International Journal of Energy, Information and Communications
Vol.8, Issue 1 (2017), pp.51-64
http://dx.doi.org/10.14257/ijeic.2017.8.1.05

An Agent-based Data Analytics Support Tool for Network Management Intelligence

Kazuto Sasai[1], Yusuke Tanimura[2], Hideyuki Takahashi[1], Gen Kitagata[1] and Tetsuo Kinoshita[1]

[1]Research Institute of Electrical Communication, Tohoku University
[2]Graduate School of Information Science, Tohoku University
kazuto@riec.tohoku.ac.jp

Abstract

Data analytics technics are attracted to the field of network management. However, since a vast amount of data is produced from the systems, it is difficult to analyze effectively in ordinal network administrative office. In this paper, we propose an agent-based approach to support network and systems data analysis. Using the functions of agent-based systems to interact each other, collaborating with an agent-base network management system, it makes the data analysis tasks to be easy. Actual data analytics experiment on the network management task, we found that the agent-based analytics tool can effectively support the novice network and systems administrators.

Keywords: *agent system, data analytics, AIR-NMS, symbiotic cognitive computing*

1. Introduction

Data analysis increases its importance in network and systems management because of the recent complex network environment such as cyber-attacks, IoT, smart phones. From that aspect many factors such as fault detection, anomaly detection, performance evaluation, and prediction [1]. For the cloud resource management, data mining is used to predict the future demand of computing resources [2, 3, 4], and for the traffic anomaly detection [5], model analysis applies to huge amount of traffic data which is obtained by long-term observation [6]. In the field of data center management, correlation analysis is used to detect anomalies in the sensors which is allocated in the large data centers [7, 8]. On the other hand, there is a study about the tools to support data analysis processes using a rich computing resource in the cloud foundation [9, 10]. Actually, some cloud companies provide the services to support data analytics [11, 12, 13]. Although the tools and the services provide many support functions for data analysis processes, it is not enough to solve the difficulty of data analytics on network and systems data because it remains some domain dependent matters how to collect the data, how to format the data and how to extract knowledge from the data. Therefore, since human administrators are forced to spend much time to configure the data processing for the analysis task, a new support function which provides the intelligence about the configuration of network and systems data analytics construction.

To reduce complexity of network and systems management, there is a challenge to assemble the autonomic features of biological systems to the network devices, which is called autonomic network management [14]. Conventionally, autonomic actions in biological systems are represented with the sensory motor coupling [15], which indicates that a biological system has several sensors and complex actuators, and cognitive and/or decision-making functions bridge the two modules. Some practical implementation of decision-making functions of network systems is demonstrated [16, 17]. However, since the studies depend on the predefined action and decision rules, the rules have to be updated according to the environmental changes. Further, although the researchers

ISSN: 2093-9655 IJEIC

continuously provide new methodologies to understand the data collected from the network systems, it is difficult to deploy new methodologies to the autonomic systems. Therefore, a new kind of autonomic network management foundation on which human developers and maintainers easily build, reuse, and exploit knowledge resources to the network management systems.

Recently, Kephart [18] asserted that a view of decision-making systems should include both of machines and humans, and they collaborate each other to solve the problems, and they called that *symbiotic cognitive computing*. In this paper, as a foundation for symbiotic cognitive computing, we propose a function of supporting network and systems data analytics collaborating with our data-centric network management system, called Active Information Resource-based Network Management System (AIR-NMS) [19]. AIR-NMS is a set of active information resources (AIR) for network and systems management, which consists of individual information resource including the autonomic features to realize self-management [20]. We introduce an agent-based design of network data analysis tools that are autonomously organized by according to the request of users. It is expected that highly reusable and deployable symbiotic cognitive systems can be established by connecting agent-based network data analytics tools and AIR-NMS. Figure 1 shows the conceptual diagram of the concept proposed in this paper.

In Section 2, we briefly introduce about AIR-NMS and agent-based analysis function for network and systems management. Section 3 includes the practical design scheme and a prototypical implementation. Section 4 shows some experiments which were conducted to evaluate the prototype system. Section 5 shows the conclusion of this paper.

