Security Monitoring for Content-Centric Networking

David Goergen, Thibault Cholez, Jérôme François, Thomas Engel

Security Monitoring for Content-Centric Networking

David Goergen, Thibault Cholez, Jérôme François, and Thomas Engel

SnT - University of Luxembourg, Luxembourg,
firstname.lastame@uni.lu

Abstract. Content-Centric Networking (CCN) is one of the most promising research area for a future Internet. The goal is to obtain a more scalable, secure, collaborative Internet supporting context-aware services. However, as a new overlay infrastructure, CCN raises the need of a new monitoring architecture to assess security of CCN devices. In particular, the stateful nature of CCN routers introduces new attack threats that need to be addressed. We propose in this paper a monitoring approach for the instrumentation of CCN enabled network nodes. The rationale of our monitoring approach is demonstrated through real experimentations to detect and mitigate network level attacks against CCN.

1 Introduction

Content-Centric Networking (CCN) is a new routing paradigm developed at PARC by Van Jacobson et al [14] but also known as Named Data Networking at a larger scale [20]. Based on the observation that today's communications are more oriented toward content retrieval (Web, P2P, etc.) than point-to-point communications (VoIP, IM, etc.)[18], CCN proposes to deeply revise the Internet architecture to best match its current usage. In a nutshell, contents are addressable, routable and authenticated, while their locations do not matter anymore. They can be replicated and stored (especially popular contents) on any CCN node. People looking for a content can securely retrieve it from the best locations available.

On one side, the client-server architecture needs more and more investments in expensive content delivery networks and server farms to be scalable. Agreements between ISPs (Internet Service Providers) and content providers tend to benefit to big web-actors that centralize user-generated contents. On the other side, the P2P paradigm makes an inefficient use of resources being mostly unaware of the physical location of the peers. In this context, we think that CCN could be an answer to the challenges that Internet will face in the near future and deserves research efforts from the community to properly investigate the applicability of this paradigm.

From a management point of view, Content-Centric Networking introduces new challenges. Firstly, it is hard for a content provider to monitor and control the diffusion of its content over the network after the initial release which leads to accountability issues because content can be distributed from any CCN node without requesting the original provider. While it is an important quest from a management point of view, we focus on the security aspects which present a more critical drawback of early deployment of CCN. Secondly, CCN routers are stateful as the route between a content and the requester has to be memorized. This stateful nature can lead to new possible DoS (Denial of Service) attacks which exploit the CCN routers limited memory size and these kind of attacks must be detected and mitigated. In this paper, we address this second issue because we think that security issues could highly decrease the appeal of the technology and reduce further research efforts. To that end, we (1) model a DoS attacks and (2) propose a Monitoring Architecture for Content-Centric Networking which is able to detect them.

Section 2 gives a brief overview of Content Centric Networking and its possible attacks. Section 3 describes our monitoring architecture adapted to CCN devices and our strategy to detect malicious traffic based on Support Vector Machines (SVM). Section 4 presents the experiments assessing the detection of real network attacks by the monitoring architecture. Finally, Section 5 presents the related work and Section 6 concludes the paper and outlines future work.

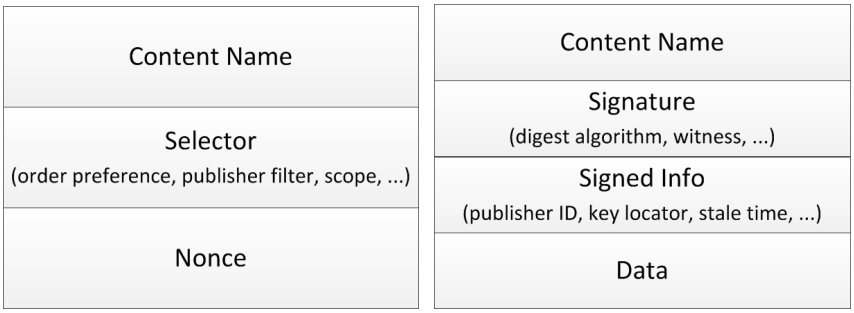
2 CCN background

2.1 CCN paradigm

The main idea behind CCN (also called Named Data Networking[3]) is a paradigm shift towards content oriented networking and routing. Today's Internet relies on the well established communication paradigm in which two end-points communicate over a network. However, regarding the behavior and habits of today's users, there is a strong shift towards content and not the location where this content is stored. Today, Internet is becoming more and more a content distribution network. Therefore some claim[11] that the best approach is to start from scratch building a new Internet architecture. However this involves long term investments for ISPs and CCN can thus be used over the already established architecture (for example CCN over IPv6).

