Named Service Networking

Shuo Chen, Junwei Cao*
Research Institute of Information Technology
Tsinghua National Laboratory
for Information Science and Technology
Tsinghua University, Beijing 100084, China
Email: *jcao@tsinghua.edu.cn

Lipeng Zhu
Smart Grid Institute of State Grid Corporation of China
Nanjing 210003, China
Email: zhulipeng@sgri.sgcc.com.cn

Abstract—**Information Centric Networking (ICN) proposes a new paradigm converting thin waist of networking from IP packet to named chunk. In the practical implementation of multiple ICN projects, applications based on ICN may not just focus on data distribution, but also service and functional components. In this paper, an abstraction of service extensions over ICN is proposed, called Named Service Networking (NSN). It demonstrates necessary components to build functional services over ICN, including security, semantics, etc. A prototype of NSN is implemented over Named Data Networking (NDN) and evaluated to test its transmission efficiency, scalability and availability.**

Keywords*—*Information Centric Networking; Named Data Networking; Named Service Networking;

I. Introduction

The Internet architecture originates from conversation between two machines. However, the main stream of today's Internet is converted from end-to-end conversation to large mount and scale of data distribution, which motivates the proposal of kinds of named data object centric networks. So-called Information Centric Networking (ICN) is named for this approach of networking [1]. The most important conversion in ICN is the network thin waist shift from IP packets to Named Data Object. The transport mechanism is not simply push, but pull or publish/subscribe. The name scope in network layer could not only describe data, but also anything meaningful upon network.

Several approaches inspired by ICN propose service centric networking (SCN), which imports service identifier into or upon network packets [2]. No matter what kind of underlying transport network, all the identity in network could be abstracted into services. This SCN approach intends to promote service exploration, service-level routing, service delivery, security issues and so on. Web Service is a service calling and integration approach over network, usually http over TCP/IP. It is an application layer specification that does not hack or impact underlying network. There is re-expression processes converting network packets to service application data. In [3], current network architecture is reviewed. The data presenting process is the major cost in network transmission. Application level framing is proposed that the format of network packet is directly defined and used by application. Most ICN projects provide meaningful naming of the network packet. Application level framing could be realized by upper level upon ICN network.

In this paper, we proposes an extension of deploying services over ICN. Compared with the current SCN design [2], it promotes more meaningful semantics for service hosts to understand each other and adopts ICN advantages in cache, network security and so on. A prototype of this service layer over Named Data Networking (NDN) [4], which is one of the project of ICN, is implemented and also evaluated on efficiency, scalability and availability.

The rest of paper is organized as follows: Section 2 shows the background of service extension design. Section 3 demonstrates the abstract design of service extension over ICN. Section 4 demonstrates our implementation over NDN. Section 5 evaluates NSN over NDN. Section 6 concludes the paper and addresses open issue of future work.

II. Background

A. *Information Centric Network*

The major shift in ICN is Named Data Object as thin waist of network stack. This design leverages in network caching and decoupling of data requesters and publishers [1]. DONA [5] proposes IP overlay naming resolution with overlay name routing. Name is registered to name resolution infrastructure so that requests of such names could be routed across resolution handlers. In PSIRP [6], names are published to network. Data requesters subscribe the name of the data. The rendezvous system manages to match publications and subscriptions.

Named Data Networking (NDN) [4] adopts a clean slate design. Each routing node consists of Forwarding Interests Table (FIB) like forwarding table in TCP/IP, with prefixes of names instead of prefixes of IP addresses. Content Store (CS) serves as in-network cache of named data packets. Pending Interest Table (PIT) stores data requests that have been issued but not satisfied yet to prevent DDOS and to reduce the stream of requests. NDN packets could be classified into interests and data packets. Security is a built-in system in NDN, since data packets carry signature for validation and encryption. In current NDN implementation, Type-length-value (TLV) structure is used as packet format of interest and data packet. This design provides flexibility to define more complicated semantics for NDN.

Different projects of ICN share some common conceptual design [1].

- Named Data Chunk as unique supported scheme regardless of data location, replication counts and so on.
- Verifiable binding between object and its name.

- In-network caching.
- Name based routing and forwarding.

