Biologically-inspired Network "Memory" for Smarter Networking

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

Mohamed Eltoweissy [1]
Department of Computer Science and Engineering,
Egypt-Japan University of Science and Technology
Alexandria, Egypt
Mohamed.eltoweissy@ejust.edu.eg

***Abstract*— Emerging technologies such as the Internet of Things generate huge amounts of network traffic and data which lead to significant challenges in a) ensuring availability of resources on-demand, b) recognizing emergent and abnormal behavior, and c) making effective decisions for efficient network operations. Network traffic data exhibit spatiotemporal patterns. Learning and maintaining the currently elusive rich *semantics* based on analyzing such patterns would help in mitigating those challenges. In this paper, we propose the concept of a network "memory" (or NetMem) to support smarter data-driven network operations as a foundational component of next generation networks. NetMem will enable networking objects to understand autonomously, at real-time, on-demand, and at low cost semantics with different levels of granularity and related to various network elements. Guided by the fact that human activities exhibit spatiotemporal data patterns; and the human memory extracts and maintains semantics to enable accordingly learning and predicting new things, we design NetMem to mimic functionalities of that memory. NetMem provides capabilities for semantics management through uniquely integrating data virtualization for homogenizing massive data originating from heterogeneous sources, cloud-like scalable storage, associative rule learning to recognize data patterns, and hidden Markov models for reasoning and extracting semantics clarifying normal/abnormal behavior. NetMem provides associative access to data patterns and relevant derived semantics to enable enhancements in early anomaly detection, more accurate behavior prediction and satisfying QoS requirements with better utilization of resources. We evaluate NetMem using simulation. Preliminary results demonstrate the positive impact of NetMem on various network management operations.**

***Keywords*—Network Semantics, Data Virtualization, Cloud Data Storage, Distributed Systems, Bio-inspired Design.**

I. INTRODUCTION

Emerging networks and technologies (e.g., Internet of Things (IoT) [1]) comprise billions and wide varieties of communicating entities, running services and applications, and types of used resources. According to that, it is expected to have huge amount of network traffic related to various sources. Such complexities pose formidable challenges for network analysts and decision making. Also, networking entities will face difficulties to keep up with the dynamic requirements and behavior of the heterogeneous network elements (e.g., applications, protocols, resources, etc.). Contemporary tools and solutions as presented in [11][12] are limited in their ability to identify dynamic behavior aspects and thus they constrain our understanding of actors and activities in the network. Moreover, the storage of traffic data for mining and analysis as presented in [8][9] would be prohibitive given the extreme volume of data and the timeliness needed in decision making.

It is widely known that network traffic data exhibit spatiotemporal patterns. Learning and maintaining semantics based on analyzing such patterns would support network core/end systems to recognize normal/abnormal behavior of diverse network elements and requirements of new emerging services besides it would aid in enhancing services' QoS, utilization of resources, and detection of anomaly. Additionally, effectively utilizing semantics would provide self and situational awareness to help improve network performance, adaptability and evolution.

Unfortunately, the current Internet [2] and other proposed network architectures in the contemporary literature, for example [3][6], in the most part, do not provide effective and efficient methodology for networking entities to discover, learn, store and utilize patterns of traffic behavior, particularly at runtime in an automated manner and over a long period of time. For example, the behavior of TCP, a reliable communication protocol, can be extracted by learning patterns of that protocol. Learned TCP patterns would lead to know normal range of port numbers used between entities, normal ratio of source/destination messages through specific time period, and values of sequence numbers in source/destination within N connections. Hence, the normal behavior of TCP protocol can be expected based on discovered features in its patterns. Additionally, up to our knowledge, hardware and software solutions proposed for enhancing network intelligence (e.g., cognitive networks [27]), did not provide systematic means to learn, store and associate network semantics that can aid in extracting information concerning diverse network elements and their normal/emergent behavior. We depict the limited utilization of traffic semantics in networking operations as the "networking semantics gap". Such gap deprives networks of efficient use of information at different levels of granularity, which would otherwise help in enhancing their operation.

In this paper, we motivate the concept of a "memory" for resolving the aforementioned semantics gap for smarter networking operation. Our proposed network memory, termed NetMem, is a shared distributed semantics management system. It provides a facility for learning at runtime spatiotemporal patterns exhibited by syntax data (SData) originating from heterogeneous networking domains and related to various network elements. By recognizing patterns, NetMem learns and discovers network semantics in a

[1] Also affiliated with the ECE Department at Virginia Tech and University of Arizona.

COLLABORATECOM 2012, October 14-17, Pittsburgh, United States

DOI 10.4108/icst.collaboratecom.2012.250508

systematic way to inform networking operations with semantics at different levels of abstraction (e.g., function and normal behavior of routing protocol and reliable service in a specific contexts such as wireless ad-hoc networks).

NetMem is inspired by the functionalities of the "human memory". The human memory [18] is capable of autonomously collecting huge amounts of data with different levels of details via our five senses, i.e., various sources, and learns their pattern and derives associative semantics accordingly. There are unified semantics representation and a scalable structured way for yielding associative semantics storage. Moreover, semantics in human memory are accumulated as sequences and updated continuously and can be associatively accessed and retrieved. Based on maintained semantics, humans can predict future events, learn things, and recognize new ones by matching their estimated semantics with those which are already registered. Our claim is that the functionalities of human memory [18] are suited to designing NetMem because of analogy in processes of deriving/matching semantics based on learning patterns and capabilities which are offered in human memory for the associative retrieval and scalable storage.

