Computer Networks 94 (2016) 285–306

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Rich dynamic mashments: An approach for network management based on mashups and situation management

Oscar Mauricio Caicedo Rendon [a,*], Felipe Estrada-Solano [a], Vinicius Guimarães [b], Liane Margarida Rockenbach Tarouco [b], Lisandro Zambenedetti Granville [b]

[a] Telematics Department - University of Cauca, Street 5 # 4–70 - Popayán, CA - Colombia
[b] Institute of Informatics - Federal University of Rio Grande do Sul, Av. Bento Gonçalves, 9500 - Porto Alegre, RS - Brazil

ARTICLE INFO

Article history:
Received 30 January 2015
Revised 14 August 2015
Accepted 5 October 2015
Available online 10 November 2015

Keywords:
Mashup
Network management
Situation management
Web-based network management

ABSTRACT

In network management, significant research efforts have been carried out to automate and facilitate the tasks conducted by network administrators. However, so far, none of these efforts has exploited the opportunities of jointly using the Situation Management discipline and the mashup technology for network management. This paper introduces an approach, called Rich Dynamic Mashments, to facilitate the daily work of network administrators when dealing with unexpected, dynamic, and heterogeneous situations. We have referred to as *nmsits* to such type of network management situations (*e.g.*, a sudden packet loss in a core router of a network backbone and an unforeseen slowness in data transmission over a link between virtual routers) and *mashments* to tunable mashups that use Situation Management for conducting network management tasks. The proposed approach is made up by the models of *nmsits* and *mashments*, the mechanisms to automatically recognize *nmsits* and dynamically compose *mashments*, and the architecture supporting these models and mechanisms. We further implement a prototype of our approach and conduct an extensive analysis on networks based on the Software-Defined Networking paradigm. The analysis results have provided directions and evidence that corroborate the feasibility of using Rich Dynamic Mashments as an effective approach for network management in terms of time-recognition, time-composition, time-consuming, time-response, and network traffic.

1. Introduction

The Situation Management (SM) discipline is intended to address situations happening or that might happen in dynamic systems [1]. SM aims to provide solutions that enable analyzing, correlating, and coordinating interactions among people, information, technologies, and actions targeted to overcome situations [2]. The basis of SM are [3]: (*i*) a situation modeled as a collection of entities in a domain, their attributes, and relationships in a time interval, (*ii*) the investigative aspect related to retrospective cause analysis of situations, (*iii*) the control aspect devised to change or preserve situations; and (*iv*) the predictive aspect aimed to foresee situations.

SM has been employed in diverse domains, such as disaster response [4], smart power grids [5], civil crisis [6], aviation [7], public health [8], electric power systems [9], and medical emergencies [10]. However, up to now, SM has not been used to deal with unexpected, dynamic, and heterogeneous situations that network administrators face in their daily work. Hereinafter, such type of situations are referred

* Corresponding author. Tel.: +57 3104918132.
E-mail addresses: omcaicedo@unicauca.edu.co, omcaicedo@gmail.com, omcrendon@inf.ufrgs.br (O.M. Caicedo Rendon), festradasolano@unicauca.edu.co (F. Estrada-Solano), vtguimaraes@inf.ufrgs.br (V. Guimarães), liane@penta.ufrgs.br (L.M. Rockenbach Tarouco), granville@inf.ufrgs.br (L.Z. Granville).

http://dx.doi.org/10.1016/j.comnet.2015.10.024
1389-1286/

to as *nmsits* [11]. Some examples of *nmsit* are: (*i*) a sudden packet loss in a core router of a network backbone, (*ii*) an unforeseen slowness in data transmission over a link between two virtual routers; and (*iii*) an unexpected increases in the number of corrupted packages transmitted by switches handled by an OpenFlow Controller.

Mashups are composite Web applications centered on end-users and created by combining resources (*e.g.*, data, application logic, and user interfaces) available along the Web [12]. In the previous definition, end-user centric means that mashups may be developed by users without advanced programming skills [13]. Mashups have been used in several domains, such as fire emergencies [14], telco services [15], and immersive mirror worlds [16]. Also, we have analyzed the mashup technology as a mechanism to compose network management applications [17] and accomplish specific tasks like virtual nodes monitoring [18].

