An Ad-hoc Distributed Reasoning Scheme for Content Centric Networking

Aiping Yi, Junnian Wang
School of Information and Electrical Engineering
Hunan University of Science and Technology
Xiangtan, China
apply@hnust.edu.cn, wangjunnian@sina.com

Zhuhua Liao
Key Lab of Knowledge Processing & Networked Manufacturing
Hunan University of Science and Technology
Xiangtan, China
zhliao@hnust.edu.cn

Abstract—Intelligently discovering knowledge and principles hidden in massive amounts of distributed and dynamic sources is a challenging issue, since the connectivity of sources and mobility of contents are frequent changing. For achieving intelligently discovering knowledge and principles in network environments, it requires a strong adaptive mechanism of distributed reasoning to discover the indirect and dynamic semantics in content-centric networking. In this paper, we propose a novel ad-hoc scheme for distributed reasoning by extending the name-based routing mechanism in content centric networking. The ad-hoc scheme of distributed reasoning is a native process of the content-centric networking. First, a two-stage reasoning mechanism is presented for reasoning query forwarding and knowledge reasoning. And then for guiding original or evolved queries to all right sources and supporting the knowledge integration in the way back, a recursive reasoning procedure and various reasoning rules are introduced.

Keywords—Content Centric Networking; Distributed reasoning; Ad-hoc communication; Knowledge Integration; Relation-al Routing

I. INTRODUCTION

Within last decade, P2P network, cloud computing and content centric networking have achieved tremendous developments and various data sharing infrastructures have been built to retrieve a variety of contents for personal users. The knowledge integration is used to integrate the fragmented knowledge together so that the knowledge islands are cross-linked for stimulating the evolution, dissemination and application of knowledge, a much expected and complete knowledge discovery and integration. The knowledge integration from controlled to fully autonomic in the network.

In recent years, although the distributed query, distributed reasoning and ontology technologies drive away at better data sharing and knowledge retrieval, there still have a very poor performance for semantic data query and knowledge discovery in the open, dynamic network. It requires a mechanism promoting the graceful evolution of semantic integration from controlled to fully autonomic in the network.

Nowadays, the content centric networking has emerged where a network is viewed as dynamic storage of information and provides end-to-end communications in terms of named contents instead of traditional IP-style addressing and location[1]. Using these new networking approaches, we can get the following advantages: content-aware routing and searching data locally[2], [6], [7]. Especially, the content centric networking protocol (CCN) [2] has designed a formidable networking architecture built on IP’s engineering principles. However, these methods mainly highlight retrieving objects by name, rather than reasoning capability for knowledge discovery and integration in content centric networking.

For efficiently discovering distributed knowledge and achieving autonomic knowledge integration in content centric networking, we propose an ad-hoc distributed reasoning scheme based on the CCN protocol. The main contributions of this paper are the following: 1) We state the query-driven reasoning representation for ad-hoc distributed reasoning to suit various requests of individual users. 2) We propose a semantic guiding approach for delivering reasoning queries to hidden sources and integrating distributed results in content centric networking. 3) We present a two-stage reasoning mechanism, which including a recursive reasoning procedure and supporting various reasoning rules, for guiding reasoning query to all right sources and supporting the ad-hoc distributed reasoning.

Based on the ad-hoc distributed reasoning called AdrCCN, we could construct more powerful knowledge discovery applications in content-centric networking, such as duly inferring potential data sources, integrating the distributed existing knowledge and discovering new knowledge which meet individual user’s goals. The traits of the ad-hoc dis-
tributed reasoning are: distributed reasoning model built-in dynamic networks; formal semantics and provable inference for declarative reasoning query; extensible concept-based classification and relation for ad-hoc inter-operability in open networks.

II. RELATED WORK

In the past years, many researchers have been devoted to efficient sequential pattern mining [19] in the Internet. The goal is to detect patterns which are comprised of sequences of sets. In the researches, the frequent pattern mining has occupied a dominant position. The frequent patterns are itemsets, subsequences, or substructures that appear in a data set with frequency no less than a user-specified threshold. However, the bottleneck of frequent pattern mining is that it is difficult to derive a compact but high quality set of patterns that are most useful in applications [20]. In recent years, researchers with strong industrial engagement have realized the need to shift from “data mining” to “knowledge discovery” [21]. In the paper, we propose an efficient knowledge discovery and integration mechanism in dynamic content centric networks for personal users.