NetMem structure adopts composable and cooperative building components forming connection patterns for data feedforward and semantics feedback. NetMem has memory components which are implemented using a cloud-like data storage [27] form - big relational tables [7]. NetMem comprises the following components:

a) Cloud data-like storage [27] components to get short-term memory (StM) and long-term memory (LtM) manifesting associative storage concept and mimicking the hierarchal memory system of classified and sequenced patterns in human brain [18];
b) Data collection and acquisition component, we call DVA, which mimics sensory memory system in human [18]. DVA includes data virtualization techniques for collecting data from various sources and unifying data representation in StM as data profiles;
c) Semantic manager (SM) which mimics neocortex functionality and it is responsible for learning and deriving semantics in LtM. SM adopts associative rule learning algorithms for pattern learning and feature extraction, Fuzzy membership functions [22] for classifying features and hidden Markov model (HMM) [25] for reasoning and semantics extraction. Feature extraction and classification processes facilitate semantics discovery operations because those processes reduce dimensionality of data, i.e., show discriminative information; and
d) Controller and interface component which represents the capability of feedforward and feedback connectivity among the above mentioned components besides allowing data/actions/alerts pass to requesting entities outside the NetMem system.

Our contribution in the paper is two folds:

- A biologically-inspired architecture and methodology for a network memory mimicking functionalities of the human memory for resolving the networking semantics gap; and
- Constructing a network semantics management system with associative semantics storage/retrieval/matching capabilities at runtime and on-demand for supporting enhancement of running services' QoS guaranties and anomaly detection.

The remainder of this paper is organized as follows. Section II presents NetMem design showing its bio-inspiration of human memory functionalities. Section III discusses related work. Section IV discusses evaluation and the obtained results. The paper concludes in section V with an outline of future work.

II. NetMem System

NetMem is a shared distributed system that can be built separately on multiple autonomous entities with capability of inter-communication and semantics integration. Also, NetMem can be attached with already shared, existing and interconnected networking entities in the current Internet (i.e., overlay networks). NetMem outputs semantics with lower levels of details based on monitoring and learning spatiotemporal patterns of huge amount of high dimensional network data, which possess higher levels of details. NetMem provides capabilities for networking entities to store/discover/retrieve at runtime and on-demand SData/semantics related to different network elements. NetMem targets minimizing resource consumption at entities and enhancing scalability in data and algorithms by limiting their storage at enormous entities to perform NetMem functionalities. Maintained data (i.e., SData and semantics) at NetMem are represented uniformly, associated in big relational tables [7], and classified into three networking concerns, namely application, communication and resource concern as those concerns are defined in [6]. At the same time, data are abstracted via FBS engineering framework [20] to functional, behavioral, and structural (FBS) aspects.

The following subsections describe concepts underlying our NetMem system, its architecture and operations, respectively.

A. Concepts

To achieve our goal, we propose that our system is driven by group of fundamental concepts, input to the system; to get system features at the output, see Fig. 1. NetMem system design and operation depends on the following concepts:

1) Formal reasoning concept states a well-founded artificial intelligence functionality based on integrated mathematical reasoning algorithms to perform data patterns learning, semantics extraction and representation, and semantics matching algorithms;
2) Data-driven conceptualization concept declares that

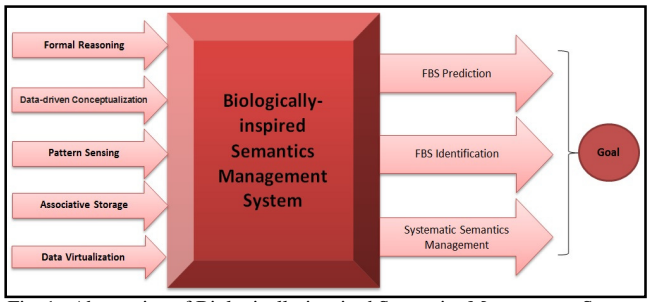


Fig. 1. Abstraction of Biologically-inspired Semantics Management System

semantics learning and deriving algorithms are based on extracted and classified features of learned data patterns;

3) Pattern sensing concept states the ability to discover patterns based on their extracted and classified features;
4) Associative storage concept shows that storage would be enabled with capabilities of identifying storage locations by their contents or part of contents and having hierarchical storage structure by showing interdependence between different storage locations based on some criteria as imply or dependence; and
5) Data virtualization (DV) concept states that massive data from heterogeneous sources can be abstracted/federated/ presented to one logical place without physical data movement, e.g., [4][5]. DV provides for our system built-in tools for data homogenization and, also, data models for unifying data representation.

NetMem system has the following features as shown in Fig. 1, i.e., output of our system:

1) *FBS Prediction:* the capability of system to conjecture semantics of new/abnormal various network elements (e.g., services, applications, protocols, etc.) on different levels of abstraction which are FBS aspects. FBS prediction is performed throughout learning patterns, features extraction and classification mechanisms besides applying implemented reasoning algorithms and concept models;
2) *FBS Identification:* the ability of the system to learn patterns of data related to communicating entities and running services and applications in complex systems to identify semantics on different levels of abstraction in case of normal operation modes or emergent behavior due to changes in internal/external contexts; and
3) *Systematic Semantics Management:* semantics are maintained and organized in a structured way, using concept models and definition language besides showing different levels of details as FBS aspects. Semantics are stored as sequences of concepts which reveal relationships among concepts.

B. Architecture

NetMem architecture comprises the following components which can inter-communicate as shown in Fig. 2.

