

Memory-enabled Autonomic Resilient Networking

Bassem Mokhtar
The Bradley Department of Electrical and
Computer Engineering,
Virginia Tech
Virginia, USA
bmokhtar@vt.edu

Mohamed Eltoweissy¹
Pacific Northwest National Laboratory
Washington, USA
mohamed.eltoweissy@pnnl.gov

Abstract—It is becoming increasingly challenging for the contemporary Internet to satisfy the explosion in networking services and applications with their widely diverse and dynamic QoS requirements and resource demands. For smarter networking management and operations, we hypothesize that next generation networks should be endowed with “memory”-like (i.e., functionally similar to human memory) capabilities and should explicitly manage their resources and services. In this paper, we introduce SmartNet, a new memory-enabled autonomic network architecture that is scalable, resilient and adaptable at runtime. SmartNet’s memory, we term NetMem, manages data related to multiple networking concerns, namely application, communication, and resource concerns. SmartNet’s NetMem maintains “knowledge” – the fourth networking concern in our networking model – in a distributed knowledge plane that can be rapidly accessed in a simple and systematic way by SmartCells (SmartNet’s basic building block) and the services they implement. Each SmartCell has intrinsic monitoring, control and analysis capabilities and can acquire resources on demand. For scalability and to facilitate inter-communications and services’ support between SmartCells, regions, termed CellDomains, may be dynamically formed. CellDomains would incorporate favored services or resources. Our simulation results demonstrate an improvement in networking operations with our memory-enabled network management.

Keywords—Next Generation Networks; Autonomic Management; Knowledge-enabled Management; Network Resilience; Biologically-inspired Networking; Cell-Oriented Architecture.

I. INTRODUCTION

Current Internet architecture that has changed the world through its seemingly expansive networking functionalities for the past few decades, has not considered resilient adaptive operations based on autonomic and explicit resource and knowledge management [1]. Since inception, the Internet has been founded on a layered architecture employing the end-to-end principle and store and forward mechanisms to provide best effort service internetworking intelligent end-nodes through a dump core of nodes. Recently, many evolutionary cross-layer networking enhancements and clean-slate architectures have been proposed, for example [2-5]. These attempts have presented new design principles for architecting the next generation networks, proposed new basic building blocks for the architecture, and advocated separating the control plane from the data plane. We conjecture the need for

network “memory” and a self-managing, “memory”-enabled network architecture with explicit resource and knowledge management. Such architecture will, by design, enable autonomic, resilient, adaptable, and evolvable operations. To the best of our knowledge, emerging networking architectures have not considered explicit autonomic resource and knowledge-enabled network management. In this paper, we propose the SmartNet architecture, which comprises composable basic building blocks, termed SmartCells, with intrinsic monitoring, analysis and control capabilities.

A SmartCell, as described in more detail shortly, is a biologically-inspired autonomous computing and communication entity. It is an abstraction of resources that can reserve available physical or logical resources on demand and can monitor its state and surrounding external context besides utilizing shared knowledge from the network “memory” for self- and situational-awareness and action. A group of SmartCells offering similar services and/or assigned resources may dynamically form a domain, a CellDomain. Each CellDomain has a unique identifier, and assigns internal addresses and authentication mechanisms for its member SmartCells. These domains are self-managed and their hierarchical recursive construction would enhance scalability and management of resources and services in the network.

For smarter networking operations and knowledge-based management, SmartNet possesses a “memory”; we call “NetMem”, realized as a distributed knowledge plane (KP). That is shared and accessed by communicating entities at runtime and on demand. NetMem will inform networking operation with relevant networking data, for example protocol and service semantics. NetMem maintains data gathered from networking entities and experts. These data pertain to the primary networking concerns: application, communication and resource concerns as defined in [2]. NetMem enables communicating entities to store, discover and retrieve relevant data on other entities at runtime and on demand to meet variations in QoS requirements and occurrences of abrupt events such as resource shortages and security threats.

We are inspired in NetMem’s design with human memory and related functionalities. Knowledge stored in the SmartCells and that in NetMem constitute the “network memory”. Knowledge-based management (KBM) supports resilience, adaptability, and evolvability of network operations. More details on SmartNet architecture, KBM, and structure of NetMem will be provided in section III. We constructed a

¹ Also affiliated with the Bradley Department of ECE at Virginia Tech.

simple application scenario for a preliminary evaluation of some NetMem-enabled networking operations. We applied a rudimentary KBM over a simple sensor network as will be discussed in the evaluation section.

Our contributions in this paper can be summarized as follows:

- Memory-enabled network architecture with composable building blocks that are, by design, resilient, scalable, adaptable, and evolvable;
- Architecture for a distributed shared self-managed knowledge pool (memory) of large sets of networking concerns' data that can be rapidly and efficiently accessed; and
- Enhanced scalability and resilience by elastic (right-sized) explicit resource management and dynamic formation of self-managed domains of services and resources.

The remainder of this paper is organized as follows. Section II provides the related work. Section III gives an overview of the SmartNet architecture, KBM, and SmartNet's NetMem. Section IV presents the evaluation and discusses the obtained results. The paper concludes in section V with an outline of future work.

II. RELATED WORK

Many proposals are emerging for both clean-slate networking architectures (for example, initiatives for future Internet by US-NSF FIND, EU FIRE, and AKARI in Japan-ASIA) [6] and evolutionary enhancements to the current Internet architecture (such as the proposals for cross-layer designs [7]). All of these efforts aim to avoid today's Internet limitations such as rigid layering and addressing, ambiguity in protocol operation, best effort services, lack of intelligence in the network core, and lack of support for delay tolerant communications [1]. In addition, these attempts consider novel design principles to enhance networking operation for example: a) moving away from the rigid layered architecture which causes barriers among constituting protocols' operations; b) self-* properties in basic building blocks of networking architectures; and c) decoupling the control and data planes.

Some research works introduce interesting innovations such as the one by the open networking foundation. That project depends on software defined networking technologies like the open flow [25] to have networks managed by operators who can program these networks via certain devices as routers and switches to attain and ensure secure services with a quality of service level.