B. Service Centric Network

Inspired by ICN, concept of services is brought upon named data object to a service-level view of networking. Service-Centric Networking in [2] proposes an extension of Content Centric Networking (CCN). It changes semantics of CCN. Interest is not just request of data, but request of functional service, while data packet also contains result of function. Host which receives the functional interest will possess according to the command and parameter encoded in interest. A more specific approach based on [2] is proposed in [7].

C. Web Service

Web Service is a method for machines to understand each other across the network. Program can access remote functional component via network conforming to specification defined in web service. UDDI, WSDL, and SOAP are three major specifications popular in practice. Besides these basic components, schemes of service discovery, security, job scheduling are also discussed in many works. Most implementations of web services are based on HTTP and TCP/IP.

III. Named Service Networking Design

In SCN [7], an initiative design of service is proposed including name resolution, service parameter and type support, service routing and service delivery. However, this is not enough to construct functional services. Web service is an overlay services integration network. Bussler demonstrates that service discovery, selection, mediation and composition are key elements to achieve scalable web services. [8]

Security is the essential issue to construct available services. Request in SCN [2] is not just to request data but also to change the status of service provider. For example, if a storage service offers remote operation API to insert and delete data. Malicious request cannot be identified or prevented without verification of requests.

Semantic negotiation or service description is missing in SCN. In [7], although commands and parameters can be encoded in requests for certain service, there is no specification to describe such service that client and service are tightly coupled in development. In web service, WSDL is used for web application to understand how to call this service.

Service discovery is also necessary in large scale network. In traditional web service, service provider registers its service information to centralized architecture like UDDI.

Service-centric networking [2] extension proposes a basic service abstraction over CCN. Service-centric networking extensions [7] extends service abstraction upon different kinds of ICNs. The basic framework can be summarized as follows:

a. Naming: $\langle content\_owner, content\_name \rangle$ to integrate hierarchical and flat naming.
b. Naming resolution: mapping between service names between service locators.
c. Service parameter: tweaking the encoding of content request in coupled approach such as CCN; Enable rich service description in matching system in decoupled approach such as PURSUIT.
d. Service deployment: network environment parameters should be provided to help making decision where to deploy service entity.

However, just points from a to d based on ICN are not enough to support scalable services. Web service is a successful approach to integrate heterogeneous services over network. A conceptual framework of web service, WSMF is proposed in [9]. It points out that document types, semantics, transport binding, exchange sequence definition, process definition, security, syntax, trading partner specific configuration are necessary elements to achieve scalable web, service discovery, selection, mediation and composition. [10]

In this paper, we propose Named Service Network (NSN) to redefine service abstraction over ICN. Here are our major principles of designing NSN.

a. DecouplingNSN over ICN. Layered design with NSN layer and ICN network layer enables evolvement and innovation in service and network level separately.
b. Security mechanism to guarantee authentication of service request.
c. Extensive service description and semantics
d. Service discovery and resolution for scalable management of service
e. Meditation among similar or different semantics of service
f. Customized adaption of different ICN, network environment and different underlying network policy.

The conceptual design of NSN over ICN is shown as Figure 1. The data network layer is ICN network in spite of NDN, DONA or PURSUIT. A decoupled service layer is built upon data network layer, which means that all the requests of services or returned results conform to the underlying network protocol. Customized adaption is necessary when underlying is optimized of specific objectives above.

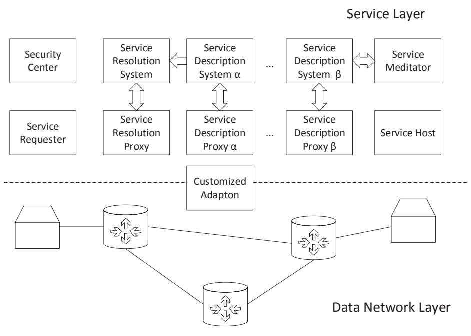

Fig. 1: NSN Architecture

A. Service Abstraction

In ICN, network is to return named data to satisfy the request of a given name. In named service networking, network is to transport the request service of a given name and response of possessing results. NSN uses ICN as underlying transport network, which means named service requests are encoded as named data requests of ICN and service responses as named data responses. NSN does not change but conforms to underlying ICN layer, including data packet specification, transport, routing and so on. NSN could be viewed as application upon ICN.

