PIT

keeps track of Interest packets that have been forwarded upstream towards content source(s) so that the returned Data can be sent downstream to its requestor(s). And if the same content is requested by several users, only one Interest is forwarded upstream and a single PIT entry records all the interfaces over which the requested content must be returned. We call this as PIT aggregation.

In CCN, when the content is published into a network repository, it is split into self-identified chunks sent upon end-user request (interest). Each node receiving an interest from an input interface, verifies if the given chunk is present in its Content Store, otherwise it forwards the interest to the interface(s) indicated by the FIB. Ongoing requests are tracked by a PIT in order to send back data through the reverse path of interests. The index structure used for lookup is ordered so that a Content Store match will be preferred over a PIT match that will be preferred over a FIB match. Thus, if there is already matching Data cached in the Content Store of the CCN Router, it will be sent out the interface which Interest packets arrived on and the Interest packets will be discarded, without requesting the PIT and FIB. That is, if the requested content has been cached in the CCN Routers, the Interest packets will be responded by these CCN Routers directly before the content caching is timeout.

Although it is believed that CCN is more secure than TCP/IP networking due to its natural characters, it still cannot eliminate either the botnet forming or the DoS attack after finishing the botnet. For example, an attacker can attempt to send malicious Interest Packets to the service rendezvous [24] in CCN to search for vulnerabilities and then confirm connection with them. Then the attacker sends malicious programs to the vulnerabilities attempting to infect them and recruit them as zombies to form the botnet, which is the source of DoS attack. The botnet can unify the "publisher filters" of the "selectors" in all the attacking Interest packets to be the target content repository(s) and specify the Content Names in Interest packets to be a large number of different content names that the target provides (e.g., ranging from different news to different videos providing by Sina), which means all the attacking Interest packets from the botnet will be flooded to the target content repository(s) storing the requested contents of the specified publisher. Figure 2(a) shows the structure of the name of data packet [23] in CCN and Figure 2(b) illustrates how the botnet attacks the target content repository(s).

In Figure 2(b), the botnet can unify the "publisher filters" of the "selectors" in all the attacking Interest packets to be 'Sina', and specify the 'b1' part of the names of attacking packets ranging from 'news' to 'videos' or just to any random names to generate a large number of attacking packets. More terribly, the botnet can specify from the 'news' to the 'videos' ('b2' part) ranging from news 'A' to news 'X' or some other random news names and from video 'U.avi' to video 'K.rmvb' or some other random video names to expand the number of the attacking packets by the exponential law. With the extremely large number of such attacking Interest packets to the target content repository(s) 'Sina.com' at a certain period, all the resources of Sina.com will be consumed and it will lead to denial of service to all the other legitimate users requesting the same content.

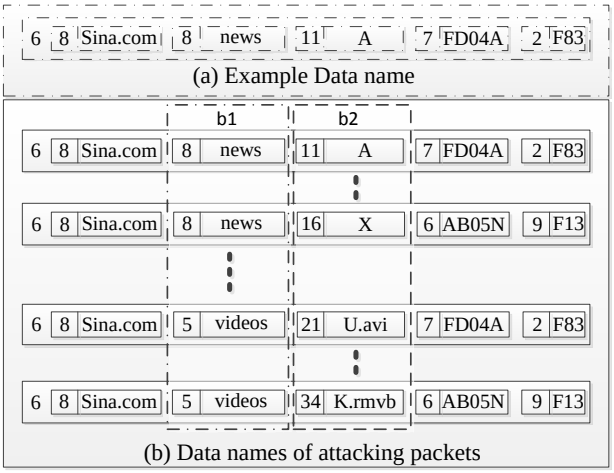

Fig. 2. Structure of the attacking packet

Neither the TCP/IP networking nor the CCN can eliminate DoS attack, but the DoS attack in CCN will be more lightweight compared with the TCP/IP networking due to the content caching. If the requested content has been cached in the CCN Routers, which is downstream the target content repository(s), the attacking Interest packets will be responded by these CCN Routers directly and will never arrive at the target content repository(s) again before the content caching is timeout. While as for the TCP/IP networking, the attacking packets can always arrive at the target server. Thus, the damage of DoS attack is decreasing in CCN but keeps the same effect in TCP/IP networking.

In this paper, we propose a DoS attack approach in CCN. Based on the queuing theory, we model the DoS attack and analyze the effect of the content caching of CCN on limiting DoS attack comparing with TCP/IP networking. Furthermore, we simulate the DoS attack in CCN compared with TCP/IP networking by NS2. Both the model evaluation and simulation results show that the damage of DoS attack in CCN is naturally decreasing step by step and CCN is more resistant than TCP/IP networking in term of preventing DoS attack due to the content

caching. Besides, we analyze how the content caching Time-To-Live (TTL) in CCN affects DoS attack with the evaluations of our DoS attack model and NS2 simulations.

The rest of this paper is organized as following. Section 2 presents some closely related work. Section 3 analyzes the effect of the CCN content caching to DoS attack. Section 4 models the DoS attack in CCN. Section 5 evaluates the effect of the content caching to DoS attack using blocking probability with the DoS attack model for CCN and simulates the effect in NS2. Besides, we analyze the different effect on limiting DoS attack of the different content caching TTLs. Finally, section 6 concludes this paper.