The Semantic Web [12] mainly addresses the web data representation and relation reasoning between data objects in static web. For example, using the Semantic Web technology [17] to compose disorderly data over the Web. The folksonomy mining [8] and ontology technologies [9] mainly figure the concept of data and the relation between concepts. Moreover, the ontology technology is concentrated in ontology establishing [10], ontology mapping [11]). The semantic link network (SLN) [13] is a loosely coupled semantic data model for managing web resources and supporting complex reasoning. But above technologies have not considered the network dynamics and data variants in large, open networks.

To the layered semantic overlay networks (SONs) [3], nodes with semantically similar content are “clustered” together. There are challenges when building SONs: needed to be able to classify queries and nodes, to decide the level of granularity for the classification, to maintain the SONs in the network with dynamic nodes and variable data. In the SQPeer middleware [4], each peer broadcasts or advertises its data schema which includes all RDF classes and properties to (or requested by) other peer nodes. The SenPeer [5] builds a two layered semantic overlay network, which uses a distributed expertise table in superpeers and uses ontologies in peers which connects superpeers, for selectively forwarding semantic query. Although these solutions can deal with the semantic query in dynamic networks, they are not effective solutions for poor scalability and maintaining a semantic overlay network in the open, dynamic network.

Recently many reasoning methods have been proposed in distributed networks. Existing approaches including the use of P2P architectures have focused on the distributed data reasoning [14], [15] and the MapReduce paradigm [16] has explored the efficient large-scale distributed reasoning. But so far they only support the simple deduction rules and need maintaining the rigorous network structure or using clumsy knowledge discovery methods in the dynamic network.

Besides above reasoning methods, the probabilistic reasoning methods can be also used to chain the disorderly content, such as the Bayesian networks [18]. But it may draw a mistaken conclusion.

To date, many people have studied content retrieval in content centric networking. Content Centric Network protocol (CCN) [2] provides network-scale content caching and user-friendly, hierarchical names for routing. The content publisher provides the name of the data object and adds digital signature to confirm content availability. Intermediate routers can validate an incoming packet using the digital signature and employ a longest-match lookup on the name to decide about the forwarding of the request. If there are multiple sources announcing content, unlike IP forwarding, the request is forwarded to all these sources. CCN routers provide content caching within the network and thus any CCN routers on the network can reuse the content. Because CCN realizes data retrieval by a name-based routing protocol, instead of using a host-based addressing scheme, in a content-centric-network which could be infrastructure-free, in this paper we explore the ad-hoc distributed reasoning in content-centric networking by extending CCN, independently of the infrastructure used.

III. AD-HOC DISTRIBUTED REASONING FRAMEWORK

A. Conceptual view of ad-hoc distributed reasoning

The purpose of AdrCCN is performing the reasoning both in forwarding query and in backtracking the integrated response results in the dynamic network according to the semantics of reasoning query. More specifically, it should resolve three problems: (1) to efficiently forward reasoning query to all right sources; (2) to duly issue the evolved queries, which is generated by reasoning based on a original query, for discovering more distributed knowledge; (3) to duly reason knowledge and prune the redundant data. The conceptual view of ad-hoc structure for semantic integration is shown in Figure 1. In the figure 1, the user1 wants to retrieve the objects but the relation constraint between the object a and each of them is R1, or wants to retrieve the objects but the relation constraint between the object d and each of them is R3, from the dynamic network. For pursuing both the capability and the efficiency of knowledge discovery, the user can set the times of association by the signature and employ a longest-match lookup on the name to decide about the forwarding of the request. If there are multiple sources announcing content, unlike IP forwarding, the request is forwarded to all these sources. CCN routers provide content caching within the network and thus any CCN routers on the network can reuse the content. Because
guiders on the way back and then the guiders will join efforts to perform some reasoning according to the integrated data. Finally, the results will be backtracked to the client, and then the final integration and reasoning are performed according to the requests of user.1. The final results included the distributed knowledge and implicit knowledge (R1 : (a, b), (a, c), (a, h), R2 : (d, e), (e, f), (f, g)) that are discovered by the distributed reasoning. So, the ad-hoc distributed reasoning model can collaborate with multiple guiders to perform reasoning for implementing relation reasoning for individual goals, such as discovering implicit relations, deriving new relations and combining a series of distributed data by explicit and indirect relations, which free from the change of connections between nodes and data mobility over network.