Some exertions for implementing new networking architectures used networking abstractions of Internet as in [7] aiming to enhance networking protocol operations and interactions. Others used new networking abstractions as our work in [2] which proposed a reference model and associated biologically-inspired architecture for a federated network. In [2], the network is viewed as a complex adaptive and evolvable system handling application, communication and resource concerns. Cross-cutting federation among communicating

entities provide for cooperative optimization and adaptation. Bouabene *et. al* in [3] proposed an autonomic network architecture "ANA" with a flexible design by supporting functional composition and scaling at runtime. Functional composition is based on local view from the nodes; and there is no global management for resources and services. Protocol operations can adapt according to surrounding environment. Authors in [4] proposed principles for a generic autonomic network architecture "GANA" whose elements possess monitoring capability and some autonomic behaviors such as in routing and QoS management. Researchers in [5] provided a new architecture for serving future Internet where they aim to have a stable, scalable, and secure network architecture. They considered the network as inter-process communication platform and the target of any connection is to serve an application. So, the basic building block is an application process. In [8], a clean-slate network architecture was proposed to have ubiquitous services in pervasive and ubiquitous computing environments. It was designed to have flow oriented context-awareness at execution time. Authors in [9] proposed MANA where they added control functionalities to meet demands of new services and protocols. MANA depends on networking services in a non-layer architecture design using programmable interfaces that can interact together. The work in [10] introduced a cross-layer composition network architecture. The traditional networking stack was decomposed into functional blocks. Every networking service will be provided via a composition of those blocks.

Some related work has touched on issues related to knowledge-based management. Our proposed biologically inspired architecture in [2] included intrinsic monitoring and regulatory units into its basic building block (network cell). Those units can send at runtime recommendations upon detection and analysis and/or self-knowledge to the embedded execution unit. The proposed ANA architecture with its autonomic capabilities in [3] has information dispatch tables within nodes where these tables define locations of services and protocols in other nodes. In GANA, monitored information is stored locally and shared between decision elements inside communicating nodes for enhancing autonomic capabilities as QoS management and decision making. The author in [11] provided concepts to allow efficient network management based on distributed information plane and reasoning and computation capabilities. But, there were no mechanisms on how to implement the information plane and its related tasks as information storing, discovery and retrieval. Also, no evaluation was provided. The introduced work in [11, 12] proposed an idea based on artificial intelligence and cognitive techniques to add a distributed KP to the current Internet to be self-knowledgeable, self-configuration, self-diagnosing, self-analyzing, and self-managing. There was no management facility based on the proposed KP and no evaluation was provided. In [13] authors proposed a non-shared centralized KP inside each communicating entity in mobile wireless ad hoc networks. That KP helps entities in storing information related to protocols of the networking stack and other entities. The proposed knowledge system facilitates protocol interactions. Also, it requires periodic messages to update information registered in communicating nodes; and this causes additional communication overhead. Transmission power control aware

protocols were designed to improve the operation's performance. Riggio *et. al* in [14] proposed a KP for wireless mesh networks where communicating nodes can exchange information related to services and operating conditions. The main objective was to enhance the autonomic management of such networks and their adaptation to surrounding circumstances. In [15], Shieh *et. al* proposed a global federated KP for Internet where that plane provides trustworthy information regarding networking properties that will be used by running applications. The proposed KP was implemented via a group of servers operated by third parties to support security issues (confidentiality and privacy). There are defined access control policies which are used for information disclosure.

In all of the above-mentioned works, there are no methods for resilient and adaptable operation based on autonomic and explicit knowledge and resource management. We propose SmartNet to address this gap.

III. OVERVIEW OF SMARTNET

SmartNet is a self-managing knowledge-enabled network architecture whose components are designed to explicitly manage the four networking concerns; namely application, communication, resource, and knowledge concerns. SmartNet is endowed with “memory”, NetMem, realized as distributed KP and maintains network concerns' data. NetMem can be rapidly accessed in a simple and systematic way by the SmartCells, SmartNet's basic building block, and the services they implement. NetMem will have to some extent similar functionalities to human memory. The shared NetMem provides knowledge services to communicating SmartCells related to networking concerns such as information about running services and their related concerns' data. Each SmartCell has autonomic monitoring, analysis, and control capabilities based on knowledge it gains by accessing its local memory, shared constructed dataset tables in NetMem, and runtime observations from surrounding internal/external contexts. A SmartCell is an intelligent self-contained computational- and decision-capable communication entity where it can monitor its states and surrounding external context for self- and situational-awareness. Fig. 1 shows the internal structure of a SmartCell as inspired by biological cells and our network cell design in [2].

As depicted in Fig. 1, each SmartCell has its intrinsic execution unit (EU), resource unit (RU), and interface unit (IU). A SmartCell also incorporates a “memory” or knowledge unit (KU). The EU executes the algorithms for the different running protocols (e.g. communication protocols). The RU is used to acquire and manage the SmartCell's resources. The IU is used for transmission/reception of data. At runtime, a SmartCell accesses shared NetMem to store/discover/retrieve networking data concerns' data of required services and related operational parameters. The KU enables self- and situational-awareness, adaptability and evolution capabilities. Controller unit (CU) and observer unit (OU) in a SmartCell are used to autonomously manage its operation where those units possess autonomic monitoring, control and analysis abilities. For detected unexpected events such as resource shortages, the

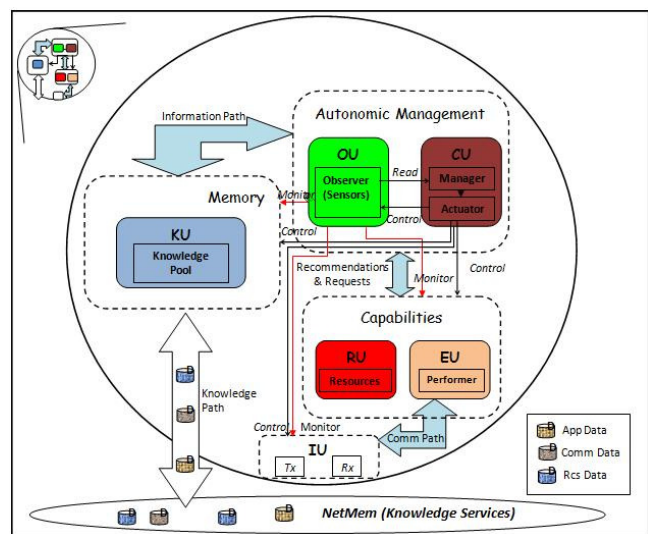


Figure 1. Internal Structure of a SmartCell

SmartCell's embedded autonomic monitoring and management facility (represented by the CU and OU) will react accordingly, for example by requesting additional storage space or asking a neighboring SmartCell to execute required services.