2.2 CCN Node Model

CCN has two main types of packets, *Interest* and *Data* as seen in Figure 1. A user who wants to access a certain content sends out an *Interest* packet, specifying the name of the content (as defined by CCN nomenclature ContentName) to all its available faces. A Face can be anything which can serve as medium for transmitting and receiving data. A node which receives this packet and that can 'satisfy' the *Interest* sends out the corresponding *Data* packet onto the face from which it received the *Interest*. By definition, CCN nodes are stateful and only send *Data* if there was an *Interest* beforehand.



(a) Interest Packet (b) Data Packet

Fig. 1: CCN packet structures

Data can only 'satisfy' a specific *Interest* if the ContentName of the *Interest* packet is a prefix of the *Data* packet. CCN names are defined in [14] as *opaque, binary objects composed of an (explicitly specified) number of components*. This structure allows a fast and efficient prefix-based lookup similar to the IP lookup currently used. It also allows names to be context dependent *i.e. /ThisRoom/Printer* references a printer in the current room. This context-naming could make possible efficient context-aware service discovery in the future Internet of Things.

CCN nodes are composed of three main table structures which handle the forwarding of packets. At the arrival of an *Interest* packet on any given face, the engine performs a longest-match lookup on its structures and action is taken depending on the lookup result. The first structure to be searched is the Content Store. It can be seen as a buffer memory of past *Data* packets on the current router. IP routers also have such a buffer but it is purged once the packet is forwarded. The Content Store however

preserves the *Data* packet based on LRU (Least-Recently-Used) scheme and enables therefore a fast retrieval of currently popular demands. If there is a match, the router forwards its local copy of the content to the face on which it received the *Interest* and updates its Content Store accordingly.

If there is no match in the Content Store, the lookup is launched on the next structure which is the PIT. The PIT stands for Pending Interest Table and keeps record of *Interests* waiting to be resolved upstream by other content source(s). If a received *Interest* matches an entry in the PIT, the engine compares the faces recorded for that entry. If there is already one existing, no update is made. Otherwise, the face from which the *Interest* was emitted is simply added to the list of already waiting faces.

If no match-up is found in the PIT then the engine searches in its last structure: the FIB. The Forward Information Base keeps record of potential content source(s) and works similarly to its IP counterpart except that it stores a list of possible providers for a given name rather than a single one only. If a match is found, the engine then creates a PIT entry for the given *Interest* and it is forwarded to all faces specified in the FIB entry. If no match could be made, it means that the current router has no information on the demanded content and discards the *Interest*.

CCN has also built-in strategy and security layers. The strategy layer is used to define policies to select which face is the best for given contents. In fact, due to its design, FIB entries contain multiple faces. CCN can send periodically *Interests* to all outgoing faces without fearing of loops and thereby testing which of the faces responds the fastest. This one will be used as preferred until another round of this experiment yields to a different result. Criteria for experimentation interval can be a threshold of packets sent, a time out, change of the SSID(Service Set Identifier), etc.

The security layer ensures that the content received by a previously announced *Interest* is authentic. As in CCN only the content matters but not the route it takes, the only thing which needs to be checked for authenticity, consistency and integrity is the content itself which reversely means that end-to-end encryption is not needed any more. Key management is another issue often discussed. In [14, 13], several solutions are discussed which range from a PKI to PGP like web-of-trust.