- Data virtualization and access (DVA): DVA is a data collection and acquisition component inspired by sensory memory system in human. DVA is attached to a sensory system to gather data from entities, channels, etc.; and it implements DV techniques [4, 5] for data homogenization; and it possess a data model for building syntactic data structure and uniform representation in NetMem tables.

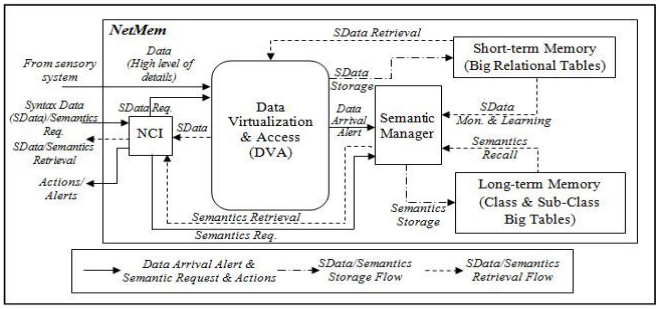


Fig. 2. Structure of NetMem

- Short-term memory (StM) and long-term memory (LtM): StM and LtM consist of sets of big extensible relational data tables. We are inspired in their design by the work in [7] where there is a capability to store large datasets in big tables based on an open source implementation technique for big tables for massive scalability defined in HBase [16] and built on top of the Java framework Hadoop [17]. StM mimics lower cortex areas in human brain which deal with spatiotime-varying data. LtM mimics higher cortex areas which deal with semantics. StM, or working memory, maintains SData related to various network elements and phenomena such as running services, applications, resources and communication flows. LtM, auto-associative memory, maintains semantics, which are based on reasoning processes for data in StM. The reasoning processes are executed within NetMem in the fourth component we call "semantic manager" or SM.
- Semantic Manager (SM): SM mimics neocortex functionality [18]; and it is responsible for discovering/ generating/matching concepts in LtM based on monitoring and learning data patterns in StM and extracted features. In another meaning, SM relates SData of high levels of details in StM to semantics, i.e., concepts with low levels of details, in LtM. SM utilizes HMMs as semantics reasoning models for extracting semantics based on data at StM. SM uses associative rule learning (ARL) algorithms [22][23] and statistical analysis to provide capabilities of associative access and learning data patterns in NetMem big tables. Also, Fuzzy membership functions (FMF) and decision trees (DT) are used to extract and classify features thus aids in learning patterns of these data. Furthermore, SM constructs relational big tables [7] of concepts to build relations between concepts.
- NetMem controller and interface (NCI): NCI is responsible for handling syntax and semantics data requests from networking entities in various domains. In other words, NCI represents the gate for data exchange between NetMem components and entities in the networking world. It has the capability for differentiating the requests, and accordingly it sends tasks, i.e., data discovery, to DVA or SM. In addition, it receives responses from DVA and SM including required data that it will enable requesting entities to access them. It might send SM additional tasks in case of the incapability of DVA to find required data. Furthermore, NCI is responsible for invoking actions/alerts based on analytical reports sent to it from DVA and SM. Analytical reports are generated based on data requests sent to DVA and SM and already data in their accessed memories where DVA has access to an internal cache memory and StM and SM has access to LtM. NCI uses defined policies and Fuzzy rules to generate actions and alerts. For instance, NCI might generate attack alert based on a report sent to it from SM showing that current learned patterns in StM refers to an attack.

C. NetMem Operations

In the following subsections, we will discuss NetMem functionalities executed by its components.

Data Virtualization and Access

This function is accomplished in NetMem DVA where it

represents uniformly syntax data (SData) from heterogeneous sources as profiles. DVA search for keywords in collected data from disparate sources and accordingly, it forms syntactic data profiles at StM. For instance, a data profile might clarify source/destination IPs, type of service such as file transfer, and packet size. DVA registers data as profiles in StM. It uses XML as a representation language clarifying type and attributes per each data profile. According to space limitation, discussion of how data are represented and associatively accessed in/from NetMem tables is beyond our scope in this paper.

Fig. 3 describes the proposed DVA component. DVA comprises the following units: a) Manager which handles data from sensory system, and requests from entities via NCI component to discover and retrieve data from StM at real time; b) Data model registry which contains the format and keywords for data structure, i.e., profile; c) Cache which is the temporary warehouse where valuable data is stored by the manager to facilitate data access and to enhance response time; d) Matching which is attached with data homogenization and classification algorithms for probabilistic dimension reduction of collected massive high-dimensional data and categorizing data into groups based on application, communication and resource concerns; and e) Transformer which represents uniformly, using defined data models, data as profiles, which comprise three concerns, in StM structured tables.