2. Related Work

2.1 DoS attack

There have been many researches on DoS attack. For example, TVA [12] is an architecture which is built on capability to prevent DoS attack, where senders only can send packets embedded the capabilities, which is provided by the receivers, to the destination. NetFence [13] uses a novel mechanism to secure congestion policing feedback and enable congestion policing inside the network and can be used to mitigate DoS attack to targeted victims and can also prevent the compromised senders and receivers from organizing into pairs to congest a network link. In [14], the authors study the optimization problem of finding a minimum set of target nodes to achieve a DoS attack and investigate the hardness of malicious attacks on multiple-tree topologies of push-based Peer-to-Peer streaming systems. The authors in [15] consider DoS attacks on DNS and propose a minor change in the caching behavior of DNS resolvers that can significantly alleviate the impact of such attacks. AITF [16] and Pushback [21] drop the unwanted packets by installing packet filters upstream from the destination. However, there is no clean way to distinguish malicious packets from other packets since the packets can be manufactured with any contents the attacker chooses, so these filters sometimes will block some legitimate traffic from the receiver. The authors in [17] propose a novel group testing (GT)-based approach deployed on back-end servers to identify application DoS attack, which aims at disrupting application service rather than depleting the network resource. In [18], the authors address the stealthy SIP flooding attack and propose a detection scheme based on the signal processing technique wavelet to identify the stealthy attack in its early stage for timely response. The authors in [19] propose a detection approach based on time-series decomposition, which divides the original time series into trend and random components, to detect stealthy DDoS attacks and the authors in [20] utilizes game theory to propose a series of optimal puzzle-based strategies for handling increasingly sophisticated flooding attack scenarios. The anomaly detection [22] classified traffic patterns as friendly or malicious depending on a particular end-to-end flow at the application level. It leads to a closed Internet that stifles innovations, because everything that is not completely standard will be probably treated as malicious.

These proposals are all attempting to mitigate DoS attack in current Internet using TCP/IP networking, but none of them refer to the DoS attack in CCN. In this paper, we focus on the effect of DoS attack in CCN, which is different from TCP/IP networking.

2.2 CCN

CCN, where Data (named content) is the first entity, is a promising architecture for future Internet. Recently, there have been several proposals on CCN. The authors in [29] explore the economic incentives for operators to deploy the routing polices in CCN. MDHT [30] is proposed as a hierarchical name resolution service for CCN and can support constant hop resolution. In [31], the authors investigate buffer management strategies and conclude that strategies with a high data refresh rate achieve the most efficient delivery and generate the smallest overhead. In [32], the authors study how to ameliorate the privacy of a user's content requests in CCN. Furthermore, the authors in [33] evaluated the suitability of existing software and hardware components of current routers for the support of CCN. In [34], the authors analyzed the bandwidth and storage sharing performance in CCN and provided an analytical characterization of statistical bandwidth and storage sharing.

All the proposals above study the CCN and do great contributions to CCN, but none of them concerns the DoS attack in CCN. In this paper, we analyze the effect of DoS attack in CCN comparing with TCP/IP networking and evaluate the survivability of CCN when suffering DoS attack.

3. Effect of Content Caching on Mitigating DoS Attack

DoS attack can still function in CCN, but it is restricted. In TCP/IP networking, the attacking packets sent by the botnet can continuously arrive at the target server and block serving legitimate users continuously. While as for

CCN, the attacking packets sent downstream the same CCN Router can only affect the target content repository(s) for a certain period, which is determined by the duration when the requested Data is finished caching in the CCN Router directly connected to the target content repository(s) (denoting as the First Class CCN Router, or FCC Router in short). When the FCC Router receives the requested Data, the content (the requested Data) will be cached in the Content Store of the FCC Router. When other members of the botnet downstream this FCC Router request the target content repository(s) for the same content, they will be answered by the FCC Router directly, without interaction with the target content repository(s). Thus, the effect of DoS attack will decrease gradually while the requested Data is cached in the FCC Routers one by one. In the end, when all the FCC Routers of the target content repository(s) receive the requested Data, all the DoS attacking Interest packets will be restricted downstream the FCC Routers and cannot affect the target content repository(s) any more. Here, we do not consider how the TTL of content caching affects the DoS attack in CCN. In fact, the effect of DoS attack will increase as the attacking packets increase while the TTL of content caching in CCN routers have been overdue one by one. We will analyze this effect simply in Section III.

Figure 3 illustrates the process described above.

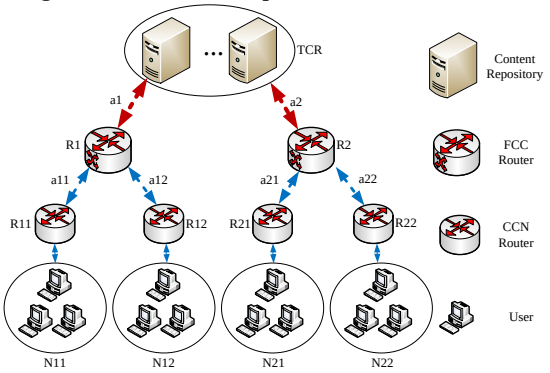

Fig. 3. Binary tree topology

In Figure 3, at the beginning of the DoS attack, the botnet *N11*, *N12*, *N21* and *N22* all flood attacking Interest packets to the *TCR* (target content repository(s)). All the attacking Interest packets can reach the *TCR* through the links *a1* and *a2*. But when the requested Data has been cached in *R1* (one of the FCC Routers), the attacking packets from the botnet *N11* and *N12* will reach *R1* to fetch the requested Data instead of the *TCR*. That is, the link *a1* does not have any attacking packets any more, which means that *N11* and *N12* lose efficacy for DoS attack. Similarly, *N21* and *N22* will lose efficacy when the requested Data has been cached in *R2*. Gradually, the DoS attack will be dispersed into all the CCN Routers such as *R11*, *R12*, *R21* and *R22* finally and its damage to CCN will be eliminated.