2.3 Threats description

Content Centric Networking improves the security of Internet communications in many ways. First of all, CCN messages can not be sent toward a node without any prior *Interest* request from that node which makes the classical denial of service scheme inefficient as the attacker would need his target to generate a lot of *Interests* to enable the DoS attack. Also, CCN strongly relies on cryptography to authenticate the contents so that users can clearly know who emitted the content and can discard those from untrustworthy sources to avoid malware. If CCN improves security in some points, it also raises the possibility of new kind of attack. Unlike a terminal host which is less exposed to attacks, CCN routers are more vulnerable than IP routers because of their stateful nature and the management of their inner tables from which result their performance and quality of service.

By focusing on CCN routers, new kinds of Denial of Service can be performed. We categorize these attacks regarding the tables they target.

PIT attack A first attack can be focused on the Pending Interest Table. This table is critical because of the stateful routing of CCN network. If an attacker can manage to fill the PIT with a lot of forged *Interests*, legitimate *Interests* might be dropped, resulting in the denial of service of the pending communications. The attack is easy to achieve from the technical aspects. The attacker only needs to generate a lot of *Interests* whatever is the requested content in order to create entries in its PIT. Such an attack would benefit from distributed attack sources that would make it more difficult to detect. Therefore, the monitoring of the PIT to avoid flooding is a critical point for a safe and efficient CCN infrastructure.

FIB attack Unlike the IP address space, the CCN address space is not clearly bounded as domains are defined through strings rather than a small IP prefix. Therefore, a possible attack consists in generating and advertising a lot of contents belonging to different domains in order to fill the FIB on a face of the router. In that way, new legitimate domains can not be routed through the CCN device which interface is full. This DoS is critical because one of the major interest of CCN is to allow end-users to directly diffuse their content in a peer-to-peer way instead of relying on big Internet content providers. This attack could reduce the diversity of routable domains and consequently the interest of CCN.

Content Store attack DoS can also be launched against the Content Store in order to decrease the efficiency of the caching mechanism which

is one of the main components that provides the incentive to deploy CCN infrastructures. According to the caching policy, an attacker could generate a lot of download requests for unpopular contents which would modify the distribution of the downloaded contents and update the cache in a inefficient way. From a technical point of view, this attack is hard to achieve for a single attacker as it would need a lot of bandwidth to have a significant impact on the distribution of contents passing through the router.

3 Monitoring architecture

3.1 Requirements

The monitoring task consists of collecting information about the functioning and the current status of the CCN nodes. The objective is to correctly select and process the necessary information to highlight important facts. In this paper, we focus on the detection of anomalies resulting from attacks. As CCN works in a distributed manner with independent nodes, monitoring the network from a global perspective is thus hard. A solution could counter this problem by leveraging a central service interacting with a large set of CCN devices to collect and analyse information. However, this raises serious issues about reliability and scalability and does not fit to the CCN paradigm where each node has to be involved in the functioning of the network including the security related aspects.

Therefore, to stick to the CCN paradigm, our architecture is implemented at each CCN device which has to monitor itself for detecting anomalies. As presented in the previous section, a device has three main components: the FIB, the PIT and the Content Store. Each of them plays a crucial role in the well operation of CCN. For example, an abnormal Content Store can provide faulty contents or make the caching inefficient; a badly populated FIB may entail erroneous forwarding and so, some content may not be accessible any more similar to a bogus PIT which leads to disrupt the data content transmission over the back path of a request. Therefore, all of these three tables have to be monitored.

The objective of our monitoring architecture is to detect attack patterns by monitoring the recent past activity over the three tables. Since they may contain many information which are related to many actions (lookup, updates, etc.), monitoring and keeping track of all individual entry or action requires many resources that may delay the process or even affect the entire functioning of a node up to a denial of service in

the worst case. To guarantee the scalability and the timeliness, our architecture is designed to represent the three monitored tables by condensed metrics which values can be easily tracked along time.

Finally, devices can also share knowledge for detecting the attacks, in particular for highly distributed ones like DDoS, as well as for preventing future ones or recovering efficiently from anomalies. This may be provided as a content where devices can express their interest through the underlying CCN itself. However, as the individual monitoring is already required prior, our paper focuses on it.

3.2 Instrumentation

As explained in the previous section, there are different components of a CCN node that can be monitored for detecting malicious activities. All of them are impacted and/or impacts the network activity. For example, a node may receive an *Interest* (ingoing network activity) which has to be forwarded (outgoing network activity), this will update its PIT. Therefore, monitoring the network activity will track the global functioning of a node and its internal components without having a particular monitoring function for each of them.