At runtime, NetMem is built and updated to help in improving adaptability and evolvability of SmartNet by having accessible records on demand of networking concerns' data related to services and resources supported by communicating SmartCells. Each recorded and supported service in NetMem will have specific application, communication, and resource attributes. Decisions taken in managing the internal operation of functionalities by each SmartCell will depend on executing operations based on the knowledge gained from the internal memory and/or NetMem. This mechanism can enhance the overall operation performance by observing at runtime any changes (e.g. high rate of resource consumption) and by utilizing information gathered before in local memory and/or NetMem (e.g. knowing an optimized transmission value for an application).

We define a concern-based communication protocol (CCP) for internal/external communication of SmartCells. Regions, we call CellDomain, may be dynamically formed for groups of SmartCells which support specific services and/or have similar resource capabilities. Each CellDomain will have its own management policies, identifiers, internal addresses and authentication mechanisms which will be used for its member SmartCells. CellDomains facilitate discovering of required resources and services. SmartCells in different CellDomains can share services and resources via NetMem. The clustering mechanism for the CellDomains and their hierarchical recursive construction will facilitate management of SmartCells and their exchange of services and resources.

The following subsections describe the knowledge management capability in SmartNet based on a shared constructed NetMem, operation resilience in SmartNet, a preliminary data representation model which is used for the shared NetMem, and a simple networking scenario deployed by SmartNet, respectively.

A. SmartNet Knowledge Management

SmartCells' local memory is incorporated to enhance self- and situational-awareness and help in autonomously managing their operation. Additionally, each SmartCell has the ability to store in its local memory its collected networking data which include supported services, operational parameters, communication protocols, and available resources and their specifications. SmartNet's NetMem maintains knowledge about networking relevant data, can be shared by communicating SmartCells at runtime and on demand. The SmartCells' local memory and NetMem enable autonomic knowledge-based management (KBM) in SmartNet, for example to establish communication paths amongst SmartCells based on required services' concerns. Via NetMem, the KBM can get information to identify regions (i.e. CellDomains) where services of interest are supported. SmartCells use simple and systematic methods to register, discover, and retrieve at runtime and on demand self- and shared knowledge based on analysis and observation of events related to services of interest and external environment conditions. In addition, the KBM can aid SmartCells in supporting unknown services that have not yet established use records. The KBM can predict their use pattern by matching their requirements with those of known services and related concerns' data in NetMem. For instance, a service is required by a SmartCell where that service has not been supported before. In addition, no previous history exists for that service in NetMem. That service has certain requirements for networking concerns (e.g. TCP and AODV protocols as communication concerns) and operational parameters (e.g. transmission power ≥ 20 dBm and noise level < 0.3 dBm). In NetMem, SmartCell's KBM facility can look for registered services which have similar requirements (i.e. networking concerns' data) or match most of the required service's demands. Also, KBM can determine CellDomains which can support that service. Hence, the KBM can acquire information (i.e. concerns and locations of support) which aid in supporting services with no previous history and experience.

1) NetMem construction:

We construct a NetMem, one logical distributed KP, by which SmartCells can share their knowledge, modeled information based on networking relevant data, related to three networking concerns: application, communication and resource. We are inspired in design of NetMem by functionalities of human memory [26] to target capabilities for smarter networking operations. Hence, for NetMem's design, we consider data virtualization techniques (e.g. in [27]), artificial intelligence (AI) mechanisms [28], machine learning (ML) algorithms [29], and function-behavior-structural (FBS) engineering framework [30,31] to provide capabilities of storing/retrieving massive networking data from heterogeneous data sources with different levels of details. NetMem consists of dataset tables which clarify networking data such as provided services, their related concerns' data and operational parameters. Those dataset tables comprise two types of memory in NetMem, short-term memory (StM) and long-term memory (LtM). StM is a working memory based on networking data collected from communicating entities. LtM is a memory of conceptual/valuable networking data relied on reserved data at StM and acquired networking data from experts. There will be reasoning algorithms depending on AI

and ML algorithms for generating/storing/recalling data to/from LtM's tables. NetMem's design and evaluation are part of our ongoing work as we will present it in details in our next paper.

Shared knowledge in NetMem will be kept in a way that facilitates discovery of required networking data such as identifying service requirements and which CellDomain can support those services. In addition, maintained data in NetMem shows date/time/location of offered services. SmartCells register their supported services in NetMem after they finish the operation process for those services. In another meaning, each time a SmartCell executes a service; it will update NetMem by a new service entry in NetMem's dataset tables. Moreover, SmartCells can get information about required services in NetMem by searching in entries of dataset tables. SmartCells can look for data in the NetMem by using data virtualization techniques and multiple search keys such as service identifiers, certain services' concerns, and date/time of tables' entries.

To realize NetMem, we use a cloud storage mechanism [32]. That mechanism yields big extendible tables to store a huge amount of datasets. Cloud storage provides for SmartNet an implicit storage at runtime and on demand based on QoS and resource requirements of supported services. We design extendible tables where networking relevant data (e.g. networking concerns' data) can be registered and retrieved (these tables are introduced in subsection C). Our dataset tables' design is inspired by the work in [16]. There is a capability in [16] to store large datasets in big tables based on an open source implementation technique for big tables for massive scalability defined in HBase [17] and built on top of the Java framework Hadoop [18].

We propose primarily dataset tables for our NetMem (i.e. tables in StM and LtM). For StM, we scheme four dataset tables which are namely service data structure table (SDST), concerns data structure table (CDST), concerns index table (CIT), and composite concerns index table (CCIT). SDST contains entries for services and their related networking concerns (application, communication, resource). Using FBS engineering design [30,31], CDST extracts each entry service in the SDST where it defines functional, behavioral, and structural data per each service's concern. The CIT and CCIT clarify in which entries in the SDST, interesting single concern's and group of concerns' attributes can be found respectively. For LtM in NetMem, we propose class dataset tables and appended sub-class (level-1 and level-2) dataset tables. LtM's tables are relational tables where they maintain conceptual networking data, classified based on three networking concerns and FBS engineering framework.

Some previous works have implemented mechanisms for storing Internet data and measurement data based on required attributes. For example, CAIDA in [19] presented the Internet measurement data catalogue (IMDC) as meta-data repositories of measurement data to achieve smooth accessibility of that data for comparative analysis purposes. IMDC provided detailed information about stored data such as its source and location and time of its occurrence. Authors in [20] proposed a scalable Internet measurement repository (SIMR) to track

Internet measurements where large databases provide information about measurements, tools, users, experiments, and datasets. These datasets can be accessed easily to perform on them analysis and comparison studies of obtained measurements at various contexts and times. Similar to SIMR, the MOME work in [21] presented a web-based way for repositories of meta-databases independent of services' types. Authors of MOME showed the capability to record information related to measurement tools and measurement data (as routing data and QoS attributes) where there is a public access capability, but processes of updating/retrieving entries are limited to registered users.