But the effect of DoS attack will increase if the TTL of content caching in CCN routers have been expired one by one. For example, if the content caching TTLs of *R11* and *R1* are both expired, the DoS attacking packets from *N11* will affect *TCR* again until the content caching of *R11* or *R1* are finished. Then we can conclude that CCN will more resistant to DoS attack if the content caching TTLs in CCN routers are set larger than the duration of DoS attack. (The details are in Section V.)

4. Modeling DoS Attack

In this section, we focus on the effect of the content caching of CCN to DoS attack and model the DoS attack in both CCN and TCP/IP networking.

4.1 Assumptions

1) Although the target content repository(s) may be in a large number, we believe the attacker can find a small enough number of target content repository(s) by narrowing the scope of the Content Name in Interest packet. In this paper, we focus on the effect of the content caching of CCN to DoS attack, while do not consider how the target content repository(s) organize(s). Thus, we set the number of the target content repository(s) to be 1 in both CCN and TCP/IP networking.

2) We assume the requested Data is cached in the FCC Router in a single step. In fact, the requested Data is cached gradually as the partitions of the requested Data are cached in the FCC Router one by one. This difference is reflected in the simulation section of this paper (Section 4), but not in the model.

3) The congestion of CCN Routers downstream the FCC Routers is not considered, because the target of DoS attack is the content repository (referred as server in TCP/IP networking) most of the time, rather than the links downstream the FCC Routers. Moreover, the latter cannot achieve the expected attacking damage. For example, the legitimate requesting users from any other links downstream the CCN Routers which is not suffering the DoS attack can still obtain the requested Data without any denial of service.

4) Both in TCP/IP networking and CCN, we assume the attacking Interest packets arrive at the target content repository following the Poisson distribution and the serving time for each packet is equal. And we suppose the size of the serving queuing in target content repository is m.

Besides, we assume the total arriving rate of the attacking Interest packet is λ and the serving rate of the target content repository is μ.

5) Assuming the attacker can make the zombies generate different Interest packets at a certain time, so all the Interest packets can be delivered to the target repository without PIT aggregation.

Then we model the DoS attack in both TCP/IP networking and CCN. Table 1 shows all the parameters values in the analysis.

Table 1: Paremeters for modeling DoS attack

Parameters	Definition
λ	total arriving rate
λ_i	arriving rate from the *i*th malicious FCC Router
μ	serving rate of the target content repository
t_i	duration from the (*i*-1)th MFR having finished caching to the *i*th MFR finishing caching requested Data
m	total serving capacity of the content repositories (servers), also referred as the size of the serving queue

4.2 TCP/IP Networking

Based on the above assumptions, the arriving pattern of DoS attacking packets can be described by the *M*/*G*/*1*/*m* queuing model, where the first parameter denotes the arriving rate following the Poisson distribution, the second parameter denotes the serving time is equal for each packet, the third parameter denotes the number of the servers (referred as content repositories in CCN) is 1 and the last parameter denotes the total serving capacity of the servers is *m*. From the *M*/*G*/*1*/*m* model, the blocking probability (BP_1) of TCP/IP networking is:

$$BP_1 = \frac{(\lambda/\mu)^m}{\sum_{n=0}^{m}(\lambda/\mu)^n} \quad (1)$$

4.3 CCN

In CCN, the content caching mechanism makes the effect of DoS attack different from the TCP/IP networking. The FCC Router forwarding attacking Interest packets is called Malicious FCC Router (MFR in short) and the number of the MFRs is *w*. We denote t_i ($i \geq 1$) (Figure 3) as the duration from the (*i*-1)th MFR having finished caching to the *i*th MFR finishing caching the requested Data. We assume that the arriving rate of the attacking Interest packet for the *i*th MFR is λ_i, so we can deduce that:

$$\sum_{i=1}^{w}\lambda_i = \lambda \quad (2)$$

At the beginning of the DoS attack in CCN, the attacking Interest packets arrive at the target content repository following the Poisson distribution at a rate λ, which fits Eq. (2). When the requested Data is cached in the *i*th MFR, the arriving rate of the attacking packets is shown as follows.

$$\lambda(i) = \lambda - \sum_{k=1}^{i}\lambda_i \quad (i = 1,2,...,w) \quad (3)$$

With the requested Data is cached in different MFRs gradually, the arriving rate is changing as the time goes. Nevertheless, the pattern of the attacking Interest packets arriving at the target content repository is still a Poisson distribution in each time scope t_i.

- Without considering the TTL of content caching

In this situation, how the TTL of content caching affects the DoS attack in CCN is not taken in consideration. Thus, when the requested Data is finished caching in the *i*th MFR, the attacking packets downstream from the 1st to the *i*th MFR will never reach the target content repository again. However, parts of the attacking packets from the sub-botnet downstream the 1st to the *i*th MFR that are not be processed by the target content repository can still remain in the serving queue and consume the serving capacity of the target content repository (we call these packets as remaining attacking packets) and the duration for processing these remaining attacking packets for each t_i is denoted as t_{ii}.

We illustrate the relation of the t_{ii} (*i*=2,3,…,w) and t_i (*i*=1,2,…,*w*) in Figure 4. And we denote the musters of t_i (*i*=1,2,3,…,w) and t_{ii} (*i*=2,3,…,w) are {t_i} (*i*=1,2,3,…,w) and {t_{ii}} (*i*=2,…,*w*) respectively.

Fig. 4. The relation of the tii and ti

Assuming the number of the remaining attacking packets is *s* and the target content repository can totally finish processing the attacking packets downstream the *i*th MFR before the (*i*+1)th MFR is finished caching the requested Data, where *i*=1,2,…,*w*. That is, for each time scope t_i, the remaining attacking packets is from the (*i*-1)th MFR only.