For detecting attacks, network statistics are retrieved periodically from the CCNx implementation [2], every τ seconds. So, for each time window t, the following metrics are considered for all active faces of the CCN devices:

- $recv\_byt_t$: number of received bytes per second
- $sent\_byt_t$: number of sent bytes per second
- $recv\_data_t$: number of received *Data* packets per second
- $sent\_data_t$: number of sent *Data* packets per second
- $recv\_intr_t$: number of received *Interests* per second
- $sent\_intr_t$: number of sent *Interests* per second

We also consider more synthetic values on the router status that are also provided by the CCNx implementation:

- on Content statistics: number of $accessioned_t$, $stored_t$, $staled_t$, $sparse_t$, $duplicated_t$ and $sent_t$ contents
- on recent *Interest* statistics: number of $named_t$, $pending_t$, $propagating_t$ and $noted_t$ *Interests*
- on total *Interest* statistics: number of $accepted_t$, $dropped_t$, $sent_t$ and $stuffed_t$ *Interests*

3.3 Classification algorithm

The objective of the classification algorithm is to label each time window as anomalous or benign. A time window t_i is a tuple defined as:

$$< recv\_byt_{t_i}, sent\_byt_{t_i}, recv\_data_{t_i}, sent\_data_{t_i}, \\ recv\_intr_{t_i}, sent\_intr_{t_i}, accessioned_{t_i}, stored_{t_i}, \\ staled_{t_i}, sparse_{t_i}, duplicated_{t_i}, sent_{t_i}, named_{t_i}, \\ pending_{t_i}, propagating_{t_i}, noted_{t_i}, accepted_{t_i}, \\ dropped_{t_i}, sent_{t_i}, stuffed_{t_i} >$$

This paper leverages Support Vector Machines (SVM) [7] which are able to efficiently classify data, even if the data points are not separable linearly, while the complexity remains low [19] allowing our solution to detect in real time attacks affecting the monitored CCN nodes. For sake of clarity, we have considered a single attack at a certain time which is handled by 2-class SVM. However, multiple attacks could be detected by building a unique multi-class classifier [9].

2-class SVM is a supervised method and so requires M training samples: $Train = \{(t_1, l_1), \ldots, (t_M, l_M)\}$ with $l_i = 1$ if the time window t_i contains an attack, else -1. For enhancing the data separability, these samples are mapped into a higher dimensional space. Defining an efficient mapping function, ϕ, is a difficult task because it corresponds to add an additional dimensional space over data not given by any features. This however may be avoided by using the kernel function defined as:

$$K(t_i, t_j) = \langle\ \phi(t_i).\phi(t_j)\ \rangle \tag{1}$$

Because our data points are vectors representing the different metrics of a time window, the Radial Basis Function (RBF) is adapted:

$$K(t_i, t_j) = e^{-q||t_i - t_j||} \tag{2}$$

where q is tunable. Different values for q have been tested to choose an optimal which provides the best result.

Once in the space, the points may be linearly separable by a hyperplane which divides the samples of two classes with the maximum margins regarding the hyperplane. This leads to the following optimisation problem.

$$max \sum_{t_i \in Train} \alpha_{t_i} - \frac{1}{2} \sum_{\substack{t_i \in Train \\ t_j \in train}} \alpha_i \alpha_j l_i l_j K(t_i, t_j) \tag{3}$$

subject to where $C = 1.0$ determined through initial experiments:

$$\sum_{t_i \in Train} \alpha_i l_i = 0$$
$$\forall\ t_i \in Train, 0 \leq \alpha_i \leq C \tag{4}$$

As highlighted by these equations, the problem solving leads also to determine α_i which is used afterwards for classifying a new time window. A major advantage of SVM is that it relies on a subset of initial samples for the decision function, *i.e.* the support vectors which represent the training points such that $\alpha_i \neq 0$. Assuming, t_x, a time window which requires a prediction about the attack, it will be labelled by the following function:

$$f(t_x) = sgn(\sum_{(t_i,l_i) \in Train, \alpha_i \neq 0} \alpha_{t_i} l_i K(t_i, t_x) + b) \tag{5}$$