Our SmartNet knowledge cloud is different from previous work where networking concerns' data related to multiple services are stored. That data can be accessed autonomously by communicating entities (i.e. SmartCells) at runtime and on demand to establish communication paths according to required services and to mitigate any abrupt events as resource shortages. In other words, SmartCells can retrieve services' networking concerns whether they have previous experience about required services or not. Moreover, SmartCells can know at which CellDomains (i.e. locations) and when (i.e. date and time) required services were previously supported. Thus, SmartCells can have up-to-date information about required services and related concerns.

B. Resilience

Beside our unique SmartNet's KBM, SmartNet's autonomic resource management intrinsically embedded in SmartCells enhances network resilience. The intrinsic monitoring facility (via the OU) in SmartCells can observe internal states and resources besides observing external context (i.e. networking environment). OU can detect abrupt and unexpected events (e.g. support unfamiliar services and resource shortages). Then, the autonomic management facility (via the CU) can react by requesting information from local memory and/or SmartNet's NetMem and additional resources or asking a neighboring networking entity to execute a required service. SmartCells can request at runtime and on demand resources and/or services from other SmartCells for support. SmartCells can get information from their local memory about previously supported services and available resources. In addition, SmartCells can acquire knowledge from the NetMem concerning running services and resource capabilities by other SmartCells. SmartNet's NetMem is built and updated at runtime to provide accessible information and to help SmartCells in improving QoS of supported services, mitigating runtime abrupt events, and meeting demands of unfamiliar services which might be supported and recorded by other SmartCells.

C. NetMem data representation model

For large and multiple datasets' tables of StM and LtM in SmartNet's NetMem, SmartNet adopts systematic and simple ways to represent networking datasets (i.e. networking concerns' data of services) in the shared NetMem. SmartCells can use those ways easily to for example match required networking concerns of a service with others found in datasets. Set theory [22] can provide a preliminary solution for data

representation for SmartNet. Data representation model will be used by SmartCells to help them in accessing needed information from datasets. For instance, SmartCells can know where (from which CellDomain and which entry in the dataset table) they can get their requests due to certain services' requirements. Our NetMem data representation model supports string, integer, double, and vector values. That model provides various sizes of data tuples which have attribute/value pairs.

We use the following notation to represent networking concerns' data and entries in the elementary proposed dataset tables in StM:

- $x_{a,i}$: the application concern's attributes of service i. (e.g. $x_{a,i} = \{\{\text{FileTransfer}\}, \{\text{PktSize_1024B}\}\}$).
- $x_{a,FBS,i}$: FBS data of application concerns per service i. (e.g. functional data: file transfer, behavioral data: expected data rate at congested environment, structural data: data packet format).
- $x_{c,i}$: the communication concern's attributes of service i. (e.g. $x_{c,i} = \{\{\text{AODV}\}, \{\text{TCP}\}\}$)
- $x_{c,FBS,i}$: FBS data of communication concerns per service i. (e.g. functional data: routing protocol, behavioral data: expected maximum delay at congested environment, structural data: routing header format).
- $x_{r,i}$: the resource concern's attributes of service i. (e.g. $x_{r,i} = \{\{\text{Tx_20dbm}\}, \{\text{StorageSpace_20GB}\}\}$)
- $x_{r,FBS,i}$: FBS data of resource concerns per service i. (e.g. functional data: sensing movement vibrations, behavioral data: expected data arrival rate, structural data: technical specifications for transmission/reception).
- S_i : the set of concerns for a service i from a group of n known services. (e.g. $S_i = \{\{x_{a,i}\}, \{x_{c,i}\}, \{x_{r,i}\}\}$)
- S: the set of all possible services defined in the SDST table. This set will have n elements (i.e. services).
- ID_i : the identifier of service i. (e.g. $ID_i = \{\text{srv_2}\}$)
- M_i : the concerns pattern of service i that how many defined attributes for application, communication, resources concerns service i has. (e.g. $M_i = \{5,3,6\}$ and this means that there are 5 application, 3 communication and 6 resource' attributes respectively).
- $F(x)$: a match function which returns identifier of services have similar concern's attributes x of a service. This function is executed over SDST/CDST tables. This function can be used to build the semantics graph (i.e. it clarifies concerns matching among known services) and also in a case of no previous history about service i. If there is no matching, the output will be $\{\phi\}$.

- $F(X,x)$: a match function which is executed over the CIT table to compare a required networking concern's attributes X with another one x registered in the CIT table. It returns the entry number in both SDST and CDST tables. (e.g. $F(X,x) = (X \cap x)$). The output for that function is true in case of matching or false (i.e. $\{\phi\}$) otherwise.
- $F(X_1 \dots X_N, x_1 \dots x_N)$: a match function which is executed over the CCIT table to compare N multiple required networking concerns' attributes $\{X_1\}, \{X_2\}, \dots, \{X_N\}$ with others $\{x_1\}, \{x_2\}, \dots, \{x_N\}$ stored in the CCIT table. It returns the entry number in both SDST and CDST tables. (e.g. $F(X_1 \dots X_N, x_1 \dots x_N) = (X_1 \cap x_1), (X_2 \cap x_2), \dots, (X_N \cap x_N)$). The output of the function is true or false (i.e. $\{\phi\}$) in each concern's section.
- $E_{SDST,k}$: the k^{th} entry in the SDST table for a service i . e.g. $E_{SDST,k} = \{\{\text{date}\}, \{\text{time}\}, \{\text{src_CellDomain_ID}\}, \{\text{dest_CellDomain_ID}\}\} \cup \{\text{ID}_i\} \cup \{\text{S}_i\} \cup \{\text{x}_{a,i}\} \cup \{\text{x}_{c,i}\} \cup \{\text{x}_{r,i}\} \cup \{F(\text{x}_{a,i}), F(\text{x}_{c,i}), F(\text{x}_{r,i})\}$.
- $E_{CDST,k}$: the k^{th} entry in the CDST table for a service i which is related with the k^{th} entry in the SDST table. e.g. $E_{CDST,k} = \{\{\text{date}\}, \{\text{time}\}, \{\text{src_CellDomain_ID}\}, \{\text{dest_CellDomain_ID}\}\} \cup \{\text{x}_{a,FBS,i}\} \cup \{\text{x}_{c,FBS,i}\} \cup \{\text{x}_{r,FBS,i}\}$.
- $E_{CIT,k}$: the k^{th} entry in the CIT table for a service i which is related to an entry in the SDST table. (e.g. $E_{CIT,k} = \{\text{x}_{a,i}\} \cup \{\text{x}_{c,i}\} \cup \{\text{x}_{r,i}\}$).
- $E_{CCIT,k}$: the k^{th} entry in the CCIT table for a service i which is related to an entry in the SDST table. (e.g. $E_{CCIT,k} = \{\text{x}_{a,i}\} \cup \{\text{x}_{c,i}\} \cup \{\text{x}_{r,i}\}$).