Since the arriving rate λ_i is constant in steady state for each t_i, the number of the remaining attacking packets in the serving queue of the target content repository at t_{ii} is in direct proportion to the arriving rate λ_{i-1}. Thus, we can deduce s_i for each t_{ii} (*i*=2,…,*w*) as following:

$$s_i = \left(\lambda_{i-1} \Big/ \left(\sum_{k=i-1}^{w}\lambda_k\right)\right)m \qquad (i = 2,3,...,w) \quad (4)$$

$$a_i = \lfloor s_i \rfloor \leq m \quad (5)$$

Denoting l is the average size of the packet, from Eq. (5) we deduce the duration for processing the remaining attacking packets, that is t_{ii}, fits the inequality below,

$$0 \le t_{ii} \le m/(\mu/l) \tag{6}$$

After t_{ii} the target content repository has finished processing the remaining attacking packets, its serving capacity (the size of the serving queue) returns to m. Thus, the serving capacity (serving queue) for each t_i except t_{ii} is m.

Based on the above analysis and using the $M/G/1/m$ queuing model in $\{t_i\}$ except $\{t_{ii}\}$, we use the Markov chain to analyze the state transition for the target content repository. Since the serving pattern is FIFO queue mechanism, the serving rate is a constant μ. Thus, the state transition for the target content repository is as Figure 5, where δ is the coefficient of the transition probability.

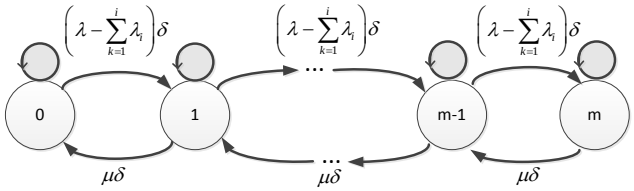

Fig. 5. State transition without considering content caching TTL

From Figure 5, we can deduce that,

$$\left(\lambda - \sum_{k=1}^{i}\lambda_i\right) p_{n-1} = \mu p_n \qquad n = 1,2,...,d \tag{7}$$

$$p_n = p_0 \left(\frac{\lambda - \sum_{k=1}^{i}\lambda_i}{\mu}\right)^n \qquad n = 1,2,...,d \tag{8}$$

$$\sum_{n=0}^{\infty} p_n = 1 \tag{9}$$

From Eq. (7) to Eq. (9), we can deduce the blocking probability (BP_2) in CCN when suffering DoS attack is,

$$BP_2 = \frac{\left(\left(\lambda - \sum_{k=1}^{i}\lambda_i\right)\Big/\mu\right)^m}{\sum_{n=0}^{m}\left(\left(\lambda - \sum_{k=1}^{i}\lambda_i\right)\Big/\mu\right)^n}, \quad \left\{\begin{array}{l} t \in \{t_i\} \quad and \quad t \notin \{t_{ii}\}, \\ and \quad i = 1,2,3,...,w \end{array}\right\} \tag{10}$$

When in $\{t_{ii}\}$, the blocking probability (BP_3) is,

$$\frac{\left(\left(\lambda - \sum_{k=1}^{i}\lambda_i\right)\Big/\mu\right)^m}{\sum_{n=0}^{m}\left(\left(\lambda - \sum_{k=1}^{i}\lambda_i\right)\Big/\mu\right)^n} \le BP_3 \le \frac{\left(\left(\lambda - \sum_{k=1}^{i-1}\lambda_i\right)\Big/\mu\right)^m}{\sum_{n=0}^{m}\left(\left(\lambda - \sum_{k=1}^{i-1}\lambda_i\right)\Big/\mu\right)^n} \tag{11}$$

- Considering the TTL of content caching

In this situation, when the requested Data is finished caching in the ith MFR, the attacking packets downstream from the 1st to the ith MFR will not reach the target content repository again until the content caching TTLs are expired. That is, the content caching TTLs can affect the DoS limiting effect.

Suppose the number of the CCN routers whose content caching TTLs have been expiration is j at time click i ($j<i$), and the packet arriving rate of the rth CCN router is v_r, then the state transition for the target content repository for each time scope t_i is as Figure 6, where δ is the coefficient of the transition probability.

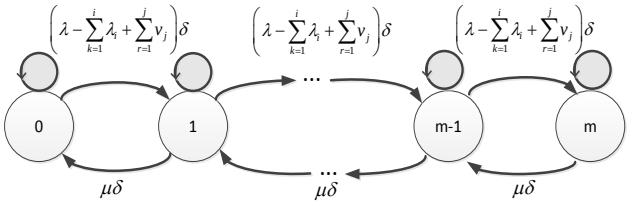

Fig. 6. State transition with considering content caching TTL

From Figure 6 we can deduce that,

$$\left(\lambda - \sum_{k=1}^{i}\lambda_i + \sum_{r=1}^{j} v_j\right) p_{n-1} = \mu p_n \quad n = 1,2,...,m \tag{12}$$

$$p_n = p_0 \left(\frac{\lambda - \sum_{k=1}^{i}\lambda_i + \sum_{r=1}^{j} v_j}{\mu}\right)^n \quad n = 1,2,...,m \tag{13}$$

From Eq. (12), Eq. (13) and Eq. (9), we can deduce the blocking probability (BP_4) is,

$$BP_4 = \frac{\left(\left(\lambda - \sum_{k=1}^{i}\lambda_i + \sum_{r=1}^{j} v_j\right)\Big/\mu\right)^m}{\sum_{n=0}^{m}\left(\left(\lambda - \sum_{k=1}^{i}\lambda_i + \sum_{r=1}^{j} v_j\right)\Big/\mu\right)^n}, \quad with \left\{\begin{array}{l} t \in \{t_i\} \quad and \quad t \notin \{t_{ii}\}, \\ and \quad i = 1,2,3,...,w \end{array}\right\} \tag{14}$$

In Eq. (14), we do not consider the effect of the minor duration of $\{t_{ii}\}$ (i=2,…,w).