Semantic Manager (SM)

SM is responsible for generating/storing/retrieving semantics to/from LtM based on 1) continual monitoring and learning for data patterns in StM; 2) maintained statistical prediction models; 3) acquired and learned data patterns from data experts; and 4) experience and history of SM such as defined HMM models. We are inspired in SM design, shown in Fig. 4, with functionalities of cortex and neurons in human brain [18]. SM uses associative rule learning [22][23] or ARL to recognize features of data profiles maintained in StM. Moreover, SM utilizes Fuzzy membership functions [24] to classify extracted features. Based on learned and classified features, SM runs HMM [25] as models for semantics reasoning and extraction. HMM is a categorical sequence labeling supervised/unsupervised algorithm (predicting sequences of categorical labels). Sequence labeling can be treated as a set of independent classification tasks. Based on extracted features and maintained data profiles by DVA in StM, SM runs HMM models after executing the training phase. The input to HMM models are sequence of profiles' features and the output is semantics (i.e., sequences of

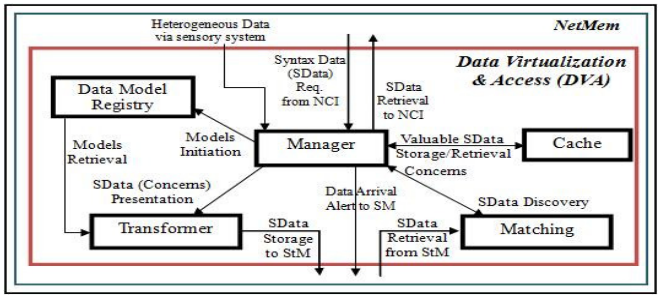


Fig. 3. Data Virtualization and Access in NetMem

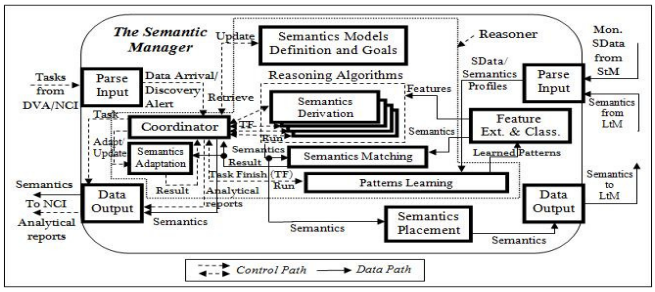


Fig. 4. Semantic Manager in NetMem

concepts) with certain probabilities. SM registers semantics at LtM using DTD XML language.

As shown in Fig. 4, SM has the following components: a) Coordinator which is responsible for handling and differentiating between enforcement signals from internal units, NCI and DVA to allow performing semantics derivation, retrieval, matching and discovery processes and to let pass networking data from StM to SM; b) Patterns learning which executes ARL algorithm to learn data patterns in StM and to learn semantics patterns, which describe relation or interdependence among different semantics, used in semantic matching process, in LtM; c) Reasoning algorithms which represent the semantics models by implementing group of AI algorithms, i.e., HMM models, for multi-stage reasoning to perform tasks of semantics derivation on different levels of granularity, i.e., providing FBS data, and generation of semantics patterns; d) Semantics matching which is embedded with a case-based reasoning algorithm, using defined semantics in LtM, and a statistical analysis model for applying semantics fitting processes; e) Semantics adaptation maintains algorithms for defining aspects and features of semantics considering changes happened in those semantics FBS data based on results from semantics matching unit and a control signal from coordinator; f) Semantics definition and goals which shows experience of the SM, preserves semantic derivation models, and definitions and intentions of semantics; g) Semantics placement which has data storage models and definition languages, such as extensible markup language (XML) document type definition, for maintaining semantics in LtM tables; h) Feature extraction and classification which maintains models for probabilistic and statistical analysis beside classification algorithms as decision trees and Fuzzy membership functions to extract and classify features of learned patterns; i) Parse input which is responsible to handle inputs to SM whether those inputs are data (from LtM and StM) or alerts and request signals from DVA and NCI, respectively; and j) Data output which is the gate where analytical reports and semantics are passed through to NCI and LtM/NCI, respectively.

Short-term Memory (StM)

StM stores temporarily SData of different network elements in various domains. Data in StM are classified to application, communication, and resource concern, as defined in [6]. We propose four big relational structured tables for StM, namely service data structured table (SDST), concerns data structured table (CDST), concerns index table (CIT), and composite concerns index table (CCIT). SDST provides data service where it contains entries for multiple network elements and

their related extracted concerns. CDST defines FBS attributes per each concern's data entry in the SDST. The CIT and CCIT clarify in which entries in the SDST, a single concern's and group of concerns' attributes can be found, respectively.

Long-term Memory (LtM)

It is a permanent auto-associative memory of semantics, which are stored on different levels of abstraction (based on three networking concerns and FBS data). LtM design depends on using sets of extensible big relational tables to represent in class tables three classes (application, communication, and resource) of features on which semantics depend. Extracted semantics are maintained in sub-classes tables via numeric vectors and alphabetic symbols which refer to their features in class tables.

NetMem Semantics Extraction Model (via an example)

SM is monitoring N data profiles registered by DVA at StM to learn patterns. We assume that profiles have a length of T features, learned and their number identified by ARL, studied by statistical analysis, and classified by FMF and DT, of group of K features that $T \leq K$. for instance, SM discovers TCP profile then it will study number of occurrence in StM and then it will classify that number using an FMF whether it is large, or normal, or small number. We design an HMM model for SM that provides a relation between sequence of T features, extracted from learned patterns at StM, and generated sequence of T concepts of total M concepts based on prior transition and observation probabilities where $T \leq M$. Based on extracted features, SM extracts semantics concerning one of three networking concerns, namely, application, communication and resource. Fig. 5 illustrates the forward algorithm used by our HMM to extract semantics.