In NSN, service is the entity that actions according to service requests. Each service has its unique name to represent itself. Service has its specification for requests and responses. This semantics of specification is organized in data packet contating format information. This format data can be interpreted by service requesters to construct service requests and service hosts to construct results.

B. Security

Fundamentally, each request and result should not be modifiable and could be encrypted for privacy. Besides, identities related to requests or results could be publicly validated. This identity could also be used for access control to define access control list for authorization of each identity.

In NDN, content-based security is adopted. All the data packets are authenticated with digital signatures [11]. Request of data cannot be validated in original design of ICNs. In design of SCN, semantics of data request is accommodated to request of service. In ICN design, extensible encoding is adopted for requests and data packets, for example, ccnb/XML encoding of CCN or TLV encoding of NDN. Verification information of requests could be encoded as a subsection of data request. Otherwise, verification identifier could be encoded sequentially following the data requests.

The basic purpose of security center is to validate request. Asymmetry key is the most common way in today's security industry. The request would be signed by the private key of the requester. The key pair can be signed hierarchically from the root trust anchor or in a P2P way by identical peers. Although keys could be stored locally by any service node in the network, it is harmful to build a scalable verification system. No matter in which trust model, there should be storage of important keys, verification host for high level keys or relays of the verification chain. Security center is not just a single host in NSN, but could be designed in a hierarchical or P2P way according to the trust model and trust policy design.

C. Service Discovery

In SCN extension design, service name resolution is mechanism to organize and to retrieve service name [7]. This is the primitive way for service request of searching concerning services. However, this is also difficult for a service requester to know what the exact name of service without prior knowledge of service.

In common web service design, UDDI is used as service discovery system. In NSN design, service resolution proxy module is delegation for service requester of service discovery. Service resolution system is actually a service provider and its semantics is public as prior settings of service requester.

NSN adopts keyword searching approach. Keywords are encoded as searching parameter in the request of service. Keyword text query like google is highly developed. Lots of open-source projects such as Apache Lucene can provide efficient text indexing and searching. They are easy to be deployed and learning cost is rather low. The process of service discovery is analogy to web page like google. The list of results is encoded in a data packet to satisfy the service discovery request.

D. Semantics

The description of a service consists of 6 elements:

a. Name: the name of service
b. Description and keywords: text to describe service. This section would be registered to service discovery system.
c. Function List: name list of functions that service could provide
d. Pre-conditions: describe what the service needs to enable the service function. Commonly, it contains schema of input parameter. Optionally for stateful service, it also contains description of state to trigger the success and lead to failure.
e. Post-conditions: describe what the service would response under different conditions. It is organized in different cases and schema of results in each cases. Optionally, it will also describe following state of service host.
f. Owner: optional. Identifier of service owner, commonly certificate of the service owner.

Here is an example of a repository service. Repository is a service to store named data packets. Retrieval of data packets conforms to underlying ICN specification, while insertion and removal of data packets are the actual service of handling data packets. The example above is encoded in pseudo code which explains the code itself.

```
Repo Service Description

Name: /THU/repo
Description: insertion and deletion of
             data packets
Keywords: repository, repo, insertion
          deletion, data packet
Function List:
    Name: insert
        Description: put data packets
        into repo
    Name: delete
        Description: remove data packets
        from repo
PreCondition:
    FunctionName: insert
        Parameter: DataName
            DataName: BYTE
    FunctionName: delete
        Parameter: DataName
            DataName: BYTE
PostCondition:
    FunctionName: insert
        Result: StatusCode
        StatusCode: INTEGER
    FunctionName: delete
        Result: StatusCode
```

The major components of WSDL are definitions, types, portType, operation, binding and service. In NSN description document, binding is the underlying ICN network. Operation is the names in the function list. The type of parameter in NSN refers to types in WSDL. Definitions is compared to the service prefix in NSN. We can say that WSDL can be transferred to NSN description document conceptually.

Service description proxy works as delegation of service requester, while service description system works as a regular service provider. When the requester has the exact name of service, it will send data requests for service description. The description will be encoded in the response data packet. The service description system is actually holder of service description data packets. The name scopes are public and pre-configured in the proxy. The practical implementation of system is not stipulated and the semantics of description can be different. For different semantics, there should be different proxy to interpret them.