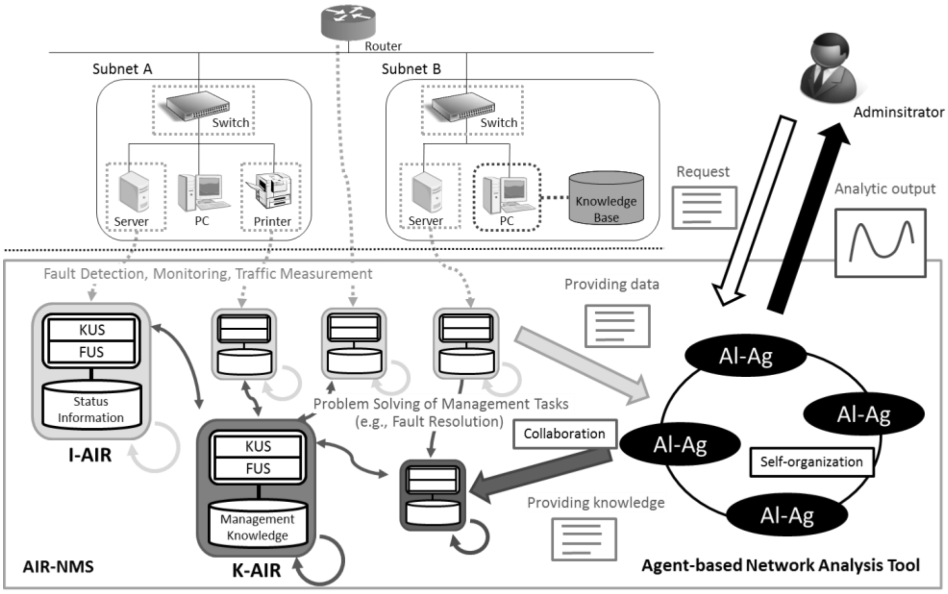

Figure 1. Conceptual Diagram of the Proposal

2. Agent-based Analytics for Network Management

Agent-based network management is a concept that the agents perform the tasks of network management instead of the human administrators, and the complexity of the network management tasks is reduced by the collaboration between the agents [21, 22]. On the other hand, the software agent technologies are applied to data analysis to scale down the complexity of the information analysis process by applying self-organization as conceptually introduced in [23]. Even so, since the origin of raw data such as detectors and these analysis tools is separated and dealt out, it is difficult to deploy new analysis methodologies into on-going analysis systems. As reviewed in [23], a data mining process for network and systems data can be formalized with the five steps of (a) target data, (b) preprocessed data, (c) transformed data, (d) mined patterns and (e) acquired knowledge. From the requirement for human-agent collaboration, we reorganize the process into a general process of data analytics for network and systems management,

A) Data collection,

B) Preprocessing,

C) Analysis,

D) Visualization.

Here, we generalize the analysis step that includes mining process, statistical process, prediction process. In other words, we extend the model of [23] to support human participation in the network data analytics. Figure 2 shows the schematic diagram of redefinition of data mining process model for the self-organized network data analytics.

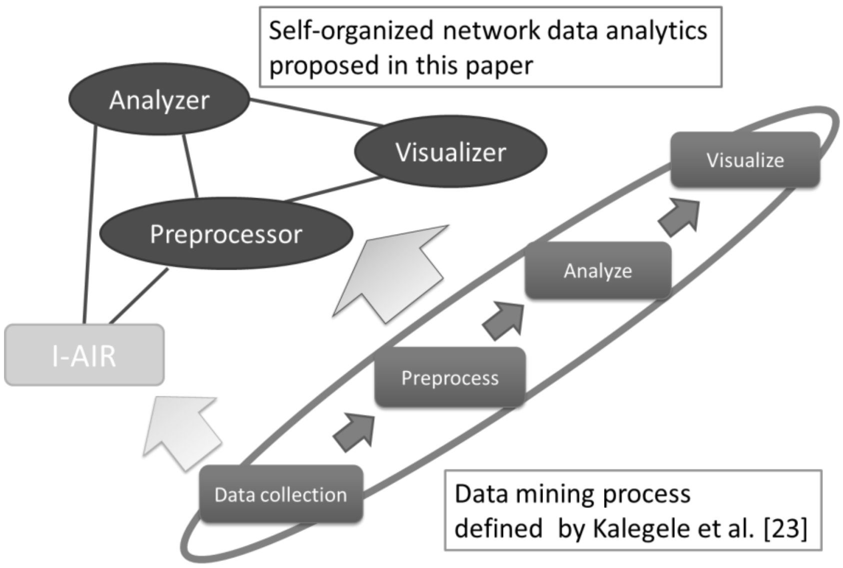

Figure 2. Redefine of the Analysis Process

2.1. AIR-NMS

So far, we have developed the agent-based information resource management model which we call Active Information Resource (AIR), and we call a network management system designed with the model as AIR-NMS [19]. An AIR is structured by the information resource itself, the mappings to access and process, and the knowledge to interpret meanings and to join the other AIRs. To share with the information resources about network management, we categorized the AIRs into two types, I-AIR and K-AIR. I-AIR is categorized that includes status data collected from servers, network equipment, and other devices. K-AIR includes the knowledge that recognizes what happen in the target network, and that execute some recovery action to the network system. The I-AIRs and the K-AIRs in an AIR-NMS are always linking each other, and autonomously exchange messages. An example of prototypical implementation is presented in [19]. The knowledge of the fault detection and recovery such as symptoms, causes, and countermeasures are stored in the K-AIRs, and they can apply knowledge by collecting specific information on the current status and the configurations from the I-AIRs. Since the I-AIRs can support data acquisition of human administrator, it is reasonable to provide the data analytics support tool as an advanced feature of AIR-NMS. In this paper, we proposed the three agents with which the steps from B) to D) are replaced, and design the self-organized feature of the data analytics process using collaboration function of the agents.

2.2. Analysis Agents

As shown in Figure 1, we define an analysis agent (Al-Ag) is an agent that performs the analysis tasks that are the component of a complete analysis process, which includes collecting, preprocessing, analyzing, and visualizing objective data. Al-Ags are designed to collaborate each other in order to connect the output of an agent to the input of another agent. Since the Al-Ags has the capability of self-organization as below, the analysis process is composed on demand, and the administrator can easily achieve the analysis results without considering the compatibility of the components, which mostly requires professional knowledge. In addition to Al-Ags, we design the collaboration scheme between an organization of Al-Ags and AIR-NMS. It supports the task of data collection using the function of I-AIRs, which can respond to the request messages of data retrieve. Further, by extracting and forming knowledge from the analysis results, it is possible to return the exploitable knowledge that can be used as the management knowledge included in K-AIRs.

In this paper, we design Al-Ags according to the development framework of software agents, ADIPS/DASH [24, 25], which provides a production system for knowledge utilization and a FIPA compatible message exchange mechanism among DASH agents. The DASH framework has an agent repository where the agents are active and ready to construct multiagent organizations according to the users' requests. A structured organization can be allocated to the distributed workplaces.

Here, we design and implement the prototypical Al-Ags, which includes experimental and simplified templates of the three agent types, Analyzer, Preprocessor, and Visualizer.

2.2.1. Analyzer: has a function to perform processing an analysis method to objective data, and also has the knowledge to support efficient processing which consists of required preprocessing, possible visualization methods and some result meaning support information of the result.