4 Experiments

4.1 Attack description and test environment

The most critical threat among the three described in Section 2.3 is clearly the attack on the PIT table. Attacks on Content Store can just reduce the efficiency of the cache and therefore do not present critical security issues. Also, attacks on the FIB will become critical when large deployments of CCN will occur, with many providers announcing contents, while PIT attacks is already a threat for local deployments involving few CCN nodes. Therefore the main threat we want to address when monitoring CCN nodes is the Pending Interest Table DoS attack as described in section 2.3. To realize the detection, we first implemented different attack strategies against the PIT in a single attack tool based on the source code of the *ccndsmoketest*[1] program provided in the CCNx implementation. The PIT stores *Interests* according to the faces they belong. To fill the PIT table we have to consider two dimensions, the number of faces created and the number of *Interests* requested.

- **Burst attack**: sending multiple *Interests* to multiple faces. Our first strategy sends a given number of *Interests* on a given number of faces. In extreme scenarios, we can send a lot of *Interests* to a single face or send a single *Interest* to several faces. Both dimensions can be combined leading to the following definition $Attack1(\#packets, \#faces)$ with the aforementioned remarkable values: $Attack1a(1, n)$, $Attack1b(n, 1)$, $Attack1c(n, 100)$, $Attack1d(100, n)$ where n defines the attack aggressiveness and may be tuned to study the impact of the attack.

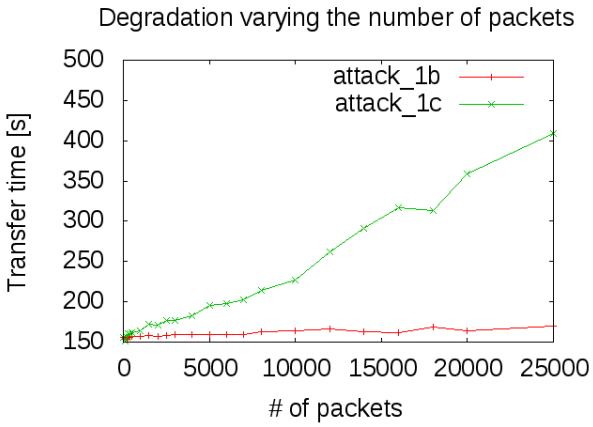


Fig. 2: Impact of varying # of packets (attack 1b, 1c)

– **Long duration attack**: keeping alive multiples faces with periodic *Interests*. Our second strategy consists into making the DoS more efficient by keeping alive a lot of faces with a small number of *Interests* that we send periodically. In this case, the attack aggressiveness, n, is the number of targeted faces. In this paper, keep alive *Interests* are sent every $t = 4$ seconds.

Our test-bed is composed of this attack tool and of two CCN devices running the routing daemon *ccnd* provided by CCNx. Both devices are on an restricted network used for this purpose. For each step in the experiment we transfer a 366MB video file from one device to the other.

We defined the impact factor of our attack as the time overhead introduced when transferring a content between our two CCN devices. Figure 2 shows the impact of the number of *Interest* packets we inject while targeting a constant number of faces. Firstly, we vary the *Interest* packet generation to inject them over one face (Attack 1b). The later we use the same principle but we inject them on 100 faces (Attack 1c). Figure 3 is similar but we vary the number of faces while maintaining a constant number of injected *Interest* packets (Attack 1a and 1d). Logically, performances are more degraded when the number of faces or *Interest* packets increase in particular if both are combined (attack 1c and 1d). Moreover, multiplying the number of faces used has a similar effect than sending multiple *Interest* packets. The second attack strategy, keeping alive many interfaces, can also significantly degrade the performance.