E. Meditation

There are two types of integration in web service: orchestration and choreography. For complicated services which can be decoupled into sub-services, there would be composition of different services. Mediator is to translate complicated service request to sub-service requests. The integration function of mediator can be compared to BPEL engine in web service. The major principles of mediator are as follows:

a. Manage registration of simple services
b. Integrate processes of different services into one complicate service. The services would be different in semantics. Mediator should understand and provide service requester a uniform semantics.
c. Handle complicated service requests and forward simple service requests. Mediator will control the whole process and reacting according to the state, but mediator will not do the practical task.
d. If a service requester sends a request in an inappropriate semantics, it could send the requests to mediator. Mediator could interpret the request into right semantics and forward the request.

F. Customized Adaption

Although NSN adopts decoupled service layer over underlying ICN network, NSN also needs customized adaption to exploit efficiency or adjust for certain purpose. One example is that NDN has strategy layer for forwarding decisions. Specific strategy could be configured to improve the certain service level.

G. Summary

If a service requester wants to request a certain service, it could send a query with keywords for the exact service name and the service description document. Then it interprets the service description and sends request to the certain service. If there is a conflict of semantics, the service would ask mediator for adjusted service description and reinterpret the service request again.

IV. Implementation of NSN over NDN

Prototype of NSN is implemented over NDN. The service level is detached from underlying NDN network, which means all the requests and responses of service conform to NDN protocol. The service abstraction could be viewed as application of NDN. Semantics of NDN interest is adjusted to service request and the result of service request is encoded into content of NDN data packet.

A. Naming

Naming adopts hierarchical naming like URL. In current NDN implementation, all the NDN are encoded in TLV format and structured name is just a sub TLV block of network packets. The design of naming is referred to command interest of repo-ng[1]. The naming semantics of service request is that:

Parameter is just a name-component which can consists TLV sub-blocks.

```
/service name/function name/parameter
```

B. Security

Signed interest[2] proposed by NDN project is a mechanism to issue an authenticated interest. It contains signature of the name, timestamp and random number as the components of the interest name. Signature is used to validate the service request. Timestamp is for detecting the service sequence and random number is to guarantee the uniqueness of the service request. So the complete name of service request is that:

```
/service name/function name/parameter
/signature/timestamp/random-value
```

Security center actually keeps public key as data packets. It is implemented as *repo-ng*Security center will response data packet of certificate with given name. NDN data packet originally contains signature for authentication.

C. Semantics

Semantics of NSN over NDN follows design in previous section. In PreConditoin and PostCondition part, parameters and results are encoded in TLV schema. For each value type in schema, there will be a type number to represent it. If service schema is developed independently, it is hard to avoid type number conflicts. Thus, services will be classified to different semantics. In NDN based implementation, *repo-ng* is also used as service description system. Semantics information is encoded in NDN data packets. We do not distinguish different semantics, but put them in a single repo-ng. PublicKeyLocator is adopted in selector of interest to address service description data packet of certain publisher.

D. Service discovery

The major principle of service discovery is to return service names with given query keywords. Service discovery system of NSN-NDN is implemented based on Lucene. The description and keywords sections of service description are extracted into small text files and tagged with the names of services. Service resolution request contains the keywords. Service resolution system extracts the keywords and queries Lucene system. Lucene will return a list of results. These results will encoded in the data packet satisfying the request.

[1] Repo-ng: repo of new generation
http://redmine.named-data.net/projects/repo-ng/wiki

[2] Signed interest:
http://redmine.named-data.net/projects/ndn-cxx/wiki/SignedInterest

E. Mediator

If format of service request does not satisfy syntax of service, service host will response the result representing malformed format. The service requester may know this would result from a semantics conflict and version of service description is out of date. Service requester could consult mediator for the right version of service description.

In NSN-NDN, mediator is also implemented by repo-ng. Whenever service host changes the semantics, it will put new service description data packet into mediator with new version tag. Mediator will republish service description signed by mediator. Currently, NSN-NDN could just handle stateless sequential process, which means simple process will execute one after another no matter what the previous status is. More complicated workflow will be discussed in future work.

F. Transport

Service requests of NSN-NDN are transported as interests of NDN and service responses are encoded in data packets satisfying the interests. The transportation of NSN-NDN conforms to NDN protocol. NDN forwarding daemon (NFD)[3] is used to forward service packets and to construct the routing information.