B. Semantic Guider

Since the semantic relations depend mainly on the data object itself, for the semantic data in content-centric networking it is natural to use the name of a data object as well as its associated properties to identify and summarize its content. We thus propose a semantic guider in dynamic networks that is able to give multi-granularity “directions” for finding or deriving the right sources by a referenced object and integrating the distributed results on the way back for reasoning new queries or discovering hidden knowledge. For implementing this purpose, it requires the guiders interconnected in an ad-hoc manner. A hierarchical namespace is very useful when one does not know the exact name of a data object but has in mind only a data category [22], and the hierarchical division partitions the large distributed data set into subgroups and the subgroups may again map into a hierarchical distribution. Therefore we use the hierarchical namespace which composed of the classification names or concepts at high level and the instance names as the lowest level (e.g. C/subC/objectname). As a consequence, each guider node only needs to hold the “hierarchical prefix” of the name of data object rather than all semantic properties to direct queries to right sources. The structure of the semantic guiders in a network can be briefly illustrated as shown in Fig. 2. A user’s query can be represented by a referenced object, also called subject, (represented as a hierarchical name) plus constraints on the subject. The query will be forwarded by guider nodes towards the potential sources according to the referenced subject and the conclusion of some reasoning. When the query finally arrives at the right sources, the constraints are used to find the exact data objects that the user wants. The results will be converged and then the reasoning will be performed, which allows for discovering new relations or emitting new queries, on their way back.

In order for different semantic guider nodes communicating or cooperating with each other, it is necessary that all semantic guiders conform to the same convention, i.e., metadata about names, relations and properties of data on all semantic guiders should be uniformly defined. The interaction between semantic guiders and the heterogeneous data sources, however, requires rewriting of both, queries and results.

C. Forwarding and reasoning engine

For implementing the ad-hoc distributed reasoning in the dynamic network, we design an engine to honor the query forwarding and knowledge reasoning by extending CCN [2]. Because its name-based guided information flooding routing models fits well in dealing with our problem, we use the CCN protocol as basic tool. The main component of the engine is the guider node model.

The roles of a guider node are to guiding reasoning queries to right sources, reasoning new knowledge by the results that returned from different sources and backtracking all relevant semantic objects or relations to correct clients. The general structure of the new engine, which is building on every guider node, is shown in the Figure 3. Besides the PIT (Pending Interest table, which we call Pending Query table in this paper), the FIB (Forwarding Information Base) and the Content Store which are included in CCN, two new data structures, called Response Relation Table (RRT) and concise knowledge base (CKB), are added to the engine.

In the engine, the FIB is used to store the classifications or hierarchical names for forwarding users’ queries to potential sources. So it only needs the data sources advertising the hierarchical name of data. Moreover, the engine adopts some techniques of CCN to deal with the query, for example, using the prefix-longest match to find the matched outfaces in

Figure 1. The ad-hoc structure of distributed reasoning.
FIB, the “nonce” option in Pending Query table (PQT) to avoid query loop forwarding.

The reasoner, integrator, forwarder and emitter are important functional modules in the engine. The reasoner is used to infer the potential sources for every query and derive new relations between some retrieved data. The CKB includes the common concepts (also included synonym and antonym of some global vocabularies) and relations, reasoning rules, the properties of relations (e.g. reflexivity, antisymmetry and transitivity) and etc. So, a combination of the reasoning knowledge, the hierarchical name carried in queries and the data cached or integrated can make stronger relation reasoning in the application routers. The integrator is responsible for chaining the fragmented data by relations and integrating the heterogeneous data while the forwarder is used to forward the received queries and the emitter is used to issue new queries according to the reasoning results.

When a user issues a reasoning query to a network, the ad-hoc distributed reasoning process is as follows: The reasoning query will be sent out to the closest guider nodes, and the guider node that received the query will decide whether to send back the reasoning results by CKB according to the cached data or to reason the outfaces and forward the query to corresponding neighbors. If a guider node receives some response packets, it will first lookup the cached data in ContentStore and make some reasoning and integration based on the related cached data and the new data. Then it will lookup the incoming face of the corresponding pending query and send back the integrated data and newly derived data towards its requester(s). In order to suppressing duplicated results, the response data will be recorded in RRT and establish a link with the corresponding pending query. The RRT record mainly includes the head information of response packets, such as the query name, infaces and the sources of these results.