The designed HMM $\lambda = (A, B, \pi)$ model for TCP and UDP service profiles, kept as StM, has T=4 states for input (i.e., features) and output (i.e., semantics). A is defined as a matrix of input state transition probability, which shows the probability to transfer from a feature (e.g., large number of profiles) to another one, B is the observation probability matrix, which shows the probability to have semantics from a certain a feature, and π is the initial state probability for each feature and it equals $1/T = 0.25$, i.e., $T = 1/$ (number of features). Four concerns are extracted from TCP/UDP profiles which are: *protocol type, buffer_size, bandwidth, long_service_duration.* We defined decision tress and FMFs that classify extracted concerns to four features like degree of bandwidth value and range of used buffer_size. Obtained features are input sequence to our HMM. Based on training sequences and HMM parameters, we have at output a sequence of four defined semantics which are *reliable service, huge resources, file transfer, stable communication*. Using the Viterbi algorithm for maximum likelihood estimation, SM extracts semantics for TCP and UDP profiles on different levels of granularity defining; application concern (*file transfer* and *reliable service*) and communication concern (*stable communication*) where it will register them at LtM.

Using HMM model $\boldsymbol{\lambda=(A,B,\pi)}$ and the forward algorithm:
$\boldsymbol{\alpha}$: the probability of partial observation sequence at certain detection time t by the SM giving that there is a certain input state (i.e., feature) at that time

a) Initialization: $\boldsymbol{\alpha_1(i) = \pi_i B_i(concept(1))}$, $1 \leq i \leq K$, where K: number of defined features, $\boldsymbol{B_i(concept(1))}$ is the probability to have first concept from feature i

b) Induction: $\boldsymbol{\alpha_{t+1}(j) = [\sum_i \alpha_t(i)\, A_{ij}] B_j(concept(t+1))}$, $1 \leq i \leq K$, $1 \leq j \leq K$, $1 \leq t \leq T-1$

c) Termination: $\boldsymbol{P(concept\ sequence/\lambda) = \sum_i \alpha_T(i)}$, $1 \leq i \leq K$, considering accumulated value of $\boldsymbol{\alpha}$ of T (length of sequence) states

Fig. 5. A Probabilistic Model using HMM for Semantics Extraction

III. Evaluation

We conducted simulations using JSim [21] for our preliminary evaluation of the efficacy of NetMem. We implemented a simple scenario for a network composed of eight hosts, two nodes with routing functionalities "routers", and a shared NetMem system represented by an overlay network comprises entities for NCI, DVA, StM, LtM, and SM operations. NetMem system was implemented by Java for handling data from sources and semantics request/response messages from nodes, profiles' building, feature extraction and classification and the used ARL algorithm and HMM model. In this scenario, we designed an HMM model for extracting semantics related to abnormal profiles and attacks. A simple communication protocol was built to enable interaction between NetMem and other nodes. Our scenario goal, basically, is to show the capability of NetMem for understanding at runtime and on-demand networking contexts (e.g., running services and their time-varying requirements) via learning and deriving semantics related to a) normal/abnormal flows of running services; and b) attacks and use those semantics for predictive operation to enhance QoS of running services and to strengthen security by anomalies detection. We have two different service classes for data transfer where the first service class uses TCP transport protocol between two hosts and the other service class uses file transfer protocols on top of UDP among other two hosts. We build a static route for each service class where intermediate routers transfer data packets of both services. The other four hosts are attackers (i.e., non-legitimate entities) which generate abnormal TCP/UDP flows to degrade services' QoS of legal entities. Table I shows simulation parameters.

Legal entities in the scenario can access NetMem at runtime and on-demand to store/discover/retrieve data and to learn semantics, or sequence of concepts, concerning their interesting services. For instance, the TCP and UDP services are provided in a specific region and at certain periods of time through the day. Data profiles regarding traffic of those services are stored in NetMem by DVA at StM. Those profiles show source/destination IP, service type, port number, packet size and type, allocated buffer size and bandwidth, and service duration. Profiles in StM exhibit patterns that can be learnt by SM to derive semantics related to those services. Communicating entities such as routers can know via a) data profiles which clarify IP attributes; and b) TCP/UDP semantics: requirements of running services, and the normal and abnormal behavior of services. For example, NetMem has semantics for the TCP service which reveal that this service, within range of IPs, uses specific resources at certain time period and at particular area because of the impact of another