For any unfamiliar service (i.e. outside the set “S”), if a SmartCell knows some/all networking concerns of that service, the SmartCell can begin to compare concerns and attributes of that service with other known services using match functions as $F(X,x)$ and/or $F(X_1 \dots X_N, x_1 \dots x_N)$. By that way, a SmartCell can get information about which entries in the SDST table contain the required service. Then, it can know CellDomain IDs where the required service can be supported. Each known service i will have a concerns' pattern M_i that each SmartCell can know how many attributes per concern it should extract.

We have the following notation for networking concerns' data and related entries of the proposed big tables in LTM memory:

- $A = \{A_1, A_2, A_3, \dots, A_m\}$; the itemset of the application concern which are found in sets of class table of that concern. Each item in the set represents an application concern (e.g. $A_1 = \text{FileTransfer}$).
- $C = \{C_1, C_2, C_3, \dots, C_n\}$; the itemset of the communication concern which are found in sets of class table of that concern. Each item in the set represents a communication concern (e.g. $C_1 = \text{TCP transport protocol}$).

- $R = \{R_1, R_2, R_3, \dots, R_q\}$; the itemset of the resource concern which are found in sets of class table of that concern. Each item in the set represents a resource concern (e.g. $R_1 = \text{wireless network interface card}$).
- $S = \{a, c, r\}$ where $a \subseteq A, c \subseteq C, r \subseteq R$; the service itemset which comprises itemsets of application, communication, and resource concerns and their related items.
- Concern item; it is a networking concern which may be application or communication or resource concern; and it relates to a specific networking operation and/or flow.
- Table ID; table identifier which entitles class and sub-class tables to make differentiation between them in their sets.
- Concern ID; the binary code which differentiate the various networking concerns where application concern ID = $\{01\}$, communication concern ID = $\{10\}$, resource concern ID = $\{11\}$;
- CcnCode = $\{\text{ConcernID}, \text{TableID}, \text{ColumnNumber}\}$; the numeric code set to represent entries of sub-class tables.
- $F_{\text{ConcernItem}}$; data which represents functions of a concern item. It is an entry in sub-class level-2 tables.
- $B_{\text{ConcernItem}}$; data which represents behavior (e.g. measurements) of a concern item. It is an entry in sub-class level-2 tables.
- $S_{\text{ConcernItem}}$; data which represents structure (e.g. syntax and code) of a concern item. It is an entry in sub-class level-2 tables.
- $E_{\text{Class, ID, k}}$ = the k^{th} entry in the class table of identifier ID. e.g. $E_{\text{Class, ID, k}} = \{\text{concernItem}_1, \text{concernItem}_2, \dots, \text{concernItem}_n\}$.
- $E_{\text{SubClass}_1, \text{ConcernID, ID, k}}$ = the k^{th} entry in the sub-class level-1 table of identifier ID for ConcernID. e.g. $E_{\text{SubClass}_1, \text{ConcernID, ID, k}} = \{\text{CcnCode}_1, \text{CcnCode}_2, \dots, \text{CcnCode}_m\}$.
- $E_{\text{SubClass}_2, \text{ConcernID, ID, k}}$ = the k^{th} entry in the sub-class level-2 table of identifier ID for ConcernID. e.g. $E_{\text{SubClass}_2, \text{ConcernID, ID, k}} = \{F_{\text{ConcernItem}}, B_{\text{ConcernItem}}, S_{\text{ConcernItem}}\}$.

D. Running of Networking Functionalities by SmartNet

Fig. 2 shows simple example for a networking operation by SmartNet to support a service X with specific networking concerns. Two different authorized SmartCells in different CellDomains establish a communication path for performing the required service X (e.g. file transfer). The connection is done throughout intermediate SmartCells in same/other CellDomains. These SmartCells will be involved in the established path depending on their capabilities to fulfill requirements of that service where its concerns can be retrieved from NetMem. The following steps reveal how the communication channel is initiated by the source SmartCell (SC):

1. Authentication and check for rights of service access are performed within related CellDomains. (each SC

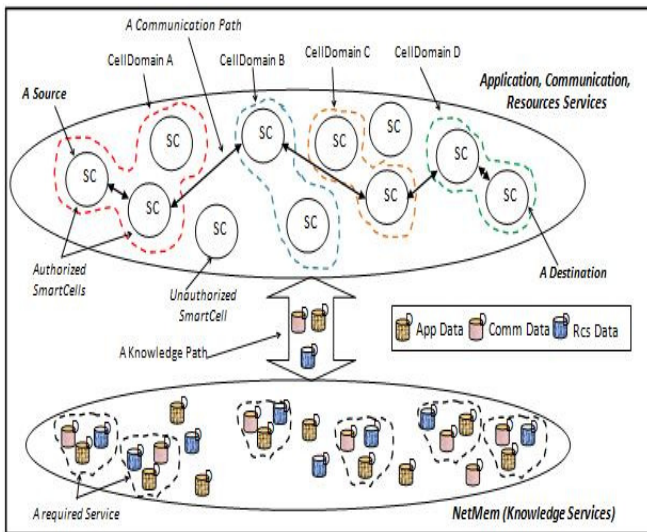


Figure 2. Knowledge-based Management by SmartNet

- has to be belonged to a CellDomain).
2. Each SC gets an internal port ID (dynamically assigned by the local CU) and gets an internal address (known only within the CellDomain).
3. Throughout self-knowledge and/or shared NetMem, the required service name (i.e. X) is determined beside its associated networking concerns' aspects.
4. Source SC will check for available internal resources to meet required QoS demands.
5. Source SC sends a request to the destination SC to serve and operate the required service.
6. Destination SC will check for applicability of serving that application based on local knowledge and resources. It may ask for support from neighboring SCs within their related CellDomains.
7. Once the requester SC gets positive feedback response, it will allocate sufficient resources and it begins to operate soft-state data transfer protocols for sending the required file.
8. Relaying and multiplexing services can be used if the requester SC can not catch the destination SC directly. As depicted in Fig. 2, two SCs in CellDomain B and C support the communication path.
9. Inside each SC, the intrinsic monitoring capability is run where internal states, resources, and other interesting external phenomena and neighboring SCs' behavior can be observed.
10. SCs involved in the communication path will recur to its knowledge system to implement any experience they have before which it may enhance the operation.
11. Requests for additional resources and/or support for the required service can be sent to neighboring SCs via using the CCP protocol or accessing NetMem (plane of knowledge services).
12. Once the service is accomplished, the connection is terminated (loosely-coupled mechanism) and resources are de-allocated and reports of interesting