From Eq. (14), we can deduce that if the content caching TTLs are long enough that the attacking packets will not arrive at the target content repository again since the caching has been finished once, the blocking probability will be smallest because the $\sum_{r=1}^{j} v_j$ will be 0 in this situation.

5. Model Evaluation and Simulation

In this section, we evaluate the DoS attack in CCN with our analytical model (proposed in Section III) and simulate the DoS attack in NS2. The grim damage of DoS attack is due to its extremely high rate of packets to totally consume the capacity of the victim. Thus, we simulate the attacking effect of DoS by sending high rate of packets from a large number of computers to the target content repository and analyze its blocking probability for legitimate users.

5.1 Topology and Parameters

Figure 7 shows the simulation topology. *U1* is a legitimate user who requests the target content repository (*TCR*) normally. The botnet which is the source of DoS attack is composed of *N1*, *N2* and *N3*. The FCC Routers *R1* to *R3* are the MFRs which are directly connected to the *TCR*.

The total serving capacity of the *TCR* is 20Mb/s with the queuing size of 500 packets, and the bandwidths of *R1* to *R3* are 100Mb (preventing the link congestion and focusing on the target content repository itself). The packets arrive at the TCR with a rate following the Poisson distribution. N_i (i=1, 2, 3) is respective composed of 10 computers. The legitimate user *U1* sends requesting packets directly to the *TCR* by 500kb/s following the Poisson distribution, but the sub-botnets downstream the *Ri* (i=1, 2, 3) flood attacking packets with 5Mb/s, 4Mb/s and 4Mb/s following the Poisson distribution. And the size for each packet is 1000 bytes. The natural content caching mechanism of CCN makes the attacking packets downstream *Ri* can only arrive at the *TCR* in a certain period ($\sum_{j=1}^{i} t_j$). In model evaluation and NS2 simulation, we satisfy this character of CCN by setting t_1=50*s*, t_2=60*s* and t_3=40*s*, which means that the attacking times for *N1*, *N2* and *N3* are 50*s*, 110*s* and 150*s*. And we set the period before DoS attack is 20*s* and the total simulation time is 200*s*.

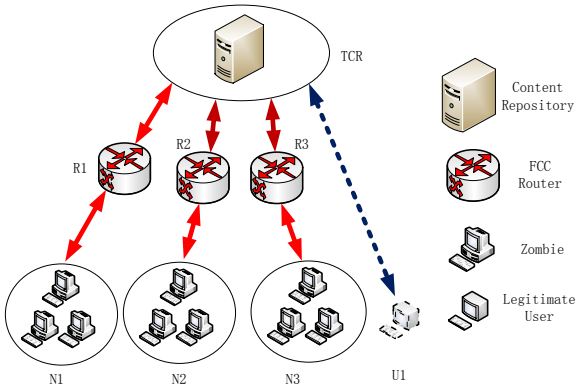

Fig. 7. Simulation topology

5.2 Results of Model Evaluation and NS2 Simulation

- Without considering the TTL of content caching

From Eq. (6), we can deduce that $t_{22} \leq 25ms$, $t_{33} \leq 25ms$ and $t_{44} \leq 25ms$. Such a minor duration can be ignored when compared with the total 200*s* evaluation time.

According to Eq. (1), Eq. (10) and Eq. (11), we evaluate the blocking probability of the target content repository in both TCP/IP networking and CCN. The model evaluation result without considering the TTL of content caching is shown in Figure 8 (the dotted line marked by '-' and '.' respectively represent the blocking probability in TCP/IP networking and CCN).

Moreover, we simulate the DoS attack to the target content repository in CCN without considering the TTL of content caching for 200 times with NS2 and the average results are also shown in Figure 8 (the lines marked by the square and '+' respectively represent the blocking probability in TCP/IP networking and CCN).

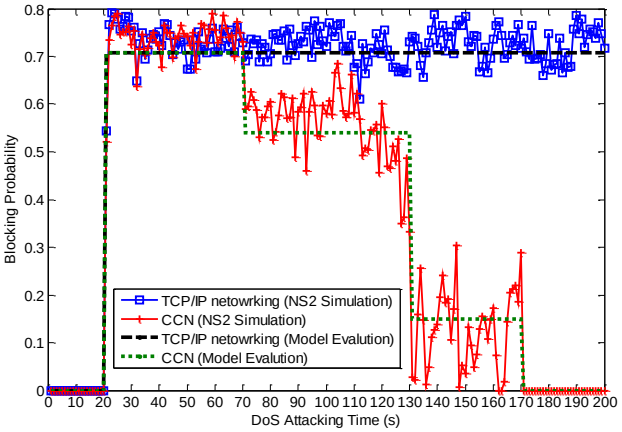

Fig. 8. The blocking probability without considering content caching TTLs

Figure 8 shows the evaluation results of our DoS attack model for CCN (Eq. (10)) match the simulation results well. When suffering DoS attack, the blocking probability in CCN is decreasing step by step while it keeps a certain value in TCP/IP networking.