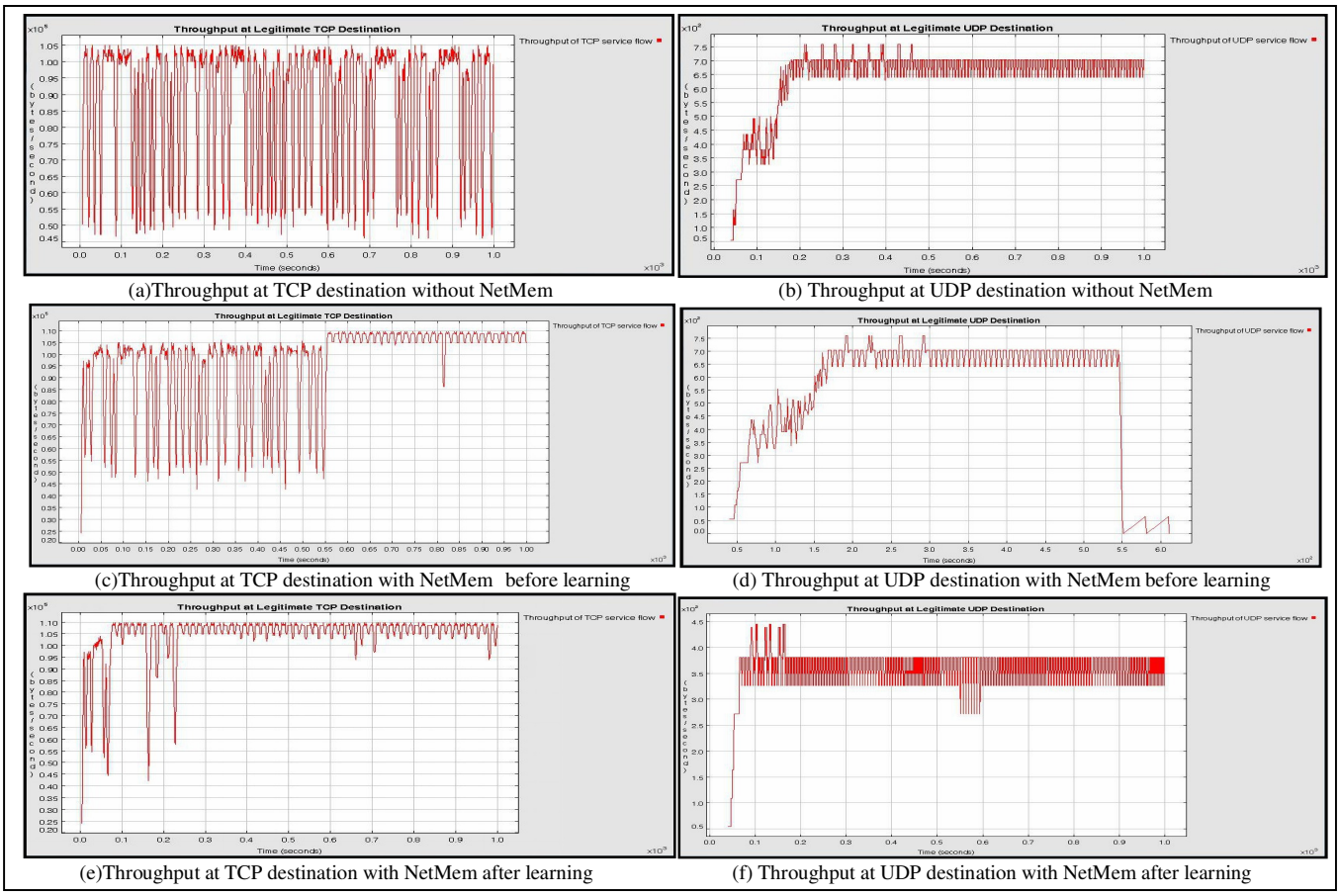
TABLE I
SIMULATION PARAMETERS

Parameter	Value
Link Data Rate (fixed)	1 Mbps
Router Buffer Size (fixed)	7000 packets
TCP MSS (fixed) Normal/Abnormal Flow (1) & (2)	512 /1024 & 2048 bytes
TCP MCWS (fixed) Normal/Abnormal Flow	128 bytes
TCP Time to Live (fixed) Normal/Abnormal Flow	255 seconds
UDP Client/Reply Timeout	30 seconds
UDP Packet Size (fixed) Normal/Abnormal Flow (1) & (2)	512/ 512 & 2048 bytes
Start Time: Normal TCP flow/two abnormal TCP flows/Normal UDP flow/two abnormal UDP flows	From beginning/form beginning/40 seconds/90 and 100 seconds
Propagation Delay (fixed)	100 mseconds
Rate of Using Sensory Functions in Hosts for Recognition of Attack Alerts (variable)	Every 60 seconds
Rate of NetMem Access and Data Patterns Detection in StM by SM (variable)	Every 100 seconds
Rate of Change for Contents (i.e. Data Patterns) in StM	Every time Hosts send/receive data
HMM approach/number of training sequences	Unsupervised using Baum-Welch algorithm/ 1000
Simulation Time	1000 seconds

service (i.e., UDP service).

To derive semantics, SM analyzes the different profiles in StM where it learns group of concerns (e.g., packet_size) in profiles using our designed ARL algorithm. After that, SM discovers features per each profile by classifying obtained concerns through using our defined decision trees and FMFs. Sequence of classified features per each profile is sent to a defined behavior HMM model to extract semantics, i.e., sequences of concepts, and to learn normal/abnormal behavior of services and attacks. The Accuracy of SM prediction for profiles' behavior is evaluated every reasoning time period, i.e., 100 seconds. In our HMM model, we defined five concepts, namely, known attack, UDP flood attack, TCP_Syn_flood_attack, normal and abnormal flow. Those concepts depend on extracted and classified features per each profile, which are related to number of each profile in StM, packet size, port number, number of synchronization packets. Normal/abnormal behaviors for services are defined in HMM model based on values' range of input features. For instance, a normal behavior of TCP services is assigned by certain range of port numbers and packet sizes. We propose four cases of operation. The first case is that NetMem operation before learning that SM does not learn and derive concepts yet. The second one is operation after learning concepts; and SM will perform concept matching processes for learned data patterns. The third case is operation before learning and with DVA control. DVA control is a simple mechanism proposed to minimize storage space in StM that each data profile will not exceed a ratio of total number of profiles in StM. The last case is operation after learning and with DVA control.