2.2.2. Preprocessor: has a function to transform a data to the appropriate format for the successive agents. For instance, in the case of data mining, preprocessing means preparation of target data set such as case balancing, probabilistic sampling and so on. Since the role of Preprocessor is important to improve the performance and accuracy of analysis, it should have the knowledge to control parameters for the transformation.

2.2.3. Visualizer: receives processed results from Analyzer. If the data format satisfies the requirements to plot format, Visualizer displays plotting results in the user interface, else if it is not, sending a request message to Preprocessor to convert data format, else if there is no conversion method, it comes back to waiting state.

2.3. Self-organized Data Analytic Process

When the user agent sends user's request to the Analyzers in the agent repository, and the Analyzers check the request whether it can be accepted or not. After that, if an analyzer accepted the request, it sends messages to the other Al-Ags necessary to make the analysis process, and simultaneously send broadcast message to ask the required data to AIR-NMS. If all conditions are satisfied, the Analyzer sends messages to the members of the organization which involves the AIRs and the Al-Ags. When the agents in the organization are instantiated, they check the individual destination, which indicates the subsequent agent, and then they can establish the link status. Finally, the I-AIR starts to send data to the next Al-Ags as mentioned above.

3. Design and Implementation of Prototype System

In this section, we explain how to design the collaboration scheme between the agents, and also between the agents and AIR-NMS. Figure 3 shows a schematic diagram of the prototypical system. The processing flow of the prototypical system is as follows:

1. When the user tells a request to the user agent (User-Ag), the User-Ag sends a request message to the agents in the agent repository.
2. When an agent receives the request message, a self-organization process is executed, and an organization is constructed as a result. In this process, the message is also sent to AIR-NMS, and an I-AIR which has the required data is selected.
3. After the self-organization process, the agents in the organization are instantiated (activated) in the target agent workplace, which is depicted as WP2 in Figure 3.
4. The I-AIR starts to send the required data to the prescribed agent.
5. Then the agents start to execute their functions.
6. The selected output agent (probably a Visualizer in usual) shows the output to the user.

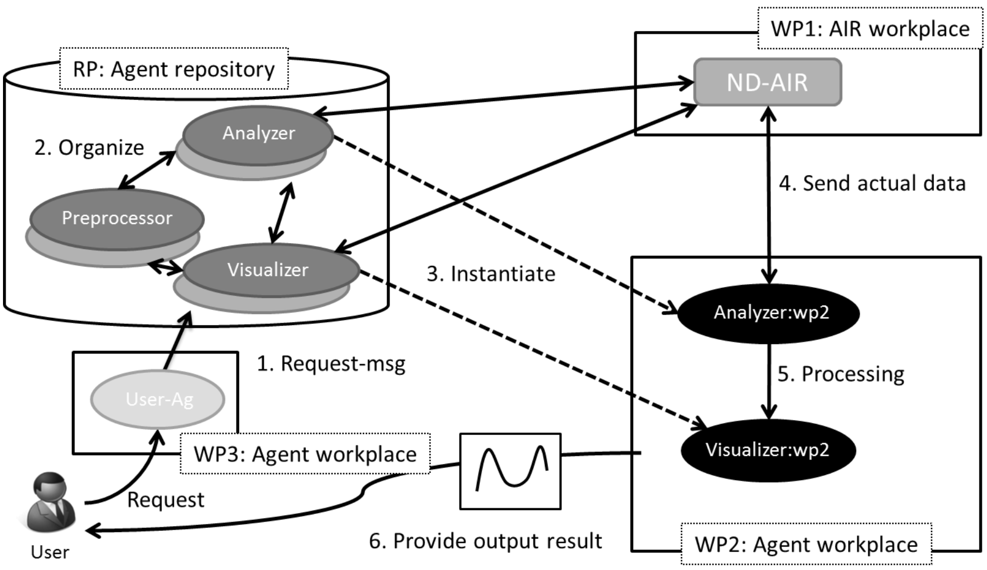

Figure 3. Schematic Diagram of Prototypical System

3.1. Design of Analysis Agents and I-AIRs

As the prototypical system, we have implemented the seven analysis agents, including the two Analyzers and the two Preprocessors and the three Visualizers, listed in Table 1. According to the design scheme of ADIPS/DASH agent development framework, an agent is structured by the three kinds of elements, initial facts, rules and base process. Table 2 shows an example structure of an analysis agent. The base process changes depending on the agents, but the other parts of the structure can be commonly used for the other agents. The initial facts and the rulesets describe the knowledge about the analysis method, which mainly includes what types of agents are possible to perform as the data source and destination of the agent. By the prescribed these knowledge, the analysis agents can compose the required organization of analytics.

Table 1. Examples of Agents

Name	Type	Description
ServerLoad	Analyzer	Calculate server load
Fourier	Analyzer	Calculate Fourier transformation
WindowFunction	Preprocessor	Apply window function
SSVtoCSV	Preprocessor	Convert text file format from ssv to csv
TimeSeriesGraph	Visualizer	Output time series graph
LineGraph	Visualizer	Output line graph
ScatterGraph	Visualizer	Output scatter graph

Table 2. Example Structure of an Analysis Agent

Initial facts:	Expression
`role: analysis`	Agent type.
`function: fourier`	Agent function.
`needInfo: null`	Required data format.
`processedInfo: spectrum`	Output data format.
`output: scatterGraph`	Possible output method.
Rules:	Expression
`receiveMsg-RA`	Receive request and search possible input and output agent.
`receiveReplyMsg-SI`	Set input.
`receiveReplyMsg-SO_thisAgRep`	Set output.
`instantiateThis`	Instantiate myself in the workplace.
`requestMsg-RD_bulk`	Request data to the AIR.
`requestReplyMsg-RD_bulk`	Start processing.
Base process:	Expression
`doFourier`	Executing Fourier transformation.