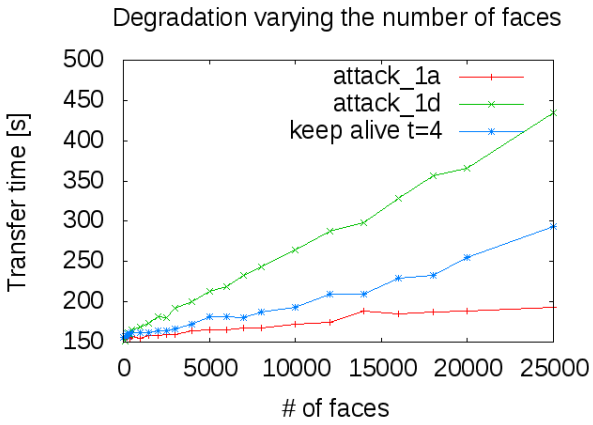


Fig. 3: Impact of varying # of faces (attack 1a, 1d, long duration)

4.2 Attack detection

As previously described attack detection is based on SVM analysing metrics over time windows. In our evaluation, the size of a window is set to one second. To assess the detection, the following metrics are used:

- the True Positive Rate (TPR): proportion of correctly identified windows presenting an attack,
- the False Positive Rate (FPR): proportion of windows without attack classified as attacks.

In order to strengthen our evaluation, only one third of the data is used for the training while the remaining is considered for testing and computing the previous metrics. Each experiment is run 10 times including a shuffle of windows for computing the average TPR and FPR. Initial experiments have been done to configure SVM for obtaining a good trade-off between TPR and FPR by using 5000 packets respectively faces as initial data.

In Figure 4 the true positive rate is plotted regarding the attack aggressiveness. This corresponds to the number of *Interest* packets for attacks 1b and 1c. The latter 1c is detected easier since the number of faces is multiplied meanwhile by 100 compared to 1b. Once the attack aggressiveness reaches 10,000 *Interest* packets, the TPR is higher than 95%. Similarly, the attack 1d is easier to monitor than 1a as the number of sent *Interest* packets is 100 times higher. Finally, the attack based on keep-alive *Interests* is well detected in any cases. In fact, such an attack last a longer time and is consequently much more visible.

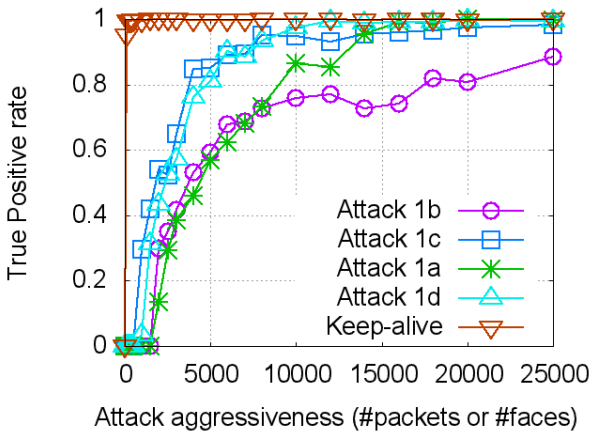


Fig. 4: True Positive Rate

Figure 5 shows that FPR remains low in most of cases. For the attacks 1b and 1c, generating a lot of *Interests*, they are never above 2%. The worst values are obtained for attacks involving a lot of faces (attack 1a, 1d and long duration attack) which seems contradictory as these attacks should be recognized easier when the number of solicited faces increases. In fact, this is due to two biases that we have investigated manually. First, the monitoring interface provided by CCNx gives metrics that are smoothed regarding the time. Hence, the impact of an attack is still visible on the monitoring interface in several time slots once it is finished (slow decrease of certain values over time) implying false positives. This all the more true with the keep alive which inject *Interest* packets periodically. This finding raises the need of a more accurate monitoring of inner values of CCN nodes for security purpose. Second, attacks involving many faces are longer to execute, which leads to have less windows without attacks. Thus, the training becomes less efficient for normal windows resulting in more false positives.