Fig. 6 shows measured average throughput at the two



(a)Throughput at TCP destination without NetMem
(b) Throughput at UDP destination without NetMem
(c)Throughput at TCP destination with NetMem before learning
(d) Throughput at UDP destination with NetMem before learning
(e)Throughput at TCP destination with NetMem after learning
(f) Throughput at UDP destination with NetMem after learning

Figure 6. Throughput at TCP/UDP Destinations with/without NetMem and before/after learning

legitimate TCP/UDP destinations in cases of operation with/without NetMem before/after learning and with/without DVA control. Fig. 6 (a, b) shows operation without NetMem that according to abnormal TCP/UDP flows, QoS of TCP service is deteriorated and also much resources consumption at UDP destination due to UDP flood attacks initiated from non-legitimate entities. In Fig. 6 (c, d), the operation is with NetMem, however, LtM does not maintain concepts that can be matched with learned patterns of running services' profiles. So, SM begins to learn and derive concepts during the simulation time. SM could learn patterns of three service profiles at StM which reveal two types of attacks and an abnormal flow behavior. Accordingly, three concepts are constructed and stored at LtM. Routers in our scenario could learn those concepts in LtM through linkage between known IPs for those routers and service types attributes in profiles at StM, and extracted semantics at LtM and thereby routers stopped allocating resources and processing data concerning abnormal services' flow. Hence, QoS of legitimate services begins to enhance, however, after overhead time due to the patterns learning phase by SM, and LtM access and semantics retrieval by routers. Legitimate UDP flow stops because arrival of many datagrams from attackers increases probability of collision which make buffer overflow and result in exceeding legitimate flow for the client and reply timeout value (30 seconds). In Fig. 6 (e, f), NetMem operates after learning patterns and maintaining semantics for abnormal flows and attacks. So, routers learn early semantics of their services and they limit resources assigned to abnormal flows.

SM keeps learning data patterns at StM and it matches them with registered semantics. Fig. 7 illustrates SM timeliness to analyze and learn patterns of service profiles, which are updated continuously at runtime, to derive semantics accordingly and to predict services' behavior. In cases operation after learning, SM takes time longer than cases of operation before learning because SM learned semantics of attacks and abnormal profiles and it found matching between analyzed and learned patterns and those semantics. Accordingly, routers will not assign resources for those profiles and normal traffics of TCP/UDP services will reach destinations successfully (data collisions and channel access contention are minimized). So, network throughput will be enhances and NetMem will be able to collect large number of normal profiles. Hence, StM size increases after learning as shown in Fig. 8. NetMem could achieve same rate of learning in case of operation before learning with/without using the simple DVA control. In addition, NetMem, with DVA control after learning, could minimize required storage space at StM. However, SM could discover 100% of semantics.

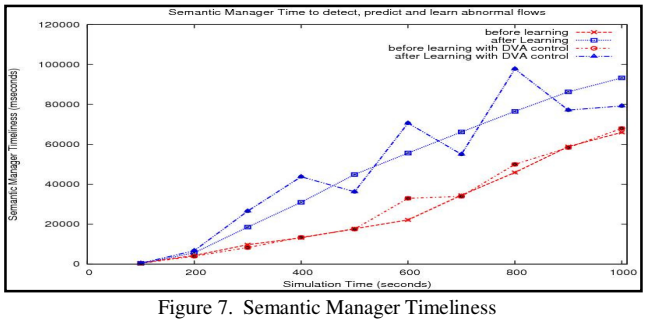


Figure 7. Semantic Manager Timeliness

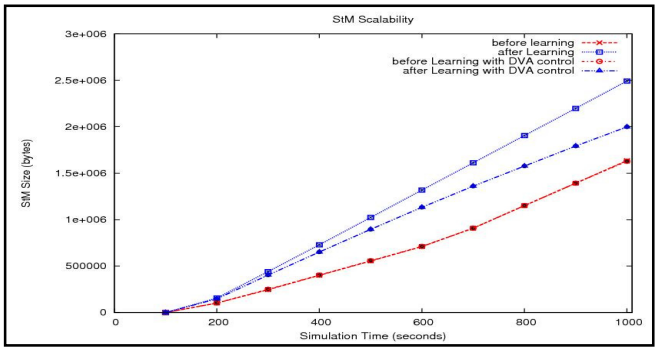


Figure 8. Data Scalability at StM with/without DVA Control

IV. Related Work

Some previous works have implemented mechanisms for storing Internet data and measurement data based on required attributes. For instance, flexible meta-databases as in [9]-[11] were proposed to enable accessing uniformly represented and correlated data from heterogeneous sources to support better data analysis and understanding of networking traffics. Also, in [11]-[15] and [19], authors presented architectures for knowledge plane (KP) for autonomic knowledge management capabilities to strengthen self-* properties and to enhance decision making and understanding network dynamics. Researchers have also investigated mechanisms for storing Internet and measurement data based on certain attributes. For example, CAIDA in [8] presented the Internet measurement data catalogue (IMDC) as metadata 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 [9] proposed a scalable Internet measurement repository (SIMR) to track Internet measurements where large databases provide information about measurements, tools, users, experiments, and datasets. These databases can be accessed easily for analyzing obtained measurements at various contexts and times. Similar to SIMR, the MOME in [10] is a web-based way for repositories of meta-databases independent of service types. Authors of MOME showed the capability to record information related to measurement tools and data (e.g., QoS attributes) where there is a public access capability, but processes of updating/retrieving entries are limited to registered users.

The work in [11] and [12] proposed to add a distributed knowledge plane (KP) to the current Internet, based on artificial intelligence and cognitive techniques, to be self-knowledgeable, self-configuration, self-diagnosing, self-analyzing, and self-managing. 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. Riggio *et. al* in [14] proposed a KP for wireless mesh networks where communicating nodes can exchange information related to services and operating conditions. 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 [19], authors presented a knowledge plane for autonomic networking based on machine learning tools to enhance self-* properties (especially for self-adaptation) of communicating entities in distributed environments. In [19], an KP is constructed virtually via knowledge sharing and discovery processes which are performed by networking entities. . The KP in [19] provides data access and retrieval on levels of internal and environmental states.