3.2. Designing Collaboration with AIR-NMS

The AIR-NMS is also designed and implemented according to the agent framework ADIPS/DASH. In this paper, we focus on the I-AIR which stores various status information about the target network system and design the message exchange scheme between the I-AIRs and the Al-Ags. The detail and the other part of AIR-NMS are explained in [20]. In the self-organization process of the Al-Ags, the agents should check whether the data required for processing exist or not. The process of data retrieving is realized by message exchange between the Al-Ag and the I-AIRs. The Al-Ag sends the data retrieve message into the I-AIRs by using the addressing service of agent repository, which manages the name of the connected workplaces and the agents instantiated on the workplaces, and it is a function of ADIPS/DASH runtime environment. Additionally, we redesign the response of the I-AIRs to send the return message to the Al-Ag. Since I-AIRs are originally designed to manage the stored data autonomously, from the perspective of the Al-Ags, they can retrieve the data without considering about the data sources. Consequently, we have established the interoperability between Al-Ags and I-AIRs, and the analysis support system consists of the Al-Ags can effectively utilize the function of the I-AIRs. We show the redesigned structure of the I-AIR in Table 3.

Table 3. Example Structure of an I-AIR

Initial facts:	Expression
`role: AIR`	Agent type.
`info: trafficVolume`	Agent function.
`element: mailServer`	Required data format.
Rules:	**Expression**
`receiveReplyMsg-SI`	Set input.
`requestMsg-RD_csv`	Request data to the AIR.
`requestMsg-RD_ssv`	Start processing.
`updateData`	Update database.
Base process:	**Expression**
`storeData`	Add new data to the database.
`getData`	Return stored data.
`toCSV`	Convert data to CSV format.
`toSSV`	Convert data to SSV format.

3.3. Designing Self-Organization Scheme

Figure 4 shows the scheme of message exchange scheme between the agents in the composition phase, which executed in the agent repository. Firstly, an analytic request is sent from User-Agent to Analyzer, here it is "Fourier." The request includes *analysis method*, *data type / name*, *data source*, and *time window*. A user has to select these items at the initial setup. After Fourier agent receives an analytic request, the agent retrieves specified data source by sending message to I-AIR. If the specified data exists, the AIR replies the location and the format. After the Fourier agent receives the reply from I-AIR, the agent compares the data format with the initial fact of "needInfo". If the fact does not match, the agent sends the message to search the Preprocessor agent. Else if the fact matches or is null, the agent sets the location as the data source. Since Fourier agent has the "processedInfo" of "scatterGraph," the agent searches the agent which has the function of scatter graph. Then, the Scatter Graph agent matches to the conditions and the Fourier agent sets the destination as "Scatter Graph."

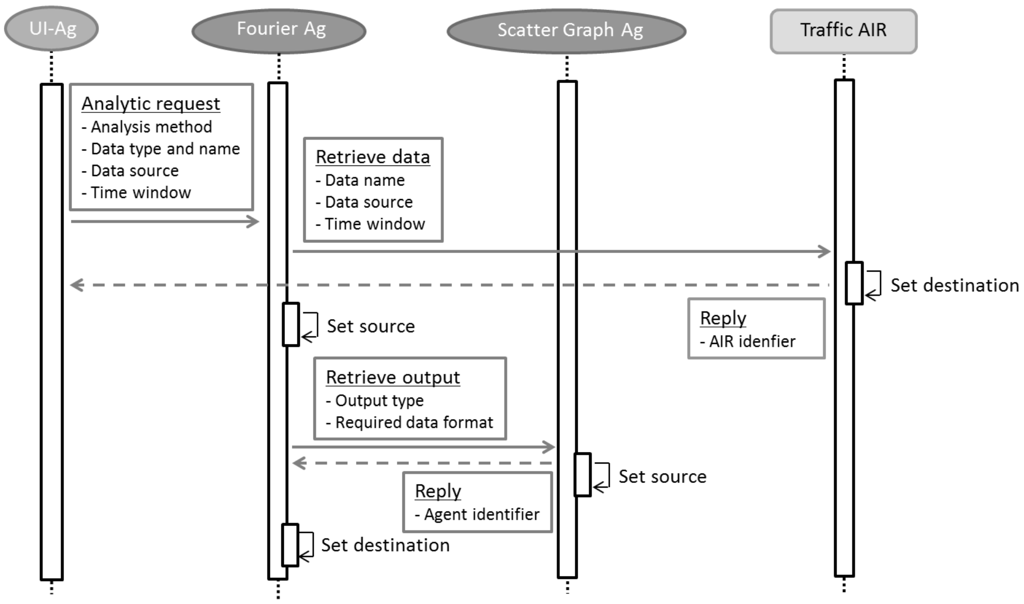

Figure 4. Schematic Diagram of Collaboration

3.4. Implementation

In this section, in order to develop the prototype system to support analysis task of network data, we consider two cases of the practical analysis of network data, a performance evaluation of a server and a time series analysis of a network traffic data. The performance evaluation is a task to calculate the load of the target server, and the time series analysis is a task to apply Fourier transformation to the time series of network traffic [26]. According to the aforementioned design, we implement eight agents, including two Analyzers, two Preprocessors, three Visualizers and User-agent, and their agents run on the ADIPS/DASH environment. Figure 5(a) is a display of IDEA, which is the development environment of ADIPS/DASH systems. The oval shapes represent the agents instantiated in the repository and the workplaces. Figure 5(b) shows the user interface provided by User-Ag in Figure 3, and the user can input the four entities, analysis method, object, target server and interval. "Analysis method"" represents the method that the user will use. The user can select the method from the drop down box. "Object" indicates what type of data is used for the analysis. The user can also select the types from the drop down box. "Target server" represents the host to which analysis is applied. "Interval" indicates the time interval of the data. Figure 5(b) shows the window generated by visualizer. Here, the result of Fourier transformation of the network traffic data is displayed. The user can recognize the result of analysis by using the individual visualization method of the agents.