Different service hosts can provide services of the same name. Service requests can be forwarded to a certain service host according to configured strategy. Forwarding strategy is a framework for making decisions on which service request is forwarded to which face. Customized adaption is made upon NDN strategy layer to better support NSN. In current implementation, a simple load balance strategy is adopted. Service request is forwarded to one of the best 3 interfaces each time.

V. Evaluation

The NSN-NDN prototype is evaluated on Amazon Web Service EC2 platform. The configuration of virtual machine is m3.large of 2 vCPU, 6.5 ECU and 32GB of SSD storage. In [12], the impact of Amazon Ec2 on network performance is analysed that virtual machine of medium configuration above has very little impact on network performance.

The operation system is ubuntu 14.04. NDN Forwarding daemon (NFD) is used as NDN protocol forwarding software and UDP as underlying transportation protocol. Repo-ng is revised as NSN storage service host. N.Virginia and Oregon zones are chosen as experiment data center.

A. Transmission Efficiency

Data distribution consists of majority of network flow in current Internet. Data transmission efficiency is an important feature of the service network. The data throughput is evaluated between NSN Repo service and repo of web service. The web service repo is developed based on gsoap[4] library. Both services codebase share the same storage backend sqlite. Figure 2 shows the throughput of putting file. The data transfer is between two hosts of different subnet in Amazon N.Virginia data center. The data fetching requests are sent in pipeline of 20. NSN has better efficiency in thoughput over soap. The throughput also stays stable as the size of file grows. Compared with gsoap over HTTP, NSN over NDN shows better transmission efficiency. When gsoap starts an service process, 3 times of handshake will lead to latency of 1.5 RTT which does not exist in NDN.

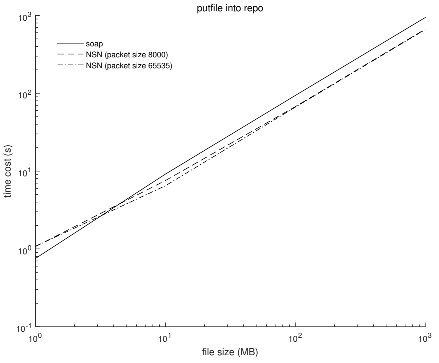

Fig. 2: Throughput of putting file

NSN-NDN also leverages repeated request transmission efficiency by content-store structure in NDN network. Figure 3 shows the total service handled time when client sends the repeated same service requests between Oregon and N.Virginia Amazon datacenter. After the NSN client fetches the first service response packet, it will check local or nearby cache first to satisfy the service request. In the meantime, web service client will keep fetching response from service host and service host will execute the service process repeatedly.

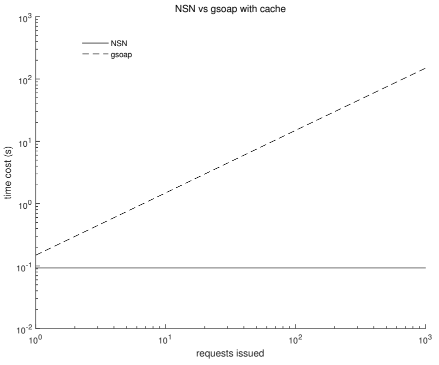

Fig. 3: Repeated Request Transmission Efficiency

B. Service Migration

Decoupling of name and location of service is the major feature of NSN-NDN. The most important reason is that security is built upon content, but not the session or connection between two hosts. Virtual machine or process migration is the common case in cloud computing. The impact of service migration between two hosts is evaluated. Two hosts are connected to the same NFD in the different subnet of the same Amazon datacenter. The client keeps sending service request to one host. At 10s, the service on one host is shut down and

[3] NDN Forwarding Daemon (NFD) is a new NDN forwarder under active development. http://redmine.named-data.net/projects/nfd/wiki

[4] gsoap Toolkit for SOAP: http://www.cs.fsu.edu/ engelen/soap.html

started on the other host. Figure 4 shows the service request responded rate. The service returns to normal in 0.2.