At the beginning of the DoS attack (from 20*s* to the 70*s*), the blocking probability is about 0.71 in both TCP/IP working and CCN. And the blocking probability in CCN decreases gradually because the requested Data reaches *R1* to *R3* one by one (decreasing at the 70*s*, the 130*s* and the 170*s*). That is, when the requested Data is finished caching in *R1* at 70*s*, the attacking of the sub-botnet downstream it loses effectiveness, and the blocking probability decreases from 0.71 to 0.54. This is because the attacking packets of the sub-botnet from *N1* are satisfied by *R1* directly and cannot arrive at the target content repository again. Analogously, when the requested Data is finished caching in *R2* at 130*s*, the blocking probability decreases from 0.54 to 0.15. And when *R3* finishes caching at 170*s*, the blocking probability decreases to a very low value (about zero) and all the DoS attack packets cannot reach the target content repository any more.

Nevertheless, the blocking probability in TCP/IP networking keeps constant at about 0.71 from the beginning of the DoS attack (at 21*s*) to the end (at 200*s*).

Thus, we can conclude that CCN is more resistant than TCP/IP networking in preventing the DoS attack due to the content caching.

- With considering the TTL of content caching

From Figure 8, *R1* is the first one that finishes content caching at 70*s* while *R2* is the second one at 130*s* and *R3* is the last one at 170*s*.

Thus, if the TTL of content caching in *R1* is larger than 130*s* (from 70*s* to 200*s*), whose value is the total attacking time minus the time the first MFR (*R1* in Figure 7) finishes content caching, the attacking Interest packets from *N1* will never arrive at *TCR* again. Analogously, the attacking Interest packets from *N2* and *N3* will never arrive at *TCR* again if their content caching TTLs are larger than 70*s* (from 130*s* to 200*s*) and 30*s* (from 170*s* to 200*s*) respectively.

To evaluate the effect of the content caching TTL on DoS attack, we set the content caching TTLs of all the CCN Router in Figure 7 to be 10*s* (<30*s*) and 300*s* (>130*s*) respectively.

According to Eq. (14), we evaluate the blocking probability of the target content repository in CCN. Moreover, we simulate the DoS attack to the target content repository with considering the TTL of content caching for 200 times with NS2. The average results of both model evaluation and NS2 simulation are shown in Figure 9. (The dotted line marked by '-' and '.' respectively represent the blocking probability in model evaluation with different content caching TTLs to be 10*s* and 300*s* respectively. And the lines marked by the square and '+' represent the blocking probability in NS2 simulation in the same scenario.)

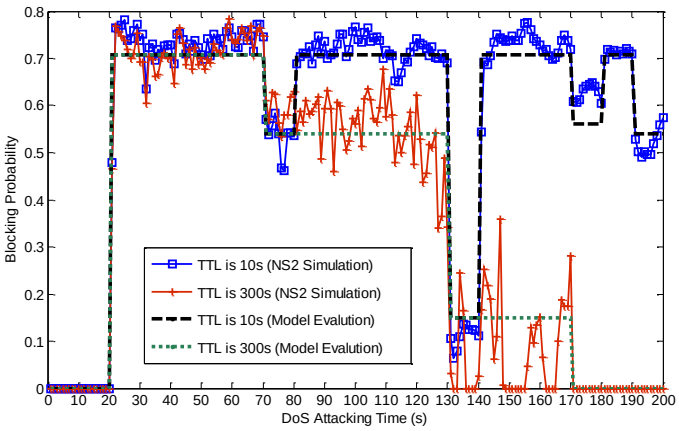

Fig. 9. The blocking probability with different content caching TTLs

Figure 9 shows the evaluation results of our DoS attack model for CCN (Eq. (14)) matches the simulation results well and different content caching TTLs bring into different DoS limiting effects.

From Figure 9, when the content caching TTL sets to be 300*s* (>130*s*), the blocking probability decreases at 70*s*, 130*s* and 170*s* as *R1*, *R2* and *R3* finish content caching at the three time points respectively. That is, the blocking probability in CCN will decrease step by step as the CCN routers finish content caching one by one with the large enough content caching TTL.

When the content caching TTL is set to be 10*s* (<30*s*), the blocking probability decreases at 70*s* but increases at 80*s* again, because the content caching is emptied at 80*s* (TTL is 10*s*), which means that the attacking Interest packets from N1 will be able to arrive at *TCR* again. At 130*s*, *R2* finishes content caching while *R1* finishes content caching the second time, so the blocking probability decreases again. But when the content caching TTLs of *R1* and *R2* are both expired at 140*s*, the blocking probability increases the second time. When *R3* finishes content caching at 170*s*, the blocking probability decreases the third time. When the content caching TTL of *R3* is expired at 180*s*, the blocking probability increases the third time. At 190*s*, *R1* finishes content caching the third time but neither *R2* nor *R3* finishes content caching at this time, so the blocking probability decreases the last time.

In Figure 9, when the content caching TTL is set to be 10*s* (<30*s*), the blocking probability almost always keep a high value about 0.7 most of the time except for some time scope such as from 70*s* to 80*s*. But when the content caching TTL is set to be 300*s* (<130*s*), the blocking probability simply decreases step by step (from 0.7 to 0.54, then from 0.54 to 0.15, and decreasing to 0 at last).

Thus, we can conclude that the DoS limiting effect is preferable if the content caching TTL sets to be larger than total attacking time minus the time the first MFR finishes

content caching.

6. Conclusion

In this paper, we propose a DoS attack model for CCN. Comparing with TCP/IP networking, we analyze the effect of CCN's content caching at limiting DoS attack in both model evaluations and NS2 simulations, using the blocking probability of the target content repository as the metric. When suffering DoS attack, the blocking probability of the target content repository in CCN is lower than in TCP/IP networking. It means the content caching makes CCN more resistive in limiting DoS attack and has better survivability and resilience than TCP/IP networking. Besides, we analyze how the content caching TTL limits DoS attack with evaluations of our DoS attack model and NS2 simulations. The results show that the larger content caching TTL can make the blocking probability keep lower for a relative longer time, and hence bring in better effectiveness at limiting DoS attack. In addition, we only analyze the effect of CCN's content caching on mitigating DoS attack. Our future work is to motivate an effective DoS limiting approach for CCN.