IV. Statement of Semantic Reasoning

Because of the dynamics of network and variations of content, the vast hidden knowledge will be evolving and changing in the open, dynamic network. So we need not to collect all semantic data to build the global semantic overlay network over all sources for knowledge discovering, but implement the ad-hoc distributed reasoning that adapts the dynamics of network and variations of content and supports individual reasoning goals, such as query-driven reasoning and integration in content-centric networking. For implementing the reasoning in the network, the individual reasoning goals should be stated declaratively, and then they were decomposed and delivered to all potential sources. The algebraic representation of the semantic reasoning is explored.

A. Relations of distributed data

The relation between data can be classified from various aspects. In semantic aspect, it mainly has inclusion, cause-effect, imply, similarity, reference, whole-parts, supplement and contrast relations. From time aspect, it mainly has sequence, precursor/successor and time distance relations. From spatial aspect, it mainly has upper, middle, below and space distance relations. We can formally define the relation between data.

Definition 1. If a data set is satisfied one of the following conditions:
(a) The data set is not empty, and its elements are all ordered pairs with same semantic relevance;
(b) The data set is empty.
We name the data set as a binary relation and denoted as $R$, called relation. If $<x, y> \in R$, it denoted as $R(x, y)$; if $<x, y> \notin R$, it denoted as $R(x, y)$. For example $R = \{<a, b>, <a, c>\}$ can be denoted as $R(a, b), R(a, c)$. $R$ is a uniform representation of semantic relation for ordered pairs.

In each source local objects are most likely to set up semantic relations, called local relations. Thus, the main task of the ad-hoc distributed reasoning is to efficiently draw the global relations from multiple related local relations across the open, dynamic network.

B. Query of semantic reasoning

For delivering the reasoning query, a referenced subject should be specified in line with the CCN. So we declare that a reasoning query comprises multiple reasoning constituents, such as reasoning expression and referenced subject (also called domain or object). Generally, for performing a reasoning, a user should give the premise of reasoning and state briefly what kind of knowledge expected. To exactly formulate what information should be reasoned in the semantic reasoning, we define them as follows:

(1) Reasoning a relation between two objects: $a \sim R[(a, b)$ or $b \sim R[(a, b),$ where the object before the “~” means referenced object;

(2) Reasoning the successors of a object with a relation constraint: $a \sim R[(a, ?b) = \{b\}R(a, b)];$

(3) Reasoning the precursors of a object with a relation constraint: $b \sim R[(?a, b) = \{a\}R(a, b)];$

In an object set $A$, there are may be multiple relations. Generally, through the known relations, we can reason some new relations.

Definition 2. Relation composition. If there are two relations $R$ and $S$ on object set $A$ and the two relations are transitive, the relation composition by the two relations is: $R \circ S = \{<x, z> \mid \exists y \{<x, y> \in R \wedge <y, z> \in S\}\}$. So $R(a, b) \circ S(b, c) = R \circ S(a, c), R(a, b) \circ R(b, c) = R \circ R(a, c).

To exactly formulate what components should to be reasoned in the relation composition, we define:

(1) Bridging relation reasoning: $a \sim R \circ S[(a, c) = \{b\}R(a, b) \wedge S(b, c)].$ It means to discover all objects $b$ which satisfy $R(a, b)$ and $S(b, c);$ 

(2) Joint relation reasoning: $b \sim R \circ S[b = \{(a, c) \mid R(a, b) \wedge S(b, c)\}.$ It means to discover all object pairs $(a, c)$ which satisfy $R(a, b)$ and $S(b, c)$.

Definition 3. Assume the relation $R$ is transitive on object set $A$ and $n$ is positive integer, the relation power $R$ is:

(a) $R^0 = \{<x, x> \mid x \in A\} = A$;

(b) $R^{n+1} = R^n \circ R$.

In fact, the relation power is a complex relation composition that performed $n$ times of relation composition with mono-relation. To exactly formulate what components should be reasoned in the relation power reasoning, we state the reasoning query in the following:

(1) $a \sim R^n[(a, ?x)$ indicates to discover the successors by performed $n$ times of relation composition with relation $R$ and the precursor $a$.

(2) $b \sim R^n[(?x, b)$ indicates to discover the precursors by performed $n$ times of relation composition with relation $R$ and the successor $b$.

