Demonstration of a Functional Chaining System Enabled by Named-Data Networking

Lei Liu, Liguang (Ted) Xie, Mehdi Bahrami, Yang Peng, Akira Ito
Fujitsu Laboratories of America, Inc.
1240 E. Arques Ave., M/S 345
Sunnyvale, California 94085, USA
lliu, lxie, mbahrami, ypeng, aito@us.fujitsu.com

Sevak Mnatsakanyan, Gang Qu, Zilong Ye, Huiping Guo
California State University, Los Angeles
5151 State University Drive
Los Angeles, California 90032, USA
smnatsa2, gqu, zye5, hphuo@calstatela.edu

ABSTRACT
In this work, we present a functional chaining system enabled by Named-Data Networking (NDN). By using the proposed naming semantics and on-the-fly processing procedures, a functional chaining request, which consists of the name of raw data and an ordered set of functions, can be executed dynamically and seamlessly. Moreover, we demonstrate a functional chaining forwarding strategy and a whole chain security scheme, which can improve the system performance in terms of forwarding efficiency and security, respectively.

CCS Concepts
• Security and privacy ~ Network security • Networks ~ Network protocols • Networks ~ Network control algorithms • Networks ~ Network experimentation • Networks ~ In-network processing

Keywords
Named-Data Networking (NDN); functional chaining; forwarding strategy; security

1. INTRODUCTION
Functional chaining is an emerging use case where the consumer requests composite data / services which consist of an ordered set of functions that are applied to raw data. In this work, we demonstrate new solutions to seamlessly enable functional chaining over a Named-Data Network (NDN) [1]. Note that, conceptually, functional chaining in this work mainly refers to the chaining of application layer functions (e.g. video processing) to generate new data, which is different from the Service Function Chaining (SFC) [2] being standardized by IETF where the steering of traffic flow with network related functions (e.g. firewalls, load balancing) is the major focus.

A previous work with the similar motivation has been presented in the Named Function Networking (NFN) project [3]. However, in this work, we present different solutions compared with NFN, including but not limited to, naming semantics, Interest/Data processing procedures, forwarding strategy and security, as will be detailed in Section 2.

ICN'16, September 26-28, 2016, Kyoto, Japan
ACM ISBN 978-1-4503-4467-8/16/09…$15.00
http://dx.doi.org/10.1145/2984356.2985239

2. FUNCTIONAL CHAINING SYSTEM
2.1 Naming
We use the arrow "←" to connect and separate different functions in a functional chain request. For example, an Interest with name */C ← /B ← /A* requests raw data *A* to be processed by a function *B* firstly and then processed by a function *C*. Note that, this naming semantics can be flexibly extended to support more complicated use cases, such as functional chaining with multiple raw data (e.g., *A1*, *A2* and *A3*) can be named as */C ← /B ← /A1+/A2+/A3*.

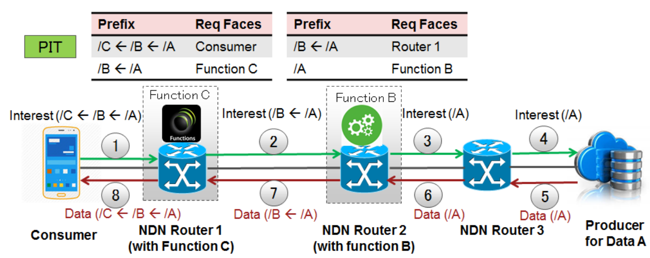

Figure 1. Interest & Data processing procedure

2.2 Interest & Data Processing Procedure
The Interest and data processing procedure is shown in Fig.1. A consumer sends a functional chaining Interest */C ← /B ← /A* to a NDN network. Given the longest prefix matching, this Interest is firstly forwarded to a NDN router which can provide function *C* (function *C* can be either resident in the router or connected to the router via an external link), i.e., NDN Router 1 in Fig.1 (step 1). NDN Router 1 with function *C* processes this Interest by changing its name to */B ← /A*, and forwards the new name Interest to the next router (step 2). Note that here, two Pending Interest Table (PIT) entries are added to the NDN Router 1. The first PIT entry is created for the original Interest */C ← /B ← /A*, facing the consumer, which is used to bring the final Data */C ← /B ← /A*, once available, back to the consumer. The second PIT entry is created for the Interest with the new name */B ← /A*, facing function *C*, which is used to forward the Data */B ← /A* to function *C* for further processing to generate the final Data */C ← /B ← /A*. After step 2, similarly, the Interest */B ← /A* is forwarded to NDN Router 2 with function *B*, which changes the Interest name to */A*, generates two PIT entries in Router 2, and forwards the Interest */A* to the producer of Data *A* (step 3 and step 4).

Once Data */A* are sent back to the Router 2 (step 5 and step 6), according to the two PIT entries at Router 2, Data */A* are firstly sent to function *B* to generate Data */B* ← */A*, and then Data */B* ← */A* are sent back to the previous router (step 7). With a similar procedure, according to the two PIT entries at Router 1, Data */B* ← */A* is processed by function C firstly and the final Data */C* ← */B* ← */A* are sent back to the consumer dynamically (step 8). Similar to standard NDN operations, the processed Data such as */B* ← */A* and */C* ← */B* ← */A* can be cached in corresponding routers for future functional chaining requests.