5 Related work

5.1 Alternative Content oriented approaches

There are several architecture, like CCN, which aim to shift away of today Internet point-to-point primitives, move to a more data-oriented and content-centric paradigm, replace the end-to-end communication network model by publish/subscribe model of a distribution network and to used

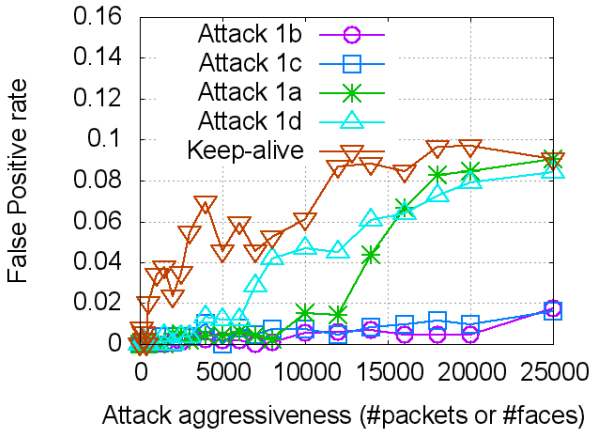


Fig. 5: False Positive Rate

cached copy of content for faster retrieval. Main differences between the different research approaches are the content naming scheme and how inter-domain routing is handled.

TRIAD [6] was the first to propose such an architecture. Names in TRIAD are based on URLs and use DNS for their resolution. Furthermore directories are used to map content to a replica server close-by. Shortly afterwards Brent Baccala in [4] expressed a similar idea of moving a more content-centric approach.

In 2007, *Koponen et al.* renewed the idea of a content-oriented network at Berkeley. DONA[15] was the name of this project. They followed another idea which consists into replacing DNS with flat and self-certifying names avoiding PKI for key verification.

The PSIRP[17] project introduces an architecture based on rendezvous points. Content is publish at the source. Each pieces has two labels, a public label used for the subscription to the content a private label used to verify the publisher.

Another research project focusing on content-centric networking is the 4WARD NetInf[8] project. Content is published using information units called InformationObject (IO). As in general every IO needs a unique identifier by which it can be referenced, a multi-level DHT (Distributed Hash Table) handles the name resolution and location lookup for a given IO.

5.2 Research efforts on CCN

The most popular architecture for research purposes is the Content Centric Networking proposed by Van Jacobson et al [12] from early 2007 and later introduced to the research community [14] in 2009. CCN current development is quite advanced thanks to the CCNx open source framework [2]. PARC pursues research efforts of their architecture, describing and implementing advanced features and functionalities as the capacity of CCN to transport voice [13] with the adapted architecture. Many issues are described in [20] and still need to be addressed to make CCN (or Named Data Networking) a viable solution, for example: the scalability of routing on names, the efficiency of key management, the management of contents or the security of CCN nodes are critical questions deserving research efforts. Also, the design of a complete model to better understand the working and the benefit of a CCN architecture according to the network configuration, as proposed by Carofiglio et al [5], is an important step forward. Privacy on the Internet is more than ever a critical topic. DiBenedetto et al proposed in [10] a application over CCN that enables privacy preserving communications while introducing less relative overhead than TOR running over IP.

Among researches on CCN, a only few security issues have been investigated. In his master thesis, Tobias Lauinger [16] identified several attacks related to caches, in particular denial-of-service attacks against CCN routers, but he only investigated another attack "cache snooping" that enables attackers to efficiently monitor which content their neighbours are retrieving.

6 Conlusion

We presented in this paper a first monitoring architecture for CCN. While this new paradigm worth being investigated for the sake of future Internet, it also raises new management challenges we presented in this paper. Among those, we investigated one of the most important problem affecting CCN devices: the possible denial of service through the flooding of the PIT table. To address this issue, we used the monitoring features of the reference implementation coupled with a classification algorithm based on SVM and which can efficiently detect such attacks with a small computational cost. In fact, we implemented and experimented different attack strategies to perform DoS and all of them can be detected with very low error rates, which could be even lower with a more accurate report of operating values.

Our research opens directions for a lot of future works. First of all, our detection mechanism will be implemented within the CCNx libraries in order to enable real-time detection and the usage of associated countermeasure to mitigate attacks. We will then extend our monitoring architecture to monitor the other tables (FIB and Content Store tables) to detect other types of attacks. Finally, we want to extend our test-bed and generate more realistic traces including traffic from different applications. Attack detection was the focus point of our approach and future work will also focus on attack prevention.