Acknowledgments

This research is supported by the National High-Tech Research and Development Program of China (863) (2011AA010701), and supported by the Fundamental Research Funds for the Central Universities (2011YJS205), supported in part by the National Natural Science Foundation of China (NSFC) (61271202, 61003283, 61001122, 61232017, 61102049) and Beijing Natural Science Foundation (4122060).

References

[1] Worm.ExploreZip,http://www.symantec.com/avcenter/venc/data/worm.explore.zip.html, 2010.

[2] P.R. Roberts, Zotob Arrest Breaks Credit Card Fraud Ring, http://www.eweek.com/article2/0,1895,1854162,00.asp, 2010.

[3] Z.S. Chen, L.X. Gao, and K. Kwiat, "Modeling the Spread of Active Worms", in Proceedings of IEEE INFOCOM, 2003, vol. 3, pp. 1890-1900.

[4] D. Moore, C. Shannon, and J. Brown, "Code-Red: A Case Study on the Spread and Victims of an Internet Worm", in Proceedings of Second Internet Measurement Workshop (IMW), 2002, pp. 273-284.

[5] W. Yu, X. Wang, P. Calyam, D. Xuan, and W. Zhao, "Modeling and Detection of Camouflaging Worm", IEEE Transactions on Dependable and Secure Computing, Vol. 8, Issue. 3, 2011, pp. 377–390.

[6] K. Wang, H.B. Luo and Y.J. Qin, "Identifier/locator Separation: A Worm Detection and Prevention Perspective", in Proceedings of 2011 International Conference on Advanced Intelligence and Awareness Internet (AIAI2011), 2011, pp. 7-11.

[7] X. Wang, W. Yu, A. Champion, X. Fu, and D. Xuan, "Detecting Worms via Mining Dynamic Program Execution", in Proceedings of the third International Conference on Security and Privacy in Communications Networks and the Workshops (SecureComm), 2007, pp. 412-421.

[8] Y.F. Ye, T. Li, Q.S. Jiang, and Y.Y. Wang, "CIMDS: Adapting Postprocessing Technique of Associative Classification for Malware Detection", IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews, Vol. 40, Issue. 3, 2010, pp. 298–307.

[9] C. Wright, S. Coull, and F. Monrose, "Traffic Morphing: An Efficient Defense Against statistical Traffic Analysis", in Proceedings of the 16th Network and Distributed Security Symposium, 2009, pp. 237-250..

[10] Y.Y. Zeng, X. Hu, and K.G Shin, "Detection of Botnets Using Combined Host- and Network-Level Information", in Proceedings of IEEE/IFIP Conference on Dependable System & Networks (DSN), 2010, pp. 291-300,.

[11] M. Ajelli, R.L. Cigno and A. Montresor, "Modeling Botnets and Epidemic Malware", in Proceedings of IEEE International Conference on Communications (ICC), 2010, pp. 1-5.

[12] X.W. Yang, D. Wetherall, and T. Anderson, "TVA: A DoS-Limiting Network Architecture", IEEE/ACM Transactions on Networking (ToN), Vol. 16, No. 6, 2008, pp. 1267–1280.

[13] X. Liu, X.W. Yang and Y. Xia, "NetFence: Preventing Internet Denial of Service from Inside Out," in Proceedings of ACM SIGCOMM, 2010.

[14] S. Grau, M. Fischer, M. Brinkmeier and G. Schäfer, "On Complexity and Approximability of Optimal DoS Attacks on Multiple-Tree P2P Streaming Topologies", IEEE Transactions on Dependable and Secure Computing, Vol. 8, Issue. 2, 2011, pp. 270-281.

[15] H. Ballani and P. Francis, "Mitigating DNS DoS Attacks", in Proceedings of the 15th ACM conference on Computer and communications security (CCS), 2008, pp. 189-198.

[16] K. Argyraki and D. Cheriton, "Active Internet Traffic Filtering: Real-Time Response to Denial-of-Service Attacks", in the Proceedings of the 2005 USENIX Annual Technical Conference, 2005.

[17] X. Ying, S. Incheol, M.T. Thai, T. Znati, "Detecting Application Denial-of-Service Attacks: A Group-Testing-Based Approach", IEEE Transactions on Parallel and Distributed Systems, Vol. 21, Issue. 8, 2010, pp. 1203-1216.

[18] J. Tang and Y. Cheng, "Quick Detection of Stealthy SIP Flooding Attacks in VoIP Networks", in Proceedings of IEEE International Conference on Communications (ICC), 2011, pp. 1-5.

[19] H. Liu and M.S. Kim, "Real-Time Detection of Stealthy DDoS Attacks Using Time-Series Decomposition", in Proceedings of IEEE International Conference on Communications (ICC), 2010, pp. 1-6.

[20] S. Grau, M. Fischer, M. Brinkmeier and G. Schäfer, "A Puzzle-Based Defense Strategy Against Flooding Attacks Using Game Theory", IEEE Transactions on Dependable and Secure Computing, Vol. 7, Issue. 1, 2010, pp. 5-19.

[21] J. Ioannidis and S. Bellovin, "Implementing Pushback: Router-Based Defense Against DoS Attacks", in Proceedings of NDSS, 2002.

[22] A. Hussain, J. Heidemann, and C. Papadopolous, "A Framework for Classifying Denial of Service Attacks", in Proceedings of ACM SIGCOMM, 2003, pp. 99-110.