parameters as packet loss and throughput are generated. The local memory will be updated with information about neighboring SCs, supported services, and interesting and measured QoS metrics.

IV. EVALUATION

We used JSim [23, 24]; a discrete time – event component-based simulator based on autonomous component architecture. The behavior of JSim components are defined in terms of contracts. Those components can be plugged to the JSim during the execution process. We refer the reader to [24] for more documentation on JSim. We implemented a networking scenario for a simple sensor network with six sensor nodes to detect signals generated from four target nodes. A sink node gathers readings from the sensor nodes according to their detection. We applied that layout for three cases. First two cases are networking operation without and with database capability (i.e. shared NetMem) respectively. The third case is with a shared database and suppression effect by the sink for no-interest sensors. Table I shows simulation parameters. In cases of a shared database, the sink and sensors can access NetMem to store their services with date/time stamps and retrieve any required information at runtime and on demand.

At bootstrapping phase, sensors register themselves in CellDomains according to services they support. Also, sensors provide performance and QoS related information as which target nodes they support and with what (signal-to-noise ratios) SNRs they detect signals. The sink can decide from which sensor it will receive sensor packets. The decision making by the sink node may depend on defined selection criteria as target nodes' identifiers and SNR values of detected signals. In addition, the sink can request some sensors that it is not interested in their packets to stop sending more packets. Hence, the sink could minimize its resource consumption and decrease the noise and contention levels in the intermediate wireless channels. We addressed the average collected SNR by the sink, the number of processed packets by the sink, average processing time at the sink, and average throughput and queue

TABLE I
SIMULATION PARAMETERS

	Parameter	Value
Network Parameters	No. of Nodes	6 sensors, 4 targets, one sink
	Simulation Area	600×500 m ²
Communication Parameters	Target (BcastRate, SampleRate)	(a signal every 20 sec, 1.0Hz)
	Propagation Model	Free Space Model
Communication Parameters	Transmission Range	250 m
	Sending Rate (av. Arrival rate)	48 Mbps
	Routing Protocol	AODV
	MAC Protocol	IEEE 802.11
System Parameters	Random Number Generator	Gaussian Distribution (random seeds)
	Mean	0.1
	Standard Deviation	0.5
	Traffic Model	Poisson
	Packet Size	512 Bytes
	Queue Type	FIFO
System Parameters	Queue Capacity	40 packets
	Simulation Time	550 seconds

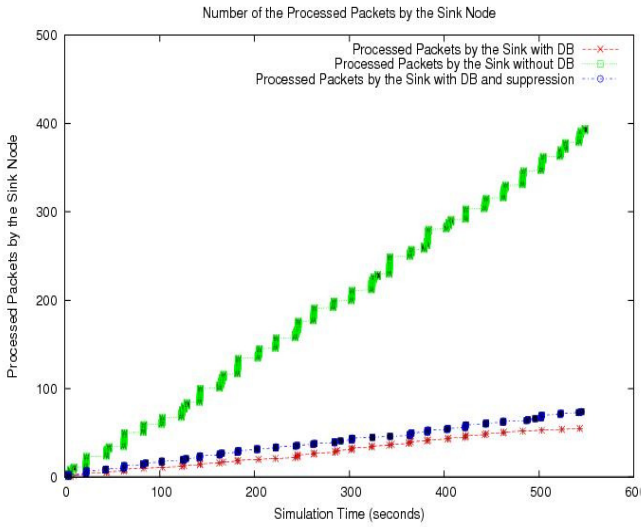


Figure 3. Packets processed by the sink node

size of the sink. Our preliminary results are shown in Fig. 3, Fig. 4, Fig. 5, Fig. 6 and Fig. 7 which reflect in approximately all cases better values in case of operation with shared database (i.e. NetMem) and suppression. Obtained results within the same scenario are calculated with confidence interval 95% using t-distribution.

Fig. 3 shows the number of packets processed by the sink. In the case of no database, there are more packets due to replication of packets sent from all sensors upon detection. Also, the case of database with suppression generated more packets than the case of database and this is because packets from no-interest sensors are stopped. Hence, lower contention level can be obtained at the sink and the packet loss will be decreased. Fig. 4 shows that the sink could collect, in cases with database, higher average SNR values than in the case without database. These enhanced results are due to the ability of the sink to know from the shared NetMem which sensors