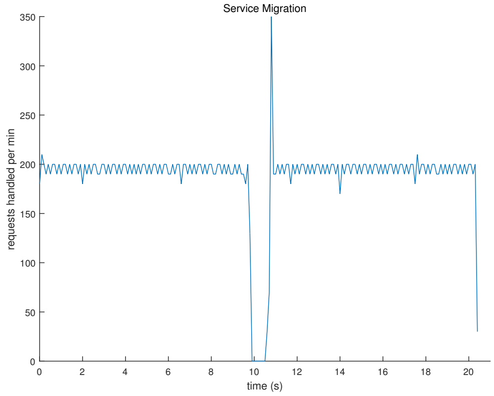

Fig. 4: Service Migration Impact on Request Handled Rate

C. Service Scalability

The service scalability refers to the performance scalability as the service scale growth. Best-route is the default strategy of NFD, which means top performance faces would be chosen to be forwarded. For real cases, requests of different name prefixes will be forwarded to different service hosts. In this section, the forwarding strategy is set to be random to emulate requests of different prefixes. The requests can be randomly and evenly be forwarded to different hosts. Multiple clients send 1000 requests per second. 2 service hosts connected to the same NFD start to handle the request at first. At 10s, other 2 service hosts connected to another NFD are started. Figure 5 shows the performance growth. Although the performance scalability conforms to NFD routing capability, the NSN-NDN architecture guarantees the forwarding would not be restricted to one node bottleneck.

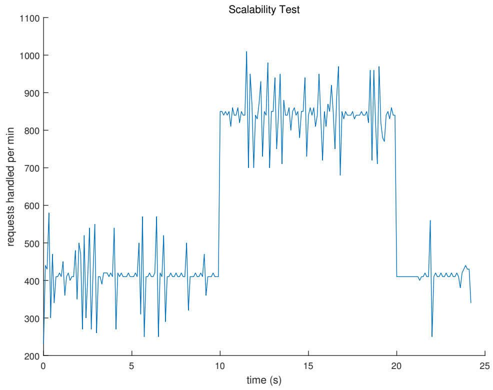

Fig. 5: Service Scalability Test

VI. Conclusion

This paper proposes design of named service network (NSN). Security, service name resolution, semantics and mediation issues are discussed as necessary elements to construct feasible and scalable service network. An implementation of NSN over NDN is demonstrated. Evaluation is done to show the benefit of building service network upon NDN.

In future, we will continue working on NSN-NDN to construct a more scalable testbed. Further research will be done on better utilizing features of ICN to optimize NSN.

Acknowledgement

This work was supported in part by National Natural Science Foundation of China (grants No. 61472200 and No. 61233016), Ministry of Science and Technology of China under National 973 Basic Research Program (grant No. 2013CB228206), and S&T projects of State Grid Corporation of China.

References

[1] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *Communications Magazine, IEEE*, vol. 50, no. 7, pp. 26–36, 2012.

[2] T. Braun, V. Hilt, M. Hofmann, I. Rimac, M. Steiner, and M. Varvello, "Service-centric networking," in *Communications Workshops (ICC), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–6.

[3] D. D. Clark and D. L. Tennenhouse, "Architectural considerations for a new generation of protocols," *ACM SIGCOMM Computer Communication Review*, vol. 20, no. 4, pp. 200–208, 1990.

[4] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. D. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, D. Massey, C. Papadopoulos *et al.*, "Named data networking (ndn) project," *Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC*, 2010.

[5] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture," in *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4. ACM, 2007, pp. 181–192.

[6] D. Lagutin, K. Visala, and S. Tarkoma, "Publish/subscribe for internet: Psirp perspective." *Future Internet Assembly*, vol. 84, 2010.

[7] T. Braun, A. Mauthe, and V. Siris, "Service-centric networking extensions," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*. ACM, 2013, pp. 583–590.

[8] C. Bussler, "The role of b2b engines in b2b integration architectures," *ACM SIGMOD Record*, vol. 31, no. 1, pp. 67–72, 2002.

[9] D. Fensel and C. Bussler, "The web service modeling framework wsmf," *Electronic Commerce Research and Applications*, vol. 1, no. 2, pp. 113–137, 2002.

[10] C. Bussler, "B2b protocal standards and their role in semantic b2b integration engines," *IEEE Data Eng. Bull.*, vol. 24, no. 1, pp. 3–11, 2001.

[11] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*. ACM, 2009, pp. 1–12.

[12] G. Wang and T. E. Ng, "The impact of virtualization on network performance of amazon ec2 data center," in *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–9.