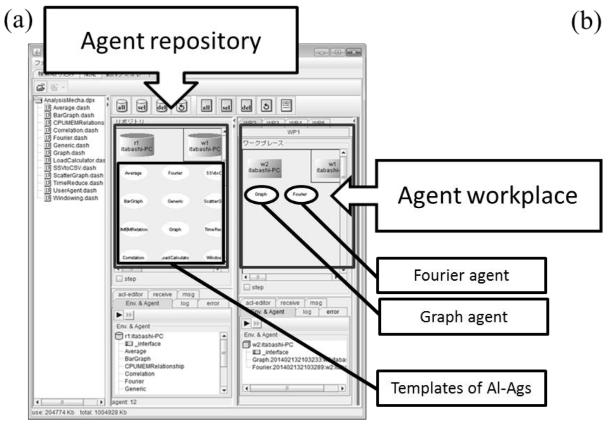

Figure 5. Implemented System

4. Evaluation Experiment

In order to evaluate the prototypical implementation as an analysis support system, we conducted an experiment with the human participants. In the experiment, the participants are asked to deal with some tasks of analyzing network data. We compare the usability of our prototype system with manual strategy that the user can only use existing tools with any user-manuals. In this paper, we employed the Fourier analysis of the network traffic data for this experiment. The task was defined by investigating the researches about analysis of network data. Figure 6(a) shows the experimental environment. The experimental system consists of two computers, Node 1 and Node 2, connecting to the same LAN. Node 1 is used as a console that the participants use. Node 2 is a server which observes network traffic and stores the data. The participant in the Figure 5(b) is performing the analysis task, and we record and monitor the behavior of the participant by the cameras and screen capturing. After the experiment, we classified the behaviors of the participants to some categories of actions, e.g., run software, search on a browser, etc.

4.1. Experimental Set Up

In an experiment, the participant can use any software on the Node 1 including search engines and analysis tools except asking to experimenter. In the case of manual strategy, some particular values such as IP address are preliminarily given by the instruction paper, which supposes a manual. The values and instructions are listed below:

- *Objective Summery* gives meanings of Fourier analysis for traffic amount data, that are to detect malicious activities on the node and to characterize statistical behavior of clients. Further, we give an additional information that usual server traffic data shows simple white noise or sometimes 1/f noise [27].
- *Analysis Steps (manual)* are presented as,
  1. Retrieve data file from Node 2,
  2. Extract traffic amount per 1 minute,
  3. Apply Fourier transformation of the target data,
  4. Make a graph from the result of the power spectrum,
  5. Estimate meaning of the result.
- *Analysis Steps (proposal)* are presented as,
  1. Execute IDEA (with introduction to DASH/IDEA),
  2. Instructions for UI (shown in Figure 5(b)),
  3. Estimate meaning of the result.
- *Available Tools* installed in Node 2 are listed.
- *Additional Information* is given as follows:
  - The IP address and hostname of Node 2.
  - Traffic data are stored on Node 2 in the form of a single file. The directory where the file is located is given in this part.

When the participants use our prototype system, the experimenter tells how to use the system at the beginning of the experiment, though the instruction is just how to run the system.

4.2. Results

Figure 7 shows the example of resulted output. Figure 7(a) is an example of the output in the manual case. The participant in that case uses Microsoft Excel to write the figure as displayed. Figure 7(b) is the case using the proposed system. It is easily found out the output from the case using proposed system can display the result matched for the purpose of the experiment. From the classified actions, we extracted the participants' actions. Table 4 shows an example of the action series of a participant. From the recoded action series data, we compare the two cases.