Similar to works in [8]-[10], NetMem adopts metadata models for uniformly representing data related to various network elements from various sources. However, NetMem is a semantics management system that maintains network semantics at different levels of granularity based on learning spatiotemporal data patterns originating from heterogeneous sources. Semantics can be accessed on-demand and at runtime to support decision making and tasks of networking built-in tools as intrusion detection systems. Compared to work in [11] and [12], we provide a biologically-inspired architecture for NetMem, where human memory functionality guides its constituting components and internal operation for collecting and homogenizing data from various sources using data virtualization techniques. The proposed KP in [12] uses rules to generate responses based on gathered observations. But, NetMem adopts predictive analytics based on learned patterns and extracted features to generate semantics. Unlike work in [13],[14] and [19], NetMem is designed to provide a shared distributed management system for network semantics, which are associatively stored using data models in big relational tables. Also, semantics can be accessed and learned by networking entities at runtime and on demand to aid in enhancing networking operations such as anomaly detection. Unlike work in [15], maintained data in NetMem are not restricted to Internet measurement data. NetMem maintains network semantics related to different network elements.

V. Conclusion

We presented NetMem, a semantics management system that mimics functionalities of the human memory with the objective of effectively and efficiently utilizing and learning spatiotemporal networking data patterns for smarter networking. NetMem provides capabilities of associative semantics storage/retrieval/matching at runtime and on-demand for supporting enhancement of running services' QoS guaranties and anomaly detection. NetMem integrates data virtualization, cloud-like data storage, associative rule learning and HMM. We provided via simulation a simple evaluation for networking operations with NetMem. Motivated by the preliminary results showing NetMem's success in improving networking operations, we are currently investigating the formalization and optimization of NetMem as a generic scalable network memory and addressing its real implementation challenges and overhead. We also seek to motivate further research particularly leading to reference model and architecture and standardization of network memory.

References

[1] L. Tan and N. Wang, "Future internet: The Internet of Things," *Advanced Computer Theory and Engineering (ICACTE), 3rd International Conference on* ,vol.5, no., pp.V5-376-V5-380, 20-22, 2010

[2] A. Feldmann, "Internet clean-slate design: what and why?" SIGCOMM Comput. Commun. Rev., vol. 37, no. 3, pp. 59–64, 2007.

[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, 2010.

[4] Data Virtualization by Composite Software. [Online] Available: http://www.compositesw.com/solutions/data- virtualization/

[5] Data Virtualization by denodo technologies. [Online] Available: http://www.denodo.com/en/solutions/technology/data_virtualization.php

[6] H. Hassan, M. Eltoweissy, and M. Youssef, "Cellnet: A bottom up approach to network design," in New Technologies, Mobility and Security (NTMS), 3rd International Conference on, pp. 1–6, 2009.

[7] S. Rozsnyai, A. Slominski, and Y. Doganata, "Large-scale distributed storage system for business provenance," in IBM Research Report, RC25154, April-2011.

[8] Correlating heterogeneous measurement data to achieve system-level analysis of internet traffic trends, in CAIDA Project. [Online]. Available: http://www.caida.org/projects/trends/

[9] 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.

[10] P. A. Ar, A. Gutirrez, A. Bulanza, M. Dabrowski, B. Kaskina, J. Quittek, F. Strohmeier, A. Vidacs, K. S, and O. Zsolt, "An advanced measurement meta-repository," March-2005.

[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, 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] Apache hbase. [Online]. Available: http://hadoop.apache.org/hbase/

[17] Apache hadoop. [Online]. Available: http://hadoop.apache.org

[18] J. Hawkins and S. Blakeslee, *On Intelligence,* Times Books Henry Holt and Company, LLC, 261 pages, First Edition 2004.

[19] M. Mbaye and F. Krief, "A collaborative knowledge plane for autonomic networks," in Autonomic Communication. Springer US, 2009, pp. 69-92.

[20] J.S. Gero "Design Prototypes: A Knowledge Representation Schema for Design," AI Magazine, vol. 11, pp. 26–36, 1990.

[21] JSim. [Online]. Available: http://sites.google.com/site/jsimofficial/start-with-j-sim.

[22] Y.-C. Sun and G. Clark, "A computational model of an intuitive reasoner for ecosystem control". *Expert Syst. Appl.* 36, 10 (December 2009), 12529-12536.

[23] S. Paul, "An optimized distributed association rule mining algorithm in parallel and distributed data mining with xml data for improved response time", *International Journal of Computer Science and Information Technology*, vol. 2, number 2, April 2010.

[24] M. Winter, *Goguen categories: a categorical approach to L-fuzzy relations*, Springer, 2007.

[25] Rabiner, L.; Juang, B.; , "An introduction to hidden Markov models," *ASSP Magazine, IEEE* , vol.3, no.1, pp.4-16, Jan 1986.

[26] W. Zeng, Y. Zhao, K. Ou, and W. Song, "Research on cloud storage architecture and key technologies," In Proc. of the 2nd International Conference on Interaction Sciences (ICIS '09). ACM, 1044-1048, 2009.

[27] R.W. Thomas, L.A. DaSilva, and A.B. MacKenzie, "Cognitive networks," *New Frontiers in Dynamic Spectrum Access Networks, First IEEE International Symposium on*, vol., no., pp.352-360, 2005.