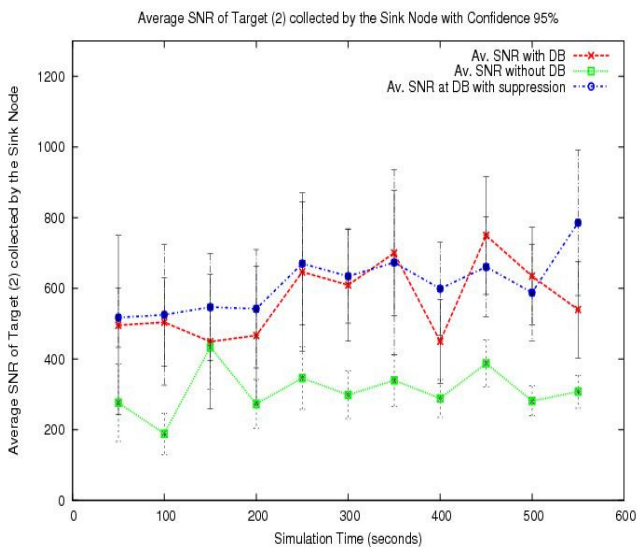


Figure 4. Average SNR Collected by the sink node from target 2

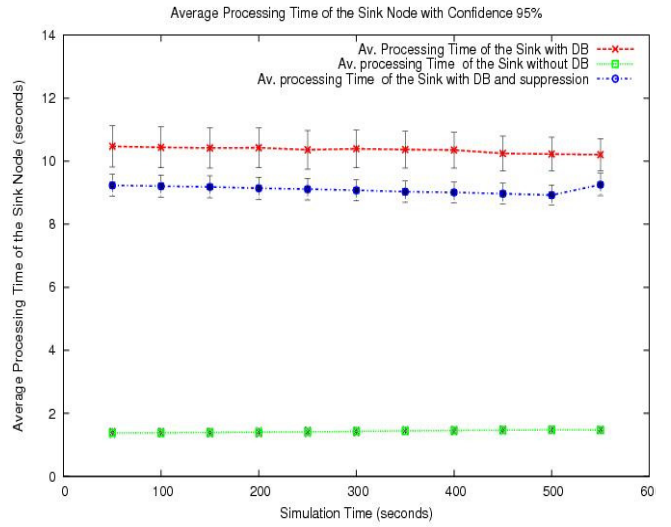


Figure 5. Average Processing Time by the sink node

can detect signals with higher SNRs from target nodes.

According to the runtime access of the shared NetMem by the sink to investigate which sensor nodes efficiently detect signals (i.e. higher SNR) from targets, the cases with database have long processing time as shown in Fig. 5. The case with suppression enhances the result compared with the case with database as the number of packets received and processed by the sink decreases. Fig. 6 and Fig. 7 reveal the average queue size and throughput of the sink node respectively. The obtained results in cases of with/without database are approximately the same as the sink node actually receives same number of packets in both cases but it processes packets from sensors of interest only in the case with database. But, when the sink requests no-interest sensors not to send more data packets, it could achieve better throughput and it made best use of its queue where the queue size has approximately one packet along the simulation time. The results show a conclusion that the efficiency is improved in cases with database. Moreover,

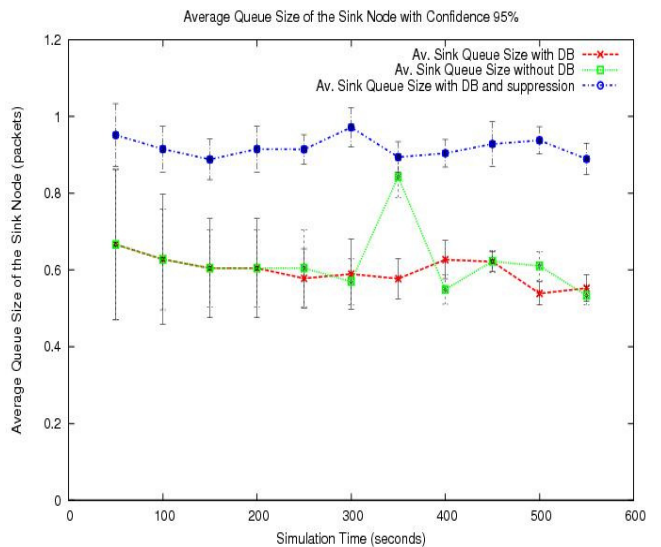


Figure 6. Average queue size of the sink node

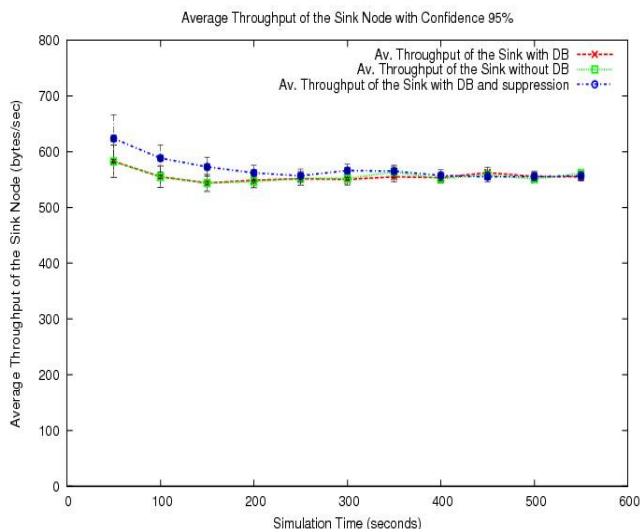


Figure 7. Average throughput of the sink node

there is capability of scalable operation where sensors, which can support different services, join the network at runtime once they register themselves to a CellDomain according to their provided services.

V. CONCLUSION

The current Internet and many of the recently proposed network architectures did not consider network memory or explicit autonomic resource management to meet the requirements of the ever-growing number of highly dynamic services with widely varying emergent QoS and resource demands. In this paper, we proposed SmartNet, a self-managing network architecture that is scalable, adaptable, resilient, and evolvable. We synthesized SmartCells, SmartNet's basic building blocks, as composable autonomous components with intrinsic monitoring, analysis, and control capabilities. Each SmartCell is endowed with internal memory to support self and situational awareness of internal/external contexts and supported services. For smarter networking operation and management, we endowed SmartNet with "memory" realized as a distributed knowledge plane. Our proposed NetMem is inspired by human memory and can be accessed at runtime and on demand by communicating entities to store/discover/retrieve networking data relevant to the three networking concerns: application, communication and resource. Hence, the shared NetMem aids in enhancing networking operation resilience and adaptation by better informing network operations. Preliminary simulation results showed an improvement in the case of NetMem-enabled operations than the contemporary network architecture without memory. The obtained results revealed improvements in the average number of processed packets, collected SNR, throughput and queue size.

Our ongoing work explores: 1) NetMem's design and evolution capabilities; 2) Efficient lightweight methods for storing and retrieving information, and extracting and managing knowledge in the shared NetMem; 3) Predictive networking protocols to support unfamiliar services that have not been served previously; and 4) Effective on-line query and

support techniques to autonomically resolve resource and capability shortages that might occur at some communicating entities.