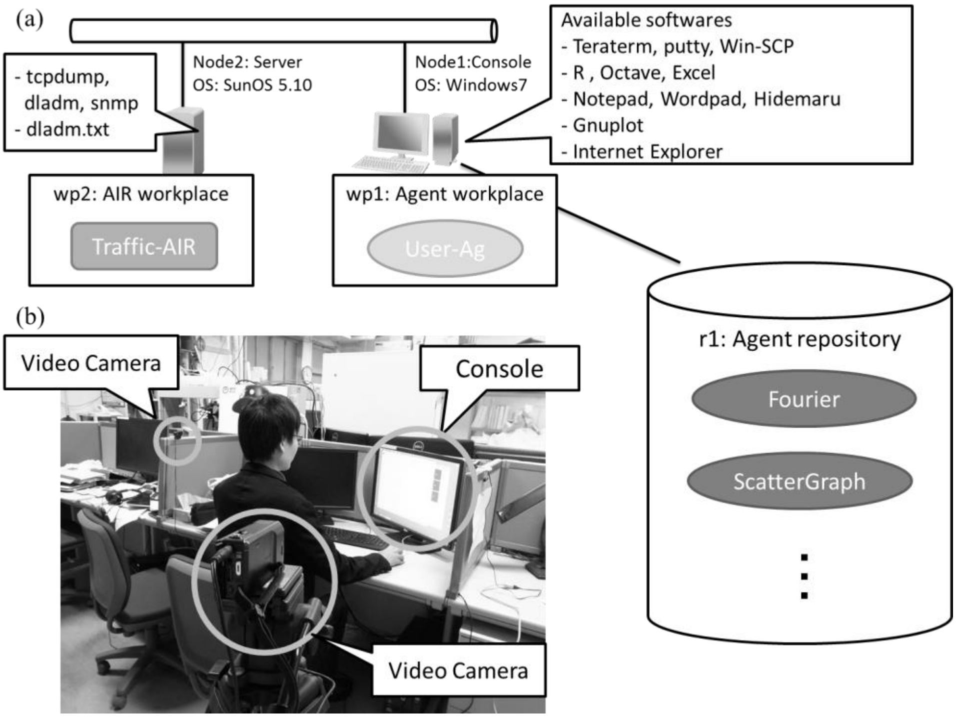

Figure 6. Experimental Environment

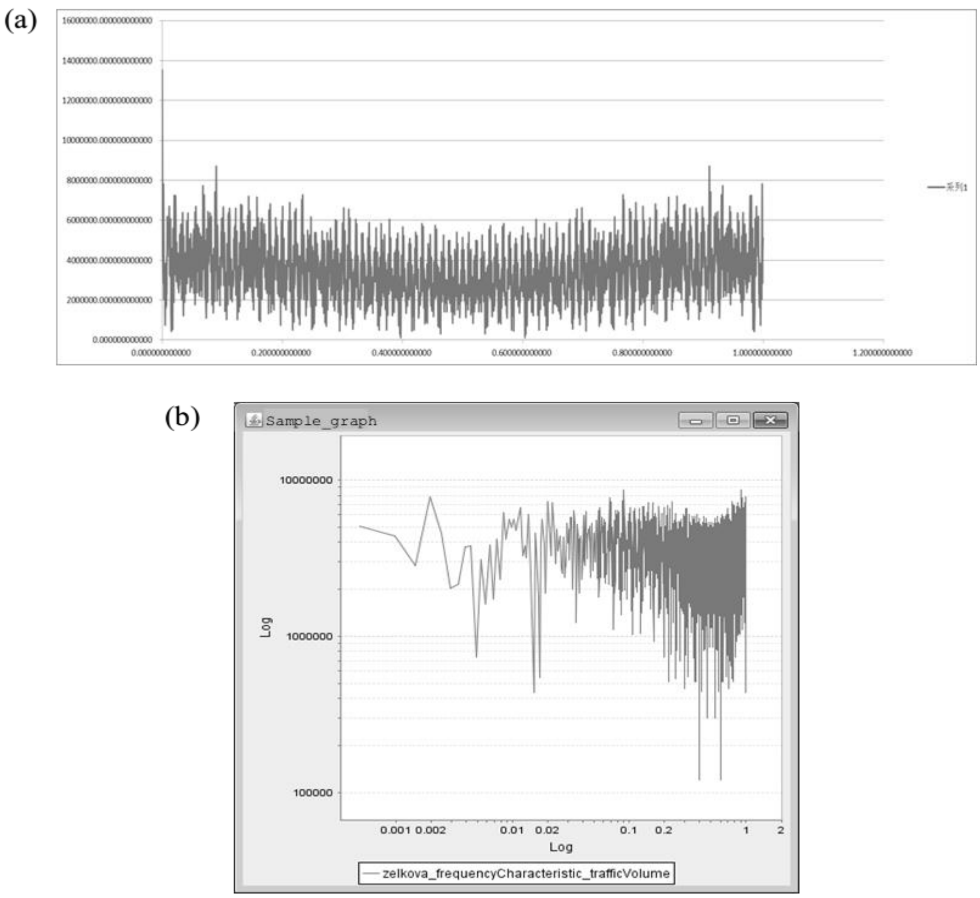

Figure 7. Example of Result Output

Table 5 shows the average value of consumption time, steps and their reduction rate of the participants. For all participants, the consumption time and steps are drastically reduced over 90%. From this result, we can confirm our prototype system effectively supported the participants' tasks. Further, we analyzed the support function of our prototype system in terms of the types of knowledge. Figure 8 shows the consumption time of manual case, according to the individual participant by separating with *active* and *inactive* steps, where active means the actions with the actual operations, e.g., input commands, and inactive means the actions connected with consideration, e.g., searching keyword on browser. In other words, since the inactive steps are related to the actions to obtain additional knowledge, thus, the steps concern the intellectual load on the analytics for the administrators. From the result, it is found that the time spent for the active steps are almost same length between the participants. However, for the inactive steps, it is easy to find difference between the participants. It implies, the difference of consumption time and steps for manual task depends on the skills, experiences and knowledge for network data analysis of individual participant. Thus, the prototypical system supported intellectual load rather than the operations, in other words, the effectiveness of the proposal is not only the automation of the task but also intellectual support.

Table 4. An Example of Recoded Steps

Step	Time	Extracted action
1	0:00:00	Execute WinSCP
2	0:00:22	Login to srvA
3	0:00:40	Change remote directory to where dladm.txt exists
4	0:00:51	Change local directory to save point
5	0:01:02	Copy dladm.txt
6	0:01:05	Open dladm.txt by Notepad.exe
7	0:01:14	Close WinSCP.exe
8	0:01:29	Open directory where the program exists with Explorer.exe
9	0:01:57	Open README for program
10	0:02:12	Execute Excel.exe
11	0:02:14	Open dladm.txt with Excel.exe
12	0:02:48	Change display format of the cells
13	0:03:28	Open README for program
14	0:03:40	Save as dladm.xlsx
15	0:04:26	Extract time stamp and interface name into Notepad.exe
16	0:04:57	Replace interface name with space
17	0:05:02	Start InternetExplorer.exe
18	0:05:31	Survey how to remove empty lines in Excel.exe
.	.	.
.	.	.
.	.	.
55	0:37:12	Select “scatter graph” with graph tool in Excel.exe
56	0:37:21	Change graph type to “scatter (chart)”
57	0:37:28	Change graph size
-	-	Finish