[23] V. Jacobson, D.K. Semtters, J.D. Thornton and M.F. Plass, "Networking Named Content", in Proceedings of CoNEXT, 2009, pp. 1-12.

[24] V. Jacobson, D. K. Smetters, N. H. Briggs, M. F. Plass, P. Stewart, J. D. Thornton, R.L. Braynard, "VoCCN: Voice Over Content-Centric Networks", in Proceedings of ReArch, 2009, pp. 1-6.

[25] M. Gritter and D. Cheriton, "An Architecture for Content Routing Support in the Internet", in Proceedings of the 3rd conference on USENIX Symposium on Internet Technologies and Systems, SFO, USA, Mar. 2001.

[26] M. Caesar, T. Condie, J. Kannan, K. Lakshminarayanan, I. Stoica, and S. Shenker. "ROFL: Routing on Flat Labels", in Proceedings of ACM SIGCOMM, 2006, pp. 363-374.

[27] Named data networking. http://www.named-data.net/.

[28] T. Koponen, M. Chawla, B. Chun, A. Ermolinskiy, K. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture", in Proceedings of ACM SIGCOMM, 2007, pp. 181-192.

[29] S. DiBenedetto, Ch. Papadopoulos and D. Massey, "Routing Policies in Named Data Networking", in Proceedings of ACM Sigcomm Workshop on Information-Centric Networking (ICN), 2011, pp. 38-43.

[30] M. D'Ambrosio, Ch. Dannewitz, H. Karl and V. Vercellone, "MDHT: A Hierarchical Name Resolution Service for Information-centric Networks", in Proceedings of ACM Sigcomm Workshop on Information-Centric Networking (ICN), 2011, pp. 7-12.

[31] F. Bjurefors, P. Gunningberg, C. Rohner and S. Tavakoli, "Congestion Avoidance in a Data-Centric Opportunistic Network", in Proceedings of ACM Sigcomm Workshop on Information-Centric Networking (ICN), 2011, pp. 32-37.

[32] S. Arianfar, T. Koponen, B. Raghavan and S. Shenker, "On Preserving Privacy in Content-Oriedted Networks", in Proceedings of ACM Sigcomm Workshop on Information-Centric Networking (ICN), 2011, pp. 19-24.

[33] D. Perino and M. Varvello, "A reality check for content centric networking", in Proceedings of ACM Sigcomm Workshop on Information-Centric Networking (ICN), pp. 44-49, August 2011.

[34] G. Carofiglio, L. Muscariello, and M. Gallo, "Bandwidth and storage sharing performance in information centric networking", in Proceedings of ACM Sigcomm Workshop on Information-Centric Networking (ICN), 2011, pp. 26-31.

[35] W.K Chai, N. Wang, I. Psaras, G. Pavlou, Ch. J. Wang, G.G. de Blas, F.J. Ramon Salguero, L. Liang, S. Spirou, A. Beben and E. Hadjioannou, "CURLING: Content-Ubiquitous Resolution and Delivery Infrastructure for Next-Generation Services", IEEE Communications Magazine, vol. 49, Issue. 3, 2011, pp. 112-120.

[36] K. Kuppusamy and S. Malathi, "Prevention of attacks under DDoS using target customer behavior", International Journal of Computer Science Issues, Vol. 9, Issue 5, No.2, 2012, pp. 301-307.

[37] M. Bekravi, S. Jamali and G. Shaker, "Detection of Denial of Service attacks against Domain Name System using neural networks", International Journal of Computer Science Issues, Vol. 6, No. 1, 2009, pp. 23-27.

[38] M. Bekravi, S. Jamali and G. Shaker, "Defense against SYN-flood Denail of Service attacks based on learning automata", International Journal of Computer Science Issues, Vol.9, Issue 3, No.1, 2012, pp. 514-520.

Kai Wang received the B.S. degree in electrical engineering from Beijing Jiaotong University, China, in 2009. He is pursuing the Ph.D. degree at national engineering laboratory for next generation Internet interconnection devices, Beijing Jiaotong University. His research interests include network security and next generation Internet technology.

Jia Chen received her BEng degree in Communication Engineering in 2005 from Beijing University of Posts and Telecommunications in China. She received Master of Research (MRes) degree in Telecommunications in 2006, and the Ph. D degree in Electrical and Electronic Engineering 2010 from Department of Electrical and Electronic Engineering, University College London, UK. She worked in British Telecommunication (UK) for an industry fellowship position from Jan. 2009 to Apr. 2009. She joined Beijing Jiaotong University (Beijing, China) as a lecturer since July 2010. Her current research interests include architecture and protocol design and analysis for the future Internet.

Huachun Zhou received his B.S. degree from People's Police Officer University of China in 1986, and the M.S. and Ph.D. degree from Beijing Jiaotong University of China in 1989 and 2008, respectively. He is currently a professor with the Institute of Electronic Information Engineering, Beijing Jiaotong University of China. His main research interests are in the area of mobility management, mobile and secure computing, routing protocols, network management technologies and database applications.

Yajuan Qin received her B.S. and M.S. degrees in Electrical Engineering from the University of Elec-tronic Science and Technology of China (formerly known as Chengdu Institute of Radio Engineering) in 1985 and 1988, respectively, and Ph.D. degree in communication engineering from Beijing University of Posts and Telecommunications in 2003. From January 2002 to April 2002, she was a research associate at CRL, Japan. In 2003, she joined the School of Electronic and Information Engineering, Beijing Jiaotong University, where she is a full professor. She has published more than 30 research papers and is the holder of more than ten patents. Her research interests are in the areas of computer networks and wireless communications.