REFERENCES

- [1] A. Feldmann, "Internet clean-slate design: what and why?" *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 3, pp. 59–64, 2007.
- [2] H. Hassan, M. Eltoweissy, and M. Youssef, "Cellnet: A bottom up approach to network design," in *New Technologies, Mobility and Security (NTMS), 2009 3rd International Conference on*, dec. 2009, pp. 1–6.
- [3] G. Bouabene, C. Jelger, C. Tschudin, S. Schmid, A. Keller, and M. May, "The autonomic network architecture (ana)," *Selected Areas in Communications, IEEE Journal on*, vol. 28, no. 1, pp. 4–14, january 2010.
- [4] A. Zafeiropoulos, A. Liakopoulos, A. Davy, and R. Chaparadza, "Monitoring within an autonomic network: a GANA based network monitoring framework," in *Proceedings of the 2009 international conference on Service-oriented computing*, ser. *ICSO/ServiceWave'09*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 303–313.
- [5] J. Day, I. Matta, and K. Mattar, "Networking is ipc: A guiding principle to a better internet," in *Proceedings of ReArch08 - Re-Architecting the Internet*, 2008.
- [6] J. Roberts, "The clean-slate approach to future Internet design: a survey of research initiatives," *Annals of Telecommunications*, vol. 64, no. 5, pp. 271–276, Jun. 2009.
- [7] F. Teraoka, "Redesigning layered network architecture for next generation networks," in *GLOBECOM Workshops, 2009 IEEE*, 30 2009-dec. 4 2009, pp. 1–6.
- [8] X. Sanchez-Loro, J. Casademont, J. Paradells, J. Ferrer, and A. Vidal, "Proposal of a clean slate network architecture for ubiquitous services provisioning," in *Future Information Networks, 2009. ICFIN 2009. First International Conference on*, oct. 2009, pp. 54–60.
- [9] A. Galis, H. Abramowicz, M. Brunner, D. Raz, P. Chemouil, J. Butler, C. Polychronopoulos, S. Clayman, H. de Meer, T. Coupaye, A. Pras, K. Sabnani, P. Massonet, and S. Naqvi, "Management and service-aware networking architectures (mana) for future internet -: position paper: System functions, capabilities and requirements," in *Communications and Networking in China, 2009. ChinaCOM 2009. Fourth International Conference on*, aug. 2009, pp. 1–13.
- [10] C. Henke, C. Schmolli, T. Zseby, A. A. Siddiqui, P. Müller, T. Magedanz, and E. Rathgeb, "Scenarios for a future internet based on crosslayer functional composition," in *5th GI/ITG KuVS Fachgespräch Future Internet*, Stuttgart, May-2010.
- [11] K. R. Sollins, "An Architecture for Network Management", *ReArch'09*, December 1, 2009, Rome, Italy.
- [12] D. Clark, C. Partridge, J. Ramming and J. Wroclawski, "A Knowledge Plane for the Internet", *SIGCOMM'03*, August 25–29, 2003, Karlsruhe, Germany.
- [13] D. Macedo, A. dos Santos, G. Pujolle, J.M.S. Nogueira, "MANKOP: A Knowledge Plane for wireless ad hoc networks," *Network Operations and Management Symposium, 2008. NOMS 2008. IEEE*, vol., no., pp.706-709, 7-11 April 2008.
- [14] R. Riggio, F. De Pellegrini, D. Miorandi, and I. Chlamtac, "A knowledge plane for wireless mesh networks," *Ad Hoc & Sensor Wireless Networks*, vol. 5, no. 3-4, pp. 293-311, 2008.
- [15] A. Shieh, E. G. Sirer, and F. B. Schneider, "NetQuery: a knowledge plane for reasoning about network properties," In *Proceedings of the ACM CoNEXT Student Workshop (CoNEXT '10 Student Workshop)*. ACM, New York, NY, USA, , Article 23 , 2 pages, 2010.
- [16] S. Rozsnyai, A. Slominski, and Y. Doganata, "Large-scale distributed storage system for business provenance," in *IBM Research Report, RC25154*, April-2011.
- [17] "Apache hbase." [Online]. Available: <http://hadoop.apache.org/hbase/>
- [18] "Apache hadoop." [Online]. Available: <http://hadoop.apache.org>
- [19] "correlating heterogeneous measurement data to achieve system-level analysis of internet traffic trends," in *CAIDA Project*. [Online]. Available: <http://www.caida.org/projects/trends/>
- [20] M. Allman, E. Blanton, and W. M. Eddy, "A scalable system for sharing internet measurements," in *In Proceedings of Passive and Active Measurement (PAM)*, 2002, pp. 189–191.

- [21] P. A. Ar, A. Gutierrez, A. Balanza, M. Dabrowski, B. Kaskina, J. Quittek, F. Strohmeier, A. Vidacs, K. S, and O. Zsolt, "An advanced measurement meta-repository," March-2005.
- [22] R.G. Clegg, "A set theoretic framework for enumerating matches in surveys and its application to reducing inaccuracies in vehicle roadside surveys", Paper submitted to the 9th Meeting of Euro Working Group on Transportation, Bari, June 2002.
- [23] A. Sobeih, W.-P. Chen, J. Hou, L.-C. Kung, N. Li, H. Lim, H.- Y. Tyan, and H. Zhang, "J-sim: a simulation environment for wireless sensor networks," in Simulation Symposium, 2005. Proceedings. 38th Annual, april 2005, pp. 175 – 187.
- [24] JSim. [Online]. Available: <http://sites.google.com/site/jsimofficial/start-with-j-sim>.
- [25] OpneFlow [Online] Available: <http://www.openflow.org>
- [26] J. Hawkins and S. Blakeslee, On Intelligence, Times Books Henry Holt and Company, LLC, 261 pages, First Edition 2004.
- [27] Data Virtualization by Composite Software. [Online] Available: <http://www.compositesw.com/solutions/data-virtualization/>
- [28] Brunette, E.S.; Flemmer, R.C.; Flemmer, C.L.; "A review of artificial intelligence," 4th International Conference on Autonomous Robots and Agents, ICARA 2009, pp.385-392, 10-12 Feb. 2009.
- [29] M. Xue; and C. Zhu; "A Study and Application on Machine Learning of Artificial Intelligence," International Joint Conference on Artificial Intelligence, 2009, IJCAI '09, pp.272-274, 25-26 April 2009.
- [30] J.S. Gero "Design Prototypes: A Knowledge Representation Schema for Design," AI Magazine, vol. 11, pp. 26–36, Winter, 1990.
- [31] K. Dorst and P. E. Vermaas, "John Gero's Function-Behaviour-Structure model of designing: a critical analysis," Research in Engineering Design, vol. 16, pp. 17-26, November, 2005.
- [32] W. Zeng, Y. Zhao, K. Ou, and W. Song, "Research on cloud storage architecture and key technologies," In Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human (ICIS '09). ACM, New York, NY, USA, 1044-1048, 2009.