Although a large number of research efforts [19–25] has been carried out to support management tasks, to the best of our knowledge, none of such efforts has jointly used SM and mashups to automate and facilitate the work of network administrators. In our previous work [11,26], we introduced the concept of *mashments* (*i.e.*, tunable mashups that automate the investigative and control aspects of SM for carrying out network management). We observed that *mashments* are a suitable approach to facilitate the tasks of network administrators when facing *nmsits*. Nevertheless, we have identified some features that are missing in current *mashments*: (*i*) they are not able to automatically recognize *nmsits*, constraining the analysis and resolution of such situations; and (*ii*) they are not dynamically composed, limiting the overcoming of recognized *nmsits*.

In this paper, we take a step further and introduce Rich Dynamic Mashments (RDM) to facilitate the work of network administrators when facing *nmsits*. We argue that the use of mechanisms to automatically recognize *nmsits* and dynamically compose *mashments* allows to make timely decisions in a less complex way. Our major contributions are

- Mechanisms to automatically recognize *nmsits* and dynamically compose *mashments*.
- An architecture that supports the above mentioned mechanisms and enables building up RDMs.
- A prototype that implements the proposed architecture.
- The demonstration, in a network that follows the Software-Defined Networking (SDN) paradigm, of the feasibility of using RDM to deal effectively with *nmsits* in terms of time-recognition, time-composition, time-consuming, time-response, and network traffic.

The remainder of this paper is organized as follows. In Section 2, we present scenarios and challenges of *nmsits*. In Section 3, we present related work. In Section 4, we introduce the RDM Architecture. In Section 5, we describe and discuss the proof-of-concept used to evaluate RDM. Finally, in Section 6, we provide conclusions and implications for future work.

2. Scenarios and challenges

In this section, we introduce motivating scenarios and their challenges. In the first scenario, let's consider network administrators manage a large company. In this company, the communication between the Pin Pads shops and the Enterprise Resource Planning system is provided by an outsourced Internet Service Provider (ISP). If a sudden failure in packet transmission in the ISP takes place or if an internal connection error in the border router of one or more shops occurs, the company might lose a significant amount of revenues because the payment by cards becomes inoperative.

In the second scenario, let's assume network administrators manage an SDN-based Network Operator. This operator provides network infrastructure to Small and Medium Enterprises (SMEs) using resources, such as OpenFlow controllers and virtual switches, supplied by several Virtual Network Providers. If an abrupt performance degradation in virtual links of one or more SMEs (*e.g.*, generated by unidentified errors in the virtual switches of providers) occurs, the operator might break the Service Level Agreements established with SMEs and, as a result, lose money.

In these both scenarios, network administrators need to easily and rapidly overcome *nmsits*. In particular, they face the following challenges: (*i*) conduct situational management operations (*e.g.*, collect, split, filter, add, and merge data) on multiple and heterogeneous devices/networks involved in *nmsits*, (*ii*) compose and tune solutions for *nmsits* in a less complex and time consuming way; and (*iii*) visualize information of *nmsits* in an understandable and friendly way.

The challenges of *nmsits* may be addressed, among other, with the following options. The first one is to use several mismatched network management solutions from the industry [27,28] and academy [24] [25], but it hinders and overloads the tasks of network administrators. Thus, this option is complex and time consuming.

A second option to cope with *nmsits*, it is to improve existing network management solutions, such as Nagios [29], ZenOSS [27], and OpenNMS [28], by deploying on them plugins that support SM. To the best of our knowledge, up to now, there is not a research that develops and adds plugins/packages based on the SM discipline to extend and enhance the above-referred solutions.

A third option to deal with *nmsits*, it is to use home-brewed scripts that integrate two or more network management solutions. The weaknesses of this option are, first, the skill required to write and run scripts (the development of scripts is a daunting and complex task for network administrators who usually do not have advanced programming knowledge). Second, the loss of focus and time of network administrators; instead of management the network itself, they are forced to acquire knowledge that is not necessarily relevant to their daily tasks.

A fourth option to address *nmsits*, it is to use our *mashments* [11,26]. In a broad sense, a mashment is a solution based on the SM discipline and the mashup technology for carrying out network management in an effective way. In particular, a mashment is defined as a tunable mashup that combines diverse types of resources from multiple providers and automates the investigative and control aspects of SM, aiming to facilitate the work of network administrators.

We argue that *mashments* are closer and more appropriate to facilitate the daily needs of network administrators. Notwithstanding, as the network administrator is even