(3) $b \sim R^n[(?x, b, ?y)$ indicates to discover the object pairs (precursor - successor) and the relation of each pair is $R$ by performed $j(0 \neq j \leq n)$ times of relation composition with relation $R$ and the successor $b$, and performed $n - j$ times of relation composition with relation $R$ and the precursor $b$. If the relation $R$ is reflexive, the $b \sim R^n[(?x, b, ?y) = b \sim R^j[(x, b) \sim b \sim R^{n-j}(b, ?y)].$ If the $j = 0$, the $b \sim R^n[(?x, b, ?y)$ is equivalent to $b \sim R^0[(b, ?y)$. If the $j = n$, the $b \sim R^n[(?x, b, ?y)$ is equivalent to $b \sim R^n[(x, b)$.

(4) $a \sim R^{\geq n}[a, ?y) = \cup_{i=1}^{n}a \sim R[a, ?y].$ It collects all relations that reasoned by 1 to $n$ times of relation composition with relation $R$ and all started with object $a$.

In the process of relation power reasoning, it can derive deep hidden objects in which each object and its neighbor objects are associated by a relation.

V. A TWO-STAGE REASONING MECHANISM

For implementing the ad-hoc distributed reasoning, we present a two-stage reasoning mechanism, that is, reasoning relations between objects based on prefix or integrated data. In the Figure 4, the dotted arrows indicate the process of the query forwarding from the user to the potential sources. In query forwarding, each router will use the prefix-based reasoning to infer that where the query is forwarded to based reasoning can derive the potential sources and reduce the recursive process. Finally, the results will be returned to the client end for the user (e.g. $R3 : (d, e), (d, f), (d, g)$). It should be noted that the integrating and reasoning procedure varies with the reasoning query.

A. Prefix-based reasoning

1) The framework of prefix-based reasoning: The prefix-based reasoning can derive the potential sources and reduce
If there are two objects and known the relations between their respective prefixes (or the relations between the classifications that respectively represented by their prefixes), we can deduce the relations between the two objects, which called classification relation reasoning. Using this reasoning, we can derive more matched prefixes by a referenced subject. For example:

$$E/G/c\ A/B/d\ R(E/G,A/B)\ \begin{cases} \rightarrow R(E/G/c,A/B/d) \end{cases}$$

B. Integration-based Reasoning

To the integrated data, there are two reasoning phases: performing reasoning in guiders or in sources respectively. The reasoning in sources is meaning that when a reasoning query is forwarded to a data source, it can derive expected knowledge according to the reasoning query by using the existing reasoning techniques. The reasoning in guiders means that it will try to discover new knowledge from different response data or determine whether to issue a new query. The reasoning in guiders can discover the knowledge after integrated the distributed data and duly prune the redundant data. In the section, we mainly explore the integration-based reasoning approaches in guiders.

1) Recursive reasoning procedure: According to the incoming time and semantics of response results, the guider nodes can derive new relations and integrate relevant reasoning results. But to some query, only received some interim results can it evolve some new reasoning queries for discovering more hidden knowledge or more needed results than the original query does. For example, in the Figure 6, if the guider1 derives a new relational data $R_3(e, f)$ after it received the response relational data $R_3(d, e)$, it will issue two new reasoning queries $e_R_3(e, ?x)$ and $f_R_3(f, ?x)$ for the deeper knowledge. Since a original query may need consecutively issue new reasoning queries for the deeper knowledge in the content-centric networking, we develop a bootstrapping procedure for automatically updating the old reasoning query and issuing new reasoning queries with the reasoning performed, which we call this procedure a recursive reasoning procedure.

For determining the termination of recursive reasoning, the maximal deepness of reasoning can be set to a reasoning query. For example, if a user want to discover the associated data that is less than or equal to the third power of data $d$ with $R_3$ relation constraint, the user will issue the reasoning query $d_R_3 \leq 3(d, ?x)$. When the $R_3(d, e)$ returned to the guider1, the deepness of the reasoning will be subtracted 1 and it will issue a new reasoning query $e_R_3 \leq 2(e, ?x)$. When it derived the new relation $R_3(e, f)$, the deepness of the reasoning will continue to be subtracted 1 again and will issue the reasoning query $f_R_3(f, ?x)$. For correctly returning the response results of the new reasoning queries, it only adds the new query names to the old query packet rather
than recreate the new query packet. In guider nodes, they record the query with the old query name in PQT, but still use the new query name to reason. So the results returned by the new reasoning queries will be correctly backtracked to the user.