Table 5. Consumption Time and Steps

	Proposed		Manual		Reduction rate	
Participant	Time [sec]	Steps [count]	Time [sec]	Steps [count]	Time [%]	Steps [%]
A	145	8	5922	59	97.6	86.4
B	97	7	5275	67	98.2	89.6
C	89	6	2248	58	96.0	89.7
D	150	7	4063	88	96.3	92.0
E	151	6	3056	60	95.1	90.0
F	144	7	3613	86	96.0	91.9
	107	7	4254	89	97.5	92.1
Avg.	**126**	**6.9**	**4057**	**72.4**	**96.7**	**90.2**

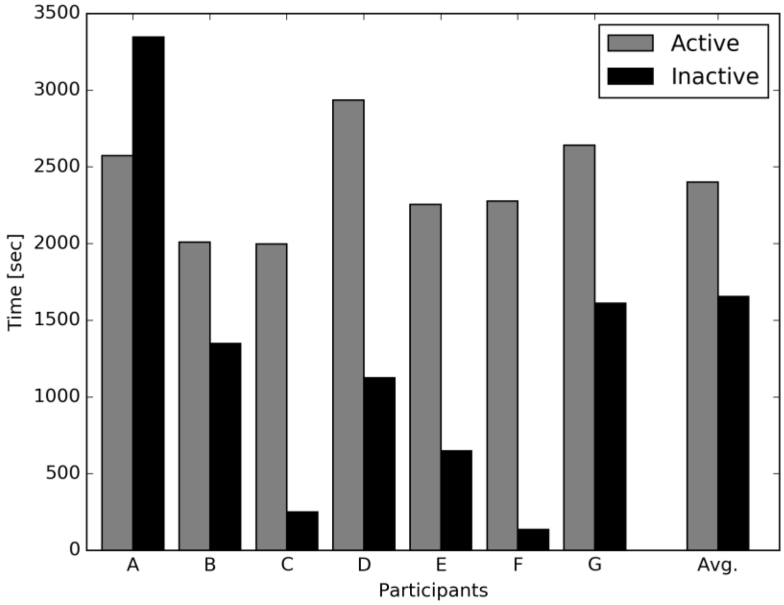

Figure 8. Active vs Inactive Time

5. Conclusion

In this paper, we propose an agent-based network data analysis support of collaboration with AIR-NMS. The Analysis Agents (AL-Ags) are categorized to three templates, Analyzer, Preprocessor, and Visualizer, and the templates can be organized according to the requested analysis pattern. The collaboration with AIR-NMS provides the function to retrieve appropriate network status data by using the functions in I-AIRs. To evaluate the prototypical implementation, we conducted an experiment to compare with and without the proposed tool in an analysis task. As a result, we conclude the prototypical analysis support system can reduce the burden of the users because the raw data is automatically retrieved and seamlessly processed in the system. For the future work, it needs to evaluate reusability and scalability of the organized analysis agents, and some anomaly detection and recovery situation is preferable to test effectiveness of the proposed analysis tool.

Acknowledgments

This work was partially supported by a Grant-in-Aid for Scientific Research (C) No.16K00292 from Japan Society for the Promotion of Science (JSPS).