References

1. CCNDSMOKETEST Manual Page, http://www.ccnx.org/releases/latest/doc/manpages/ccndsmoketest.1.html
2. Content Centric Networking, http://www.ccnx.org
3. Named Data Networking, http://named-data.net
4. Baccala, B.: Data-oriented networking. INTERNET-DRAFT (August 2002)
5. Carofiglio, G., Gallo, M., Muscariello, L., Perino, D.: Modeling data transfer in content-centric networking. In: Proceedings of the 23rd International Teletraffic Congress. pp. 111–118. ITC '11, ITCP (2011), http://dl.acm.org/citation.cfm?id=2043468.2043487
6. Cheriton, D.R., Gritter, M.: Triad: A new next-generation internet architecture (July 2000)
7. Cristianini, N., Shawe-Taylor, J.: An introduction to support Vector Machines: and other kernel-based learning methods. Cambridge University Press, New York, USA (2000)
8. Dannewitz, C., Herlich, M., Bauer, E., Becker, M., Beister, F., Dertmann, N., Hrestic, R., Kionka, M., Mohr, M., Mühe, M., Murali, D., Steffen, F., Stey, S., Unruh, E., Wang, Q., Weber, S.: Opennetinf documentation design and implementation (September 2011)
9. Debnath, R., Takahide, N., Takahashi, H.: A decision based one-against-one method for multi-class support vector machine. Pattern Anal. Appl. 7(2), 164–175 (2004)
10. DiBenedetto, S., Gasti, P., Tsudik, G., Uzun, E.: Andana: Anonymous named data networking application. CoRR abs/1112.2205 (2011), http://dblp.uni-trier.de/db/journals/corr/corr1112.html#abs-1112-2205
11. Ghodsi, A., Shenker, S., Koponen, T., Singla, A., Raghavan, B., Wilcox, J.: Information-centric networking: seeing the forest for the trees. In: Proceedings of the 10th ACM Workshop on Hot Topics in Networks. pp. 1:1–1:6. HotNets '11, ACM, New York, NY, USA (2011)
12. Jacobson, V., Mosko, M., Smetters, D., Garcia-Luna-Aceves, J.J.: Content-centric networking: Whitepaper describing future assurable global networks. Response to DARPA RFI SN07-12 (2007)
13. Jacobson, V., Smetters, D.K., Briggs, N.H., Plass, M.F., Stewart, P., Thornton, J.D., Braynard, R.L.: VoCCN: voice-over content-centric networks. In: Proceedings of the 2009 workshop on Re-architecting the internet. pp. 1–6. ReArch '09, ACM, New York, NY, USA (2009)

14. Jacobson, V., Smetters, D.K., Thornton, J.D., Plass, M.F., Briggs, N.H., Braynard, R.L.: Networking named content. In: Proceedings of the 5th international conference on Emerging networking experiments and technologies. pp. 1–12. CoNEXT '09, ACM, New York, NY, USA (2009)
15. Koponen, T., Chawla, M., Chun, B.G., Ermolinskiy, A., Kim, K.H., Shenker, S., Stoica, I.: A data-oriented (and beyond) network architecture. In: Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications. pp. 181–192. SIGCOMM '07, ACM, New York, NY, USA (2007)
16. Lauinger, T.: Security & scalability of content-centric networking (September 2010), `http://tubiblio.ulb.tu-darmstadt.de/46912/`
17. Särelä, M., Rinta-aho, T., Tarkoma, S.: RTFM: Publish/Subscribe Internetworking Architecture. ICT-MobileSummit Conference (2008)
18. Schulze, H., Mochalski, K.: Internet study 2008/2009 (2009)
19. Wang, L. (ed.): Support Vector Machines: Theory and Applications, Studies in Fuzziness and Soft Computing, vol. 177. Springer (2005)
20. Zhang, L., Estrin, D., Burke, J., Jacobson, V., Thornton, J.D., Smetters, D.K., Zhang, B., Tsudik, G., kc claffy, Krioukov, D., Massey, D., Papadopoulos, C., Abdelzaher, T., Wang, L., Crowley, P., Yeh, E.: Named Data Networking (NDN) Project (October 2010)