2) Reasoning rules: To the integrated data, we define some practical reasoning rules in this section for improving the capability of knowledge discovery. The main work of the single relation reasoning is to deduce whether a relation on some data. For example, if there are two data a and b, it will use the existing reasoning knowledge to infer whether it is R(a, b), such as first-order logic based on the concepts and relations about them.

(1) Reasoning rules based on Relation characteristics. According to the characteristics of relation, we can infer that there are new relations among some data For example, the reflexive, transitive and symmetrical characteristics of relation.

(2) Reasoning rules based on relation power
The reasoning based on relation power can discover indirect relations by different ways. For relation R1 and R2 on data set A, we know there is R1^0 = R2^0 = A and R1^1 = R. So there are some new reasoning rules:
(a) Forward reasoning: R^n = R \circ R^{n-1};
(b) Backward reasoning: R^n = R^{n-1} \circ R;
(c) Bidirectional reasoning: R_{n=j+k+1} = R^j \circ R \circ R^k.

(3) Reasoning rules based on inverse relation
Generally, every relation has its corresponding inverse relation. For example, the relation pairs: precursor and successor, inclusion and inclusion by, cause-effect and effect-cause are the inverse relation with each other. Assume two relation R1, R2, the relation R1 is expressed by antonym of the relation R2, we call the R1 and R2 are inverse relation with each other and they can be denoted as R1 = -R2 (or R2 = -R1). Using the inverse relation, we can reason some new relations, for example:
(a) If there is R(a,b), then R(b,a)=-R(a,b);
(b) If there are R(a,b) and R(b,c), then R(a, c) = R(a, b) \circ R(b, c);
(c) If there are R(a,b) and R(c,b), then R(a, c) = R(a, b) \circ (-R(c, b));
(d) If there are R(b,a) and R(b,c), then R(a, c) = -R(b, a) \circ R(b, c);
(e) If there are R(b,a) and R(c,b), then R(a, c) = (-R(b, a)) \circ (-R(c, b)).

VI. IMPLEMENTATION

The AdrCCN system mainly include two parts: one is the forwarding engine, and it realize the reasoning queries forwarding and results forwarding; Other is the reasoning component in each guider, and it realize the relations reasoning in accordance with the declared rules. The system is an on-going implementation. For the forwarding engine, we have developed it on an overlay network in a PC Cluster. In the reasoning component, the class names, individualities, properties and relations should be defined firstly. More importantly, the reasoning rules should also be declared. And then, implement the reasoning process for reasoning query in the course of forwarding and data integration in the way back. Since the forwarding engine inherits the CCN, the performance of routing and forwarding is approximate to those of CCN when the cost of reasoning is neglected. For build the CKB, we have referred to the OWL language\(^2\) to define the class names, individualities, properties and relations about the computer science, such as Computer Network, DataBase Management. For the reasoning component, we have developed some rule reasoning algorithms based on the implement mechanism of some reasoning framework (e.g. Jena\(^2\)). So the rule reasoning algorithms can act in obedience to the declared rules.

VII. CONCLUSION AND FUTURE WORK

In this paper, we present a novel ad-hoc scheme for distributed reasoning in content centric networking. The scheme can perform semantic reasoning when the response results on the way back. With this scheme the semantic reasoning becomes a transparent operation to the end user. Using the reasoning rules and reasoning mechanisms in guiders, it can efficiently discover potential sources and hidden and distributed knowledge in the dynamic network without knowing where these fragmented semantics and contents are stored. Moreover, the irrelevant results will be pruned at data sources, and duplicated results will be suppressed at intermediate guiders. In the future, we will evaluate the performance of the AdrCCN in the large-scale and dynamic network and hope it will have good opportunity to be incrementally deployed.

Certainly, it should be acknowledged that there are many possible optimizations that can be made to this scheme, for example, the performance of forwarding in which act in association with reasoning, optimization of this scheme for high scalability and resilience, the performance evaluation should be carried out to justify the appropriateness of

\(^2\)http://www.w3.org/TR/owl-features
complex distributed reasoning. We believe that it will benefit to build a knowledge-aware network over the large-scale and dynamic content-centric networking that crowded the distributed, variable and fractional data.

ACKNOWLEDGMENTS

This work is supported by the Sub-project of the new generation broadband wireless mobile communication network of the National Science and Technology Major Projects Grant #2011ZX03002-002-03 and the Beijing Natural Science Foundation Grant #4112057.

REFERENCES