References

[1] K. Shiomoto, "Applications of Big Data Analytics Technologies for Traffic and Network Management Data – Gaining Useful Insights from Big Data of Traffic and Network Management", NTT Technical Review, vol. 11, no. 11, **(2013)**, pp. 1-6.
[2] Z. Chen, Y. Zhu, Y. Di and S. Feng, "Self-Adaptive Prediction of Cloud Resource Demands Using Ensemble Model and Subtractive-Fuzzy Clustering Based Fuzzy Network", Computational Intelligence and Neuroscience, vol. 2015, **(2015)**, pp. 919805/1-14.
[3] G.K. Shyam, S.S. Manvi, "Virtual resource prediction in cloud environment: A Bayesian approach", Journal of Network and Computer Applications, vol. 65, **(2016)**, pp. 144-154.
[4] Q. Huang, K. Shuang, P. Xu, J. Li, X. Liu and S. Su, "Prediction-based Dynamic Resource Scheduling for Visualized Cloud Systems", Journal of Networks, vol. 9, no. 2, **(2014)**, pp. 375-383.
[5] P. Casas, A. D'Alconzo, T. Zseby, M. Mellia, "Big-DAMA: Big Data Analytics for Network Traffic Monitoring and Analysis", Proceedings of LANCOMM, **(2016)**, pp. 1-3.
[6] A. Scherrer, N. Larrieu, P. Owezarski, P. Borgnat and P. Abry, "Non-Gaussian an d Long Memory Statistical Characterizations for Internet Traffic with Anomalies", IEEE Transactions on Dependable and Secure Computing, vol. 4, no. 1, **(2007)**, pp. 56-70.
[7] X. Ma, C. Hu and K. Chen, "Error Tolerant Address Configuration for Data Center Networks with Malfunctioning Devices", Proceedings of 32nd International Conference on Distributed Computing Systems", **(2012)**, pp. 708-717.
[8] N. El-Sayed, I.A. Stedanovici, G. Amvrosiadis, A.A. Hwang and B. Schroeder, "Temperature management in data centers: why some (might) like it hot", Proceedings of 12th ACM SIGMETRICS/PERFORMANCE joint international conference on Measurement and Modeling of Computer Systems, **(2012)**, pp. 163-174.
[9] H. Demirkan and D. Delen, "Leveraging the capabilities of service-oriented decision support systems: Putting analytics and big data in cloud", Decision Support Systems, vol. 55, no. 1, **(2013)**, pp. 412-421.
[10] D. Chong and H. Shi, "Big data analytics: a literature review", Journal of Management Analytics, vol. 2, no. 3, **(2015)**, pp. 175-201.
[11] Google Analytics, https://www.google.com/analytics/.
[12] Microsoft Azure, https://azure.microsoft.com/en-us/.
[13] IBM Watson Analytics, http://www.ibm.com/analytics/watson-analytics/.
[14] N. Samaan and A. Karmouch, "Towards Autonomic Network Management: an Analysis of Current and Future research Directions", IEEE Communications Surveys & Tutorials, vol. 11, no. 3, **(2009)**, pp. 22-36.
[15] M. Flanders, "What is the biological basis of sensorimotor integration?", Biological Cybernetics, vol. 104, **(2011)**, pp. 1-8.
[16] M. Maggio, H. Hoffmann, A.V. Papadopulos, J. Panerati, M.D. Santambrogio, A. Agarwal and A. Leva, "Comparison of Decision-Making Strategies for Self-Optimization in Autonomic Computing Systems", ACM Transactions on Autonomous and Adaptive Systems, vol. 7, no. 4, **(2012)**, pp. 36:1-32.
[17] N. Paton, M.A.T. de Aragão, K. Lee, A.A.A. Fernandes and R. Sakellariou, "Optimizing utility in cloud computing through autonomic workload execution", Bulletin of the Technical Committee on Data Engineering, vol. 32, no. 1, **(2009)**, pp. 51-58.
[18] J.O. Kephart and J. Lechner, "A Symbiotic Cognitive Computing Perspective on Autonomic Computing", Proceedings of IEEE 12th International Conference on Autonomic Computing, **(2015)**, pp. 109-114.
[19] K. Sasai, J. Sveholm, G. Kitagata and T. Kinoshita, "A Practical Design and Implementation of Active Information Resource based Network Management System", International Journal of Energy, Information and Communications, vol. 2, no. 4, **(2011)**, pp. 67-86.
[20] T. Kinoshita, G. Kitagata, H. Takahashi, K. Sasai and K. Kalegele, "An Agent-based Network Management System", International Journal of Advanced Smart Convergence, vol. 2, no. 2, **(2013)**, pp. 10-15.
[21] A. Bieszczad, B. Pagurek, T. White, "Mobile Agents for Network Management", IEEE Communications Surveys, vol. 2, no. 2, **(1998)**, pp. 10-15.
[22] A. Terauchi, O. Akashi, M. Maruyama, T. Sugawara, K. Fukuda, T. Hirotsu, S. Kurihara, K. Koyanagi, "Agent organization system for multi-agent based network management", Transactions of the Japanese Society for Artificial Intelligence, vol. 22, no. 5, **(2007)**, pp. 482-492.
[23] K. Kalegele, K. Sasai, H. Takahashi, G. Kitagata, T. Kinoshita, "Four Decades of Data Mining in Network and Systems Management", IEEE Transactions on Knowledge & Data Engineering, vol. 27, no. 10, **(2015)**, pp. 2700-2716.
[24] T. Kinoshita and K. Sugawara, "ADIPS Framework for Flexible Distributed Systems", Multiagent Platforms, Edited T. Ishide, LNAI, vol. 1599, **(1998)**, pp. 326-333.
[25] T. Uchiya, T. Maemura, L. Xiaolu, T. Kinoshita, "Design and Implementation of Interactive Design Environment of Agent System", LNAI, vol. 4570, **(2007)**, pp. 1088-1097.
[26] A. Shevtekar, K. Anantharam and N. Ansari, "Low Rate TCP Denial-of-Service Attack Detection at Edge Routers", IEEE Communications Letters, vol. 9, no. 4, **(2005)**, pp. 363-365.

[27] I. Csabai, "1/f noise in computer network traffic", Journal of Physics A: Mathematical and General, vol. 27, np. 12, **(1994)**, pp. L417.

Authors

Kazuto Sasai, is an assistant professor of Research Institute of Electrical Communication, Tohoku University, Japan. He received his Dr. Sci. in Earth and Planetary Science from Kobe University. His research interests include network management, network analysis, complex systems, social systems and agent-based systems. Dr. Sasai is a member of IEICE, IPSJ, JSAI, BSJ.

Yusuke Tanimura, received his B.E. degree and the M.S. in Information Sciences from Tohoku University in 2013 and 2015. He is currently a Ph.D. student at the Graduate School of Information Sciences (GSIS), Tohoku University. His research interests include network management and agent-based computing. He is a member of IPSJ.

Hideyuki Takahashi, is an assistant professor of Research Institute of Electrical Communication of Tohoku University, Japan. He received his doctoral degree in Information Sciences from Tohoku University in 2008. His research interests include ubiquitous computing, Internet of Things, and agent-based computing, Internet of Things, and agent-based computing. He is a member of IEICE and IPSJ.

Gen Kitagata, born in 1972, is currently an associate professor at RIEC (Research Institute of Electrical Communication), Tohoku University, Japan. He received a doctoral degree from Graduate School of Information Sciences, Tohoku University. His research interests include agent-based computing, intelligent networking, and resilient networking. He is a member of IEICE, IPSJ.

Tetsuo Kinoshita is a professor of Research Institute of Electrical Communication of Tohoku University. He received the B.E. degree in electronic engineering from Ibaraki University, Japan, in 1977, and the M.E. and Dr. Eng. degrees in information engineering from Tohoku University, Japan, in 1979 and 1993, respectively. His research interests include agent engineering, knowledge engineering, knowledge-based systems and agent-based systems. Dr. Kinoshita is a member of IEEE, ACM, AAAI, IEICE, IPSJ, and JSAI.