The above procedure can seamlessly support functional chaining, which is enabled by only using naming semantics along with two straightforward operations: 1) name changes on Interests, and 2) creation of two PIT entries at the routers which can provide the requested functions. On the other hand, the NFN uses a 3-phase procedure to support a functional chaining request, i.e., 1) upstream fetch, 2) separate code and data fetch, and 3) computation push, as presented in [3]. In particular, the Interest is forwarded to a NFN capable router firstly (in the phase 2) which divides the functional chaining request into multiple individual Interests to fetch the functions and data separately. If the phase 2 fails, the data are pushed to the functional node in the phase 3. In this case, the functional chaining procedure presented in this work is simpler than NFN to support functional chaining.

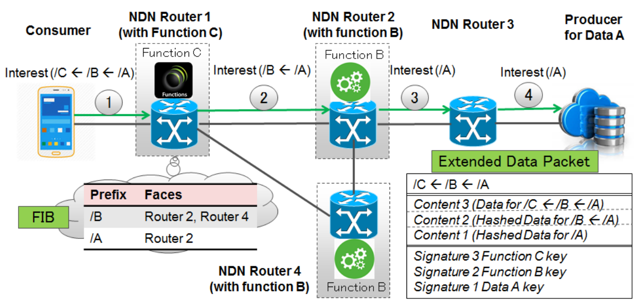

Figure 2. Forwarding strategy & extended Data packet

2.3 Functional Chaining Forwarding Strategy

To improve the forwarding performance, we implement and demonstrate a new forwarding strategy, referred to as functional chaining (FC) forwarding strategy. As the Interest contains the name of all the requested functions and data, each router can take the whole chain information into account to make the forwarding decision. For example, as shown in Fig.2, once the Interest */C* ← */B* ← */A* is received at Router 1, Router 1 not only knows that the function C can be provided locally, but also knows that *B* and *A* will be required later. Then Router 1 can check its Forwarding Information Base (FIB) table and, in this example, find that both */B* and */A* can be reached via the face to Router 2. Therefore, although Router 4 can also provide function B with a same cost (i.e. hops), Router 1 will forward the Interest to Router 2 rather than Router 4 as it is potentially a better path to reach both */B* and */A* later. In short, the FC forwarding strategy is, if there is an identical face for the required functions/data, the Interest is forwarded to that particular face; otherwise, the default NDN forwarding strategy is applied. Compared to IP networks where only the destination IP address is used for forwarding, NDN can support a smart forwarding by considering multiple names for a functional chaining request in nature.

2.4 Whole Chain Security

We also present a whole chain security solution, which allows the consumer to verify the data source and all intermediate functional nodes along the whole chain process. As shown in Fig.2, an extended Data packet of NDN is introduced, which includes not only the signature of the final function *C* and the final content */C* ← */B* ← */A* in the chain process, but also the previous signatures from intermediate functions *B* and *A*, as well as the hashed data (we used fixed length hash to reduce data packet overhead) for intermediate content */B* ← */A* and */A*. Therefore, by using this extended Data packet, the consumer can verify the whole chain security to detect any malicious producers/functions during the entire processing procedure.

3. DEMO SCENARIO & SETUP

To showcase the proposed solutions, we demonstrate a use case which is a video processing chain. Our demonstration is set up as shown in Fig.3, which includes consumers, video producers, and NDN routers with video combination and video compression functions. We assume the audiences in the stadium capture video clips for a live sport game (e.g. a goal in a soccer game) from different angles, and a consumer requests the multiple-angle video clips to be combined together into one video and then compressed into a smaller file. All the solutions presented in Section 2, including the naming semantics, processing procedures, FC forwarding strategy and whole chain security are implemented and demonstrated to validate the overall feasibility and efficiency of the proposed solutions to support this functional chaining use case. In this demonstration, FIB tables are manually configured and a routing protocol will be introduced to automatically create / update FIB tables in our future works.

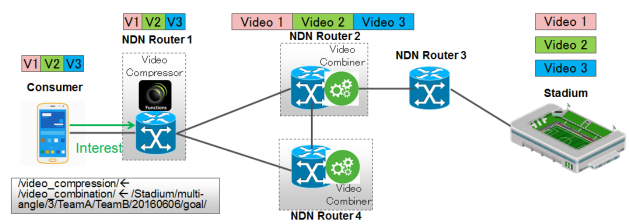

Figure 3. Demo setup and use case

4. CONCLUSIONS

In this work, a functional chaining system enabled by NDN is presented. By using the proposed naming semantics and on-the-fly processing procedures, functional chaining requests can be processed dynamically and seamlessly. An efficient chaining forwarding strategy and a whole chain security scheme are also demonstrated in a functional chaining use case.

5. REFERENCES

[1] Zhang, L., Afanasyev, A., Burke, J., Jacobson, V., Claffy, K., Crowley, P., Papadopoulos, C., Wang, L., and Zhang, B. 2014. Named data networking. *SIGCOMM Comput. Commun. Rev.* 44, 3 (Jul. 2014), 66-73. DOI= http://dx.doi.org/10.1145/2656877.2656887.

[2] Halpern, J., and Pignataro, C. Eds. 2015. Service function chaining (SFC) architecture. IETF RFC 7665 (Oct. 2015).

[3] Sifalakis, M., Kohler, B., Scherb, C., and Tschudin, C. 2014. An information centric network for computing the distribution of computations. In *Proceedings of the 1st ACM Conference on Information-Centric Networking* (Paris, France, Sep. 24-26, 2014). ICN'14. ACM, New York, NY, 137-146. DOI= http://dx.doi.org/10.1145/2